IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZRXA-E-D
PRODUCT NAME:	RX11 SYSTEM RELIABILITY TEST
DATE:	APRIL 1976
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	DAVID L. ADAMS

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT

THE RX11 SYSTEM RELIABILITY PROGRAM CONSISTS OF SELECTABLE TESTS THAT CHECK THE OPERATION OF THE RX11 SYSTEM, BY WRITING, READING AND VERIFYING VARIOUS DATA PATTERNS, UNDER VARIOUS (SELECTABLE) HEAD MOVEMENTS. IT CAN TRANSFER DATA AND CHECK FOR ERRORS OVER THE ENTIRE DISKETTE, ALL TRACKS AND SECTORS, OR BETWEEN SEPARATELY SELECTABLE TRACK AND SECTOR ADDRESS LIMITS.

AS WELL AS TRANSFERRING DATA THE PROGRAM ACCUMULATES STATICTICAL DATA ON THE FOLLOWING:

A. COMPLETED PASSES OF THE PROGRAM
B. NUMBER OF RESTARTS
C. NUMBER OF SECTORS WRITTEN/READ
D. RECOVERABLE AND UNRECOVERABLE FAULTS AS FOLLOWS:

1. PARITY ERRORS
2. ERROR FLAG ERRORS
3. INTERRUPT ERRORS
4. SEEK ERRORS
5. DATA CRC ERRORS
6. CRC NO DATA ERRORS
7. DATA NO CRC ERRORS
8. DELETED DATA MARK ERRORS
9. WRITE ERRORS
10. READ ERRORS

E. TOTAL OF EACH ERROR CODE DETECTED
F. TOTAL OF TIMES EACH TRACK WAS ACCESSED
G. TOTAL OF TIMES HEAD MOVED TO A TRACK.
H. TOTAL OF ERRORS DETECTED PER TRACK PER DRIVE.

THE ABOVE INFORMATION IS REPORTED BY RUNNING THE ERROR DUMP PROGRAM.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

THE FOLLOWING EQUIPMENT IS REQUIRED.

A. PDP-11 SERIES OF COMPUTER WITH MIN 8K MEMORY

B. RX11 FLOPPY DISK SYSTEM INCLUDING DUAL OR SINGLE DRIVE RX01 AND PDP-11 INTERFACE. (SEE SECTION 2.3 FOR SELECTION OF REGISTER ADDRESSES AND VECTOR ADDRESS)
NOTE: A DISKETTE MUST BE INCLUDED WITH EACH DRIVE TESTED.

C. CONSOLE TELEPRINTER

1.2.1 SOFTWARE REQUIREMENTS

THIS PROGRAM ASSUMES THAT THE RX11 INTERFACE DIAGNOSTIC (MAINDEC - 11 - DZRX8-*) HAS BEEN SUCCESSFULLY RUN ON THIS SYSTEM.

2.0 OPERATING INSTRUCTIONS

2.0.1 OUTLINE OF OPREATING PROCEDURE

THE STANDARD OPERATING PROCEDURE FOR THE RX11 RELIABILITY TEST (TO RUN ALL TESTS ON BOTH DRIVES WITH NO OPERATOR INTERVENTION VIA THE SWITCH REGISTER) IS AS FOLLOWS:

A. LOAD THE PROGRAM INTO MEMORY
   1. IF IT'S BEING LOADED FROM A DISKETTE,REPLACE THE "LIBRARY DISKETTE" WITH A "SCRATCH DISKETTE"

B. START THE PROGRAM RUNNING AT LOCATION 200

C. THE PROGRAM WILL TYPE OUT THE FOLLOWING:
   1. PROGRAM NAME AND REVISION
   2. RX11 REGISTER AND VECTOR ADDRESSES
   3. UNITS BEING TESTED
   4. "P"ATTERN, "T"EST, AND "S"EQUENCE

   IT THEN STARTS RUNNING UNDER THOSE CONDITIONS

D. IF THERE ARE NO ERRORS, AT THE END OF THE COMPLETED PASS A "D" AND "BELL" IS TYPED,AND THE PROGRAM WILL CONTINUE ON FOR ANOTHER PASS.

E. HALT THE PROGRAM AS FOLLOWS:

   1. IF THERE IS A HARDWARE SWITCH REGISTER, PUT SW 14 UP TO HALT AT THE END OF PASS.

   2. IF THERE IS NO SWITCH REGISTER, OR TO HALT AT ANY TIME, HALT THE PROCESSOR.

2.1 LOADING PROCEDURE

LOAD THE PROGRAM INTO MEMORY USING STANDARD PROCEDURE FOR BINARY PAPER TAPES.

MAKE SURE THAT THE TOTAL SYSTEM IS READY FOR OPERATION, DISKETTE INSERTED CORRECTLY, DUORS CLOSED ON DRIVES TO BE TESTED, ETC.

2.2 STARTING ADDRESSES

THE PROGRAM HAS THREE (3) STARTING LOCATIONS.

2.2.1 INITIAL START (LOC. 200)

THIS STARTING LOCATION INITIALIZES THE PROGRAM AS FOLLOWS:

A. CLEARS ALL ERROR LOGS
B. RESETS COUNTERS AND CONSTANTS
C. TYPES OUT RX11 REGISTER AND VECTOR ADDRESSES BEING USED.
D. TYPES THE DRIVE AND TEST SELECTION IN "DTESTP"
E. INITIATES THE COLLECTION OF STATISTICAL DATA PERTAINING TO THE OPERATION OF THE RX11 SYSTEM.

2.2.2 RESTART (LOC. 202)

THIS STARTING LOCATION DIRECTS THE PROGRAM TO CONTINUE RUNNING USING DRIVE AND TEST SELECTIONS SPECIFIED IN THE PREVIOUS INITIAL START. IT ALSO CONTINUES TO ACCUMULATE STATICTICAL INFORMATION, AMMENDING IT TO THE DATA ALREADY COLLECTED PRIOR TO THIS RESTART.

2.2.3 ERROR DUMP (LOC. 204)

THIS STARTING ADDRESS DIRECTS THE PROGRAM TO PRINT OUT ON THE CONSOLE TELEPRINTER ALL THE DATA ACCUMULATED FROM THE LAST "INITIAL START" TO THE TIME OF THE PRINTOUT. TYPING OUT THIS INFORMATION DOES NOT DESTROY IT, SO A RESTART CAN BE INITIATED AND INFORMATION WILL CONTINUE TO BE COLLECTED.

2.3 OPERATOR ACTION BEFORE STARTING PROGRAM

2.3.1 DEVICE ADDRESS SELECTION

LIKE MOST OPTIONS ON THE PDP11 THE RX11 INTERFACE CARD HAS JUMPERABLE REGISTER AND VECTOR ADDRESSES. THIS ALLOWS FOR DEVICES WITH THE SAME STANDARD ADDRESSES TO BE JUMPERED TO AN OTHER ADDRESS SO THEY WILL RUN WITHOUT CONFLICT.

THE PROGRAM MUST KNOW WHAT ADDRESSES ARE BEING USED, AS IT IS THROUGH THESE REGISTERS AND VECTOR THAT ALL COMMUNICATION BETWEEN THE PDP11 AND RX11 IS HANDLED.

IF THE RX11 SYSTEM UNDER TEST IS JUMPERED FOR REGISTER ADDRESS OTHER THAN STANDARD, WHICH IS RXCS=177170 AND RXDB=177172. PLACE IN THE MEMORY LOCATION CALLED "RXCS" (LOC. 1206) ITS NEW ADDRESS, AND IN LOCATION "RXDB" (LOC. 1210) ITS NEW ADDRESS. IF THERE AS A NONSTANDARD INTERRUPT VECTOR ADDRESS (STANDARD IS LOC 264) THEN PLACE IN MEMORY LOCATION CALLED "INTVEC" (LOC. 1204) ITS NEW ADDRESS.

IF THESE THREE MEMORY LOCATIONS DO NOT CONTAIN THE ADDRESSES THAT THE INTERFACE BOARD IS WIRED TO THE PROGRAM WILL REPORT "TEST HUNG" AND HALT, OR HALT AT THE VECTOR ADDRESS THAT IS JUMPERED IN. (NOTE: THE VECTOR ADDRESSES CAN NOT BE JUMPERED FOR LOCATIONS 200 THROUGH 220 AS THESE ARE USED BY THE PROGRAM)

2.3.2 NON-STANDARD DISKETTE ADDRESS SELECTION

IF IT IS DESIREABLE TO TEST THE DISKETTES BETWEEN TRACK AND SECTOR ADDRESS LIMITS OTHER THAN BETWEEN THE NORMAL OUTER DIAMETER (OD) AND INNER DIAMETER (ID) TRACK ADDRESSES, AND/OR MINIMUM (FIRST) AND MAXIMUM (LAST) SECTOR ADDRESS. THIS IS DONE BY THE OPERATOR MAKING CHANGES TO TWO (2) MEMORY LOCATIONS BEFORE THE PROGRAM IS STARTED. ONE LOCATION IS CALLED OD WHICH CONTAINS TWO BYTES ONE FOR OD AND THE OTHER ID TRACK ADDRESSES. THE OTHER LOCATION IS CALLED FIRST AND IT TOO CONTAINS TWO BYTES ONE FOR FIRST THE OTHER FOR LAST SECTOR ADDRESS. (IF THESE TWO LOCATIONS ARE LEFT CLEARED THE MAX AND MIN VALUES FOR THE ADDRESSES ARE ASSUMED.)

A. DEFINITIONS:

OD = ADDRESS OF TRACK AT OUTER DIAMETER (MIN VALUE=0)
ID = ADDRESS OF TRACK AT INNER DIAMETER (MAX VALUE=114 [OCTAL])
FIRST = ADDRESS OF FIRST SECTOR OF TRACK (MIN VALUE=1)
LAST = ADDRESS OF LAST SECTOR OF TRACK (MAX VALUE=32 [OCTAL])

B. LOCATIONS:

```
TRACKS LOC. 1200 BITS   14------8   6------0
                           ID          OD

SECTORS LOC. 1202 BITS  12------8   4------0
                          LAST        FIRST
```

C. RESTRICTIONS:

THE CONTENTS OF "OD" MUST BE LESS THAN OR EQUAL TO THE CONTENTS OF "ID"
THE CONTENTS OF FIRST MUST BE LESS THAN OR EQUAL TO THE CONTENTS OF "LAST"

IF THESE LOCATIONS ARE CHANGED TO NEW LIMITS, THEN THE PROGRAM WILL ACCESS ONLY THOSE ADDRESSES INCLUSIVE OF AND BETWEEN THESE LIMITS.

2.3.3 SOFTWARE SWITCH REGISTER (LOC. 176)

FOR THE PDP 11 PROCESSORS THAT DO NOT HAVE A HARDWARE SWITCH REGISTER, OR IF THE OPERATOR WISHES TO SELECT THE SOFTWARE SWITCH REGISTER,BY PUTTING ALL THE SWITCHES UP TO A "1",(THIS MUST BE DONE EACH TIME THE PROGRAM IS STARTED AT LOCATION 200, OTHER WISE THE PROGRAM WILL USE THE HARDWARE SWR.) LOCATION 176 IS ASSIGNED AS THE SWITCH REGISTER. BITS SET TO A "1" IN THIS LOCATION HAVE THE SAME FUNCTION AS THE CORRISPONDING SWITCH IN THE HARDWARE SWITCH REGISTER. ALL REFERENCES TO THE SWR ARE INDIRECT AND THE PROGRAM ASSIGNS THE CORRECT ADDRESS OF THE SWR AT "INITIAL START". SEE SECTION 2.4.2 FOR THE SELECTION OF OPERATING CONDITIONS.

TO CHANGE THE SOFTWARE SWR. WHILE THE PROGRAM IS RUNNING, TYPE "CONTROL G". EACH TIME THE SWR. IS TO BE TESTED THE PROGRAM WILL CHECK TO SEE IF THE SOFTWARE SWR IS SELECTED,AND THE PROGRAM IS NOT RUNNING IN AUTO MODE OF RXDP/ACT11. IF BOTH CONDITIONS EXIST THEN THE PROGRAM CHECKS FOR THE CTRL G IN THE KEYBOARD BUFFER. IF THE CTRL G IS THERE THE CONTENTS OF THE SOFTWARE SWR. ARE PRINTED AND A "NEW =" IS ASKED FOR. THE OPERATOR MAY NOW TYPE IN THE NEW SWITCH REGISTER CONTENTS, TERMINATED BY A CARRIAGE RETURN (CR), OR IF HE DOESN'T WANT TO CHANGE THE SWR. JUST TERMINATE WITH THE (CR). NOTE SEE THE CHARACTER RESTRICTIONS BELOW.

WHEN THE PROGRAM DETECTS THE (CR) IT WILL REPLACE THE CONTENTS OF THE SOFTWARE SWR.,IF A NEW ONE HAS BEEN TYPED IN, AND RETURN TO THE FLOW OF THE PROGRAM.

NOTE: CHARACTER RESTRICTIONS FOR CHANGING THE SOFTWARE SWR.

1. ONLY OCTAL NUMBERS 0 - 7 ARE ACCEPTED. ANY OTHER CHARACTER TYPED WILL BE PRINTED AS A ? AND THE WHOLE SWR MUST BE RETYPED.

2. TO WIPE OUT A "NEW" CONTENTS JUST TYPED IN, TYPE CTRL U. NOW A NEW CONTENTS CAN BE RETYPED.

3. ONLY 6 OCTAL CHARACTERS WILL BE PUT INTO THE SWR. IF MORE THAN 6 CHARACTERS ARE TYPED IN ONLY THE LAST 6 WILL BE PUT INTO THE SWR.

2.3.4 TEST PARAMETER SELECTION ("DTESTP" LOC. 1212)

THE DRIVE AND TEST SELECTION MUST BE DONE BEFORE THE PROGRAM STARTS. "DTESTP" (LOCATION 1212) IS WHERE THE BITS ARE SET TO TELL THE PROGRAM WHAT DRIVES ARE WANTED AND WHAT TEST TO RUN, AS INDICATED BELOW. WHEN THE PROGRAM STARTS IT WILL TYPE OUT THE TEST CONDITIONS IT IS RUNNING UNDER.

BIT 15 (1) SELECT DRIVE UNIT 1
BIT 14 (1) SELECT DRIVE UNIT 0

THEN SET THE TEST CONDITIONS IN BITS 8 THROUGH 0 AS SHOWN BELOW.

"DTESTP" BITS	15 14	13 12 11 10	9	8 7 6	5 4 3	2 1 0
	U1 U0	NOT USED	D	P P P	T T T	S S S

BIT 9 IF BIT 9 IS ON, ALL WRITE/READ FUNCTIONS WILL BE IN THE DELETED DATA MODE.
BIT 8,7,6 SELECTS A DATA PATTERN TO BE USED.
BIT 5,4,3 SELECTS A TEST TO BE PERFORMED.
BIT 2,1,0 SELECTS A HEAD MOVEMENT SEQUENCE.

THE SELECTIONS ARE DEFINED AS FOLLOWS:

P = DATA PATTERN SELECTION

0	DEFAULT TO 7
1	ZEROS
2	ONES
3	FLOATING ZERO
4	FLOATING ONE
5	125
6	314
7	RANDOM

T = FUNCTIONAL TESTS

0	DEFAULT TO 7
1	WRITE ONLY
2	WRITE/READ
3	WRITE/READ CHECK
4	READ CHECK ONLY
5	READ ONLY (CRC CHECK)
6	WRITE/READ CHECK ON ALTERNATING DRIVES *
7	WRITE/READ/READ CHECK **

* NOTE: TEST 6 WRITES THEN READ CHECKS ANY SELECTED DATA PATTERN USING ANY TRACK SEQUENCE, BUT ONE TRACK AT A TIME, FIRST ON UNIT 0 THEN UNIT 1. WHEN BOTH UNITS HAVE ACCESSED THAT TRACK, IT GOES BACK TO UNIT 0 FOR THE NEXT TRACK, ETC.

** NOTE: THE FIRST HALF OF TEST 7 FORCES THE TRACK SEQUENCE TO INCREMENT UP THROUGH ALL TRACKS DOING WRITE/READ CHECK FUNCTIONS. THIS VERIFIES THAT ALL TRACKS ARE ACCESSABLE. THE SECOND HALF OF THE PASS WILL USE THE SEQUENCE SELECTED BY THE OPERATOR AS INDICATED BELOW, AND ONLY READ CHECK THE DATA JUST WRITEN. THIS VERIFIES THAT THE DATA CAN BE READ FROM A TRACK AFTER THE HEAD HAS BEEN MOVED AWAY FROM AND BACK TO THAT TRACK. AT THE COMPLETION OF THE PASS THE DELETED DATA BIT IN TEST CONDITIONS IS COMPLEMENTED AND THE NEXT PASS WILL BE RUN UNDER THIS NEW CONDITION.

S = TRACK SEQUENCING

0	DEFAULT TO 7
1	INCREMENT
2	DECREMENT
3	INCREMENT/DECREMENT
4	BOUNCE
5	DECREASING BOUNCE
6	STROBE
7	RANDOM

IF NO BITS ARE SET (ZEROED "DTESTP:) THEN THE PROGRAM WILL SELECT ALL DRIVE UNITS THAT ARE READY AND DEFAULT TO TEST CONDITIONS AS INDICATED.

THE PROGRAM NEXT PRINTS THE REGISTER AND VECTOR ADDRESSES IT WILL USE IN COMMUNICATING WITH THE RX11 SYSTEM, AS EXPLAINED IN SECTION 2.3.1 THE OPERATOR MUST VERIFY THAT THESE ADDRESSES ARE THE ONES JUMPERED ON THE RX11 INTERFACE BOARD.

2.4 OPERATOR ACTION TO RUN THE PROGRAM

2.4.1 STARTING THE PROGRAM

SET THE DESIRED STARTING ADDRESS INTO THE SWITCH REGISTER, DEPENDING UPON THE TYPE OF CONSOLE AVAILABLE, LOAD ADDRESS, AND PRESS START.

THE PROGRAM WILL TYPE ITS "MAINDEC" NUMBER AND REVISION, AND DEPENDING UPON THE STARTING ADDRESS DO THE FOLLOWING:

SA200 - THE PROGRAM WILL TYPE DRIVE AND TEST PARAMETERS, THE TWO REGISTER ADDRESSES, AND VECTOR ADDRESS. IT WILL THEN BEGIN FUNCTIONAL TESTING, AND INFORMATION COLLECTION.

SA202 - THE PROGRAM WILL CONFIGURE TO CONDITIONS SET IN PREVIOUS "INITIAL START", PRINT OUT THESE CONDITIONS AND CONTINUE FUNCTIONAL TESTING AND DATA COLLECTING. THE ONLY OPERATOR ACTION REQUIRED IS THE DYNAMIC SELECTION OF OPERATING CONDITIONS IN THE SWITCH REGISTER AS REQUIRED.

SA204 - THIS PROGRAM WILL REPORT VIA THE TELEPRINTER ALL DATA COLLECTED. NO OTHER OPERATOR ACTION IS REQUIRED.

2.4.2 OPERATING CONDITIONS

THE PROGRAM CHECKS FOR OPERATING CONDITIONS AT VARIOUS POINTS WHILE RUNNING. IF THERE IS A HARDWARE SWR THESE CONDITIONS CAN BE CHANGED AND SET WHILE THE PROGRAM IS RUNNING. IF THE SOFTWARE SWR IS IN USE, THEN THESE CONDITIONS MUST BE SET IN LOCATION 176 BEFORE THE PORGRAM STARTS.

SW15 = HALT ON ERROR
SW14 = HALT AT END OF PASS
SW13 = DON'T PRINT ERROR MESSAGE
SW12 = TYPE ONLY 10 DATA ERRORS
SW11 = NO RETRY ON ERROR. LOG HARD ERROR
SW08 = NO RECALIBRATION ON SEEK ERRORS
SW15-SW0 (1) = SELECT SOFTWARE SWITCH REGISTER

NOTE: IF THERE IS A HARDWARE SWITCH REGISTER, AND THE OPERATOR WANTS THE SOFTWARE SWITCH REGISTER, PUT ALL SWITCHES UP (1) BEFORE STARTING THE PROGRAM AT THE INITIAL START ADDRESS.

THE PROGRAM WILL PRINT A "DRIVE(S)" SELECTED CONFIRMATION MESSAGE THAT IT WAS SUCCESSFUL IN FINDING AT LEAST ONE DRIVE READY (DRY) CONDITION ON AN OPERATOR SELECTED DRIVE OR EITHER OR BOTH IF NO SELECTION WAS MADE. THE DRIVES CONFIRMED AS BEING SELECTED BY THE PROGRAM MAY DIFFER FROM THAT SELECTED BY THE OPERATOR IF THE PROGRAM DETECTED A DRIVE TO BE NOT READY.

IF HOWEVER THERE ARE NO DRIVES IN THE DRIVE READY CONDITION THE PROGRAM WILL TYPE "NO DRIVES READY" AND HALT, AS IT CAN'T FUNCTION WITH NO DRIVES READY. WHEN THE REASON FOR ALL DRIVES TO BE NOT READY IS FOUND AND CORRECTED, THE OPERATOR MAY PRESS CONTINUE AND THE PROGRAM WILL GO BACK TO "INITIAL START" OR HE MAY RELOAD THE STARTING ADDRESS HIMSELF.

AS WELL AS "DRIVE(S)" SELECTED THE PROGRAM WILL TYPE OUT THE TEST PARAMETERS SELECTED OR DEFAULTED TO. IF NON-STANDARD TRACK AND/OR SECTOR ADDRESS LIMITS WERE SELECTED (SEE SECTION 2.3.2 OF THIS DOCUMENT) THESE LIMITS WILL ALSO BE PRINTED OUT FOR VARIFICATION BY THE OPERATOR.

2.4.3 ACT11 & XXDP HOOKS

THE PROGRAM HAS THE NECESSARY LOCATIONS SET UP FOR OPERATION UNDER ACT11 AND XXDP OPERATION. THE PROGRAM LOOKS AT THE LOADING MEDIA LOCATION AND IF IT CONTAINS THE NUMBER 10, INDICATING THE FLOPPY DISK LOADED THE PROGRAM, WILL TYPE THE FOLLOWING PROMPT MESSAGE & WAIT FOR A USER RESONSE:

"CAUTION - IF YOU DESIRE TO TEST UNIT 0
REPLACE LOAD MEDIUM WITH A SCRATCH DISKETTE
THEN PRESS CONTINUE"

THUS, IF THE OPERATOR WISHES TO TEST DRIVE UNIT 0, AFTER LOADING THE PROGRAM, HE REMOVES THE LIBRARY DISKETTE FROM DRIVE 0, INSERTS A SCRATCH DISKETTE INTO DRIVE 0, AND PRESSES THE 'CONTINUE' SWITCH

2.5 PROGRAM OPTIONS

THERE ARE A COUPLE OF WAYS THE PROGRAM CAN BE SET UP TO RUN FOR OPTIONAL TESTING.

2.5.1 PSEUDO - SCOPE LOOP

BY SETTING THE ADDRESS LIMITS IN OD AND OR FIRST TO ONE OF TWO TRACKS AND OR SECTORS IT IS POSSIBLE TO PRODUCE A FAIRLY TIGHT PSEUDO - SCOPE LOOP. BY RUNNING ONLY A TEST THAT DOES WRITE ONLY OR READ ONLY ITS POSSIBLE TO SCOPE SPECIFIC FUNCTIONS, I.E. FILL BUFFER, WRITE, READ, EMPTY BUFFER.

2.5.2 DISKETTE COMPATABILITY

TO CHECK FOR DATA TRANSFER COMPATABILITY BETWEEN DRIVE UNITS USE THE FOLLOWING PROCEDURE. (IT IS ASSUMED THE SYSTEM HAS DUAL DRIVE RX01, IF NOT YOU MAY TEST COMPAT-ABILITY BETWEEN DIFFERENT RX11 SYSTEMS BY RUNNING THE SAME TESTS ON BOTH.)
THIS TEST INSURES PROPER HEAD ALIGNMENT.

A. HAVE DISKETTES IN BOTH DRIVES AND DRIVES READY.

B. CLEAR THE OD/ID AND FIRST/LAST MEMORY LOCATION, FOR TOTAL DISKETTE TRANSFERS.

C. LOAD THE INITIAL START ADDRESS (LOC 200) AND START THE PROGRAM RUNNING.

D. SET LOCATION "DTESTP" FOR "DRIVE AND TEST CONDITIONS" ON BOTH DRIVES, ANY "P"ATTERN OF DATA, "S"EQUENCE #1 (INCRIMENT TRACKS,TO INSURE ALL HEAD POSITIONS ARE ALIGNED) AND "T"EST 3 (WRITE/READ CHECK). THIS WILL WRITE THEN READ AND VERIFY THE DATA ON BOTH DISKETTES.

E. ALLOW THE PROGRAM TO RUN FOR AT LEAST 1 COMPLETE PASS.

NOTE: IF RANDOM PATTERN (0 OR 7) IS SELCTED YOU MUST HALT AT THE END OF THE FIRST PASS, AS ADDITIONAL PASSES CHANGES THE DATA AND YOU WILL GET DATA ERRORS WHEN YOU TRY TO REREAD.

TO HALT AT THE END OF PASS PUT SW14 UP (1) WHEN "OPERATING CONDITIONS" ARE REQUESTED BY THE PROGRAM.

F. AFTER COMPLETION OF THE PASS, PROGRAM HALTED. SWAP DISKETTES, AND AS YOU ONLY WANT TO READ VERIFY THE DATA, START THE PROGRAM AGAIN AT LOC 200 (INITIAL START).

G. WHEN REQUESTED SELECT THE SAME DRIVES, PATTERN, AND SEQUENCE, BUT SELECT TEST 4 (READ CHECK ONLY).

H. ALLOW THE PROGRAM TO RUN AS LONG AS YOU WISH TO VERIFY THAT DATA WRITTEN AND CHECKED ON ONE DRIVE CAN BE READ AND VERIFIED ON THE OTHER DRIVE.

2.5.3 RX11 TO RX8 COMPATABILITY

TO WRITE A DISKETTE ON THE RX11 AND READ/VERIFY THE SAME DATA ON A RX8 REQUIRES THE FOLLOWING:

A. SET UP LOCATION "DTESTP" (1212) WITH ONE OF THE FOLLOWING DATA PATTERNS

P=1 (0'S)
P=2 (1'S)
P=5 (125) *
P=6 (314) *

* NOTE: IF ONE OF THESE PATTERNS IS SELECTED A MODIFICATION TO THE PROGRAM MUST BE MADE TO ALLOW THIS DATA TO BE READ ON A RX8 SYSTEM.
THE MODIFICATION IS IN THE "PAT125" ROUTEEN, THE TWO (2) LOCATIONS CONTAINING THE "COMB DATABYTE" INSTRUCTION MUST BE CHANGED TO TWO (2) NOP'S (OCTAL 000240)

B. SET IN "DTESTP" TEST 3 WHICH WILL WRITE AND READ CHECK THE DATA THEREBY VERIFYING THE QUALITY OF THE DATA PRIOR TO THE DISKETTE SWAP.

SET THE SEQUENCE TO 1 TO INSURE THAT ALL TRACKS HAVE BEEN WRITTEN ON.

C. HALT THE PROGRAM AT THE COMPLETION OF ONE PASS.

D. THE DISKETTE IS NOW READY TO BE READ ON THE RX8 SYSTEM. SEE MD-8-DIRX8-* FOR THE PROCEDURE TO READ THIS DISKETTE.

TO READ A DISKETTE THAT WAS WRITTEN ON A RX8 SYSTEM SET UP AS FOLLOWS:

E. SET IN "DTESTP" THE DATA PATTERN THAT WAS USED BY THE RX8 SYSTEM. (SEE NOTE UNDER SECTION A.)

F. SET "DTESTP" FOR TEST 4 (READ CHECK ONLY) AND SEQUENCE 1 TO CHECK ALL TRACKS. START THE PROGRAM, IT SHOULD RUN WITHOUT DATA ERRORS.

2.6 RUN TIME

RUN TIME PER PASS DEPENDS UPON NUMBER OF A FUNCTIONS IN THE TEST SELECTED AND THE TRACK SEQUENCE.
EXAMPLES OF RUN TIMES FOLLOW.
(THESE TIMES ARE FOR 1 DRIVE AND COMPLETE DISKETTE,MAXIMUM ID/OD, FIRST/LAST LIMITS.IF OPERATING ON 2 DRIVES DOUBLE THE TIME.)
THESE TIMES ARE FOR A PDP 11/05 PROCESSOR AND MAY CHANGE SLIGHTLY FOR A FASTER PROCESSOR.

TEST SELECTION	P	T	S	TIME/PASS
	7	6	5	3 MIN.30 SEC.
	7	7	4	2 MIN.47 SEC.
	7	7	7	2 MIN.23 SEC.
	6	3	4	1 MIN.46 SEC.
	7	3	1	1 MIN.30 SEC.
	7	1	1	51 SEC.

NOTE: DUE TO THE SLOW SPEED OF THE LSI 11 PROCESSOR, THE RUN TIME IS ABOUT DOUBLE THAT LISTED ABOVE. TO SPEED UP THE RUNNING OF THIS PROGRAM IN THE LSI 11 YOU CAN CHANGE THE INTERLEAVE FACTOR, OF THE SECTORS, USED IN THE PROGRAM. TO DO THIS CHANGE THE CONTENTS OF LOCATION "THREE" FROM OCTAL 3 TO OCTAL 5.

3.0 ERROR DETECTION

3.1 PROGRAM DEFINITIONS

ON MOST ERRORS THE PROGRAM WILL TYPE OUT THE CONTENTS OF "STATUS A" AND "STATUS B".

STATUS A IS THE CONTENTS OF THE RXES (ERROR AND STATUS REGISTER) AT THE TIME THE ERROR IS DETECTED. IT SHOWS THE CRC,PAR,ETC. ERRORS.

STATUS B IS THE "DEFINITIVE ERROR CODES" THAT THE RX01 DETECTED, THAT MAY HAVE CAUSED THE ERROR CONDITION. THESE ERROR CODES ARE DEFINED IN SECTION 3.2.

THE PROGRAM HAS DEFINED THE FOLLOWING AS ERRORS.

3.1.1 WRITE ERROR

A WRITE ERROR IS A RETRIED READ ERROR IF THE DATA BEING READ IS OF UNKNOWN QUALITY (THE DATA READ IS BEING READ FOR THE FIRST TIME AFTER A WRITE FUNCTION)

3.1.2 READ (CRC) ERROR

A READ ERROR IS A RETRIED READ ERROR WHERE THE QUALITY OF THE DATA BEING READ IS KNOWN GOOD. (THE DATA HAS BEEN READ CORRECTLY SOME TIME PREVIOUSLY.)

3.1.3 CRC AND DATA ERROR

3.1.4 NO CRC ERROR BUT DATA ERROR

3.1.5 CRC ERROR BUT NO DATA ERROR

THE ABOVE THREE ERRORS ARE DETECTED WHEN THE PROGRAM IS VERIFYING THE DATA READ OFF THE DISKETTE AGAINST THE DATA THAT SHOULD HAVE BEEN READ.

UPON A NON-COMPARISON THE PROGRAM TYPES OUT THE "BYTE" NUMBER IN THE SECTOR, THE DATA READ FROM THE DISKETTE "BAD" AND THE EXPECTED DATA "GOOD".

BYTE# BAD GOOD
(THE DATA PATTERNS ARE FORMATTED AS SHOWN)

0 (TRACK ADDRESS; BITS 6 - 0)
1 (SECTOR ADDRESS; BITS 4 - 0)

BYTES 2 THROUGH 125 CONTAIN THE SELECTED "P"ATTERN.

126 (THE SUM OF ALL BYTES 0 - 125)
127 (THE NEGATIVE OF 2 TIMES BYTE 126)

THE PROGRAM DETECTS A CHECKSUM ERROR BY SUMMING ALL THE DATA READ FROM THE DISKETTE AND COMPARING THAT SUM TO 0.

AT THE END OF THE DATA ERROR TYPEOUT THE PROGRAM TYPES OUT IF THE CHECK SUM ACCUMULATED IS "GOOD"OR HAD "ERRORS".
IF BYTES 0 OR 1 HAVE DATA ERRORS THE OPERATOR MUST CHECK THE RESULTS OF THE CHECK SUM. IF IT IS ALSO BAD,THEN THERE WAS A TRUE DATA ERROR.
IF THE CEHCK SUM IS GOOD,THEN IT MIGHT BE THAT THE HEAD IS NOT OVER THE TRACK EXPECTED,AND THERE IS A POSITIONING ERROR.

3.1.6 SEEK ERROR

A SEEK ERROR IS DEFINED AS NOT A CRC AND NOT A PARITY ERROR.
A PROGRAMED RECALIBRATE IS ISSUED TO TRY TO CORRECT EACH SEEK ERROR.

3.1.7 PARITY ERROR

A PARITY ERROR RESULTS FROM AN INCORRECT TRANSFER OF A COMMAND WORD FROM THE RX11 INTERFACE TO THE RX01 MICRO-PROCESSOR CONTROLLER.

3.2 DEFINITIVE ERROR CODES

THE RX01 MICROCONTROLLER HAS DEFINED THE ERROR CODES AND MEANINGS WHICH ARE AVAILABLE TO THE PROGRAM BY ISSUING COMMAND #7 "READ THE B-CODE STATUS REGISTER".

A DEFINITIVE ERROR CODE REPRESENTS [WHERE] WITHIN A FUNCTION AN ERROR WAS DETECTED.

THE FOLLOWING ARE THE DEFINITIVE ERROR CODES AND MEANINGS:

00 - NO ERROR
10 - DRIVE 0 FAILED TO SEE HOME FROM INITIALIZE
20 - DRIVE 1 FAILED TO SEE HOME FROM INITIALIZE
30 - HOME FOUND WHEN STEPPING OUT 10 TRACKS FROM INIT
40 - TRIED TO ACCESS A TRACK GREATER THAN /7(DECIMAL)
50 - HOME WAS FOUND BEFORE DESIRED TRACK
60 - SELF DIAGNOSTIC ERROR
70 - DESIRED SECTOR NOT FOUND AFTER SAMPLING 52 HEADERS
100 - WRITE PROTECT ERROR
110 - MORE THAN 40US AND NO SEP CLOCK DETECTED
120 - A PREAMBLE COULD NOT BE FOUND
130 - PREAMBLE FOUND BUT NO ID MARO FOUND IN TIME
140 - CRC ERROR ON SUPPOSIDLY GOOD HEADER
150 - GOOD HEADER (NO CRC ERROR) BUT TRACK COMPARE ERROR
160 - IDAM NOT FOUND IN TOME
170 - DATA AM NOT FOUND IN TIME
200 - DATA CRC ERROR
210 - ALL PARITY ERRORS

3.3 UNEXPECTED OR MISSING ERROR CONDITIONS

3.3.1 MISSING DD MARK

AN ERROR WHEN THE PROGRAM WROTE DELETED DATA INFORMATION BUT NO DELETED DATA MARK WAS DETECTED WHEN THE DATA WAS READ.

3.3.2 UNEXPECTED DD MARK

AN ERROR WHEN A DELETED DATA MARK IS DETECTED BUT NO DELETED DATA WAS WRITTEN.

3.3.3 NO INTERRUPT ON DONE

THE INTERRUPT ENABLE BIT WAS SET AND THE DONE FLAG WAS SET BUT NO INTERRUPT OCCURRED.

3.3.4 UNKNOWN INTERRUPT

IF AN INTERRUPT OCCURS FROM ANY OTHER DEVICE WHILE THIS PROGRAM IS RUNNING IT WILL HALT AT THE INTERRUPT VECTOR LOCATION.

IF AN INTERRUPT OCCURS ON INTERRUPT VECTOR LOCATION 264 (RX11), AND THERE IS NO ERROR, DONE, OR ERROR STATUS CONDITION SET, THEN IT WILL BE TAGGED AS AN UNKNOWN INTERRUPT.

3.4 POWER FAILURE

THE PROGRAM TESTS FOR TWO TYPES OF POWER FAILURE, TOTAL SYSTEM POWER LOSS AND RX11 POWER LOSS RESULTING IN A RECALIBRATION OF THE DRIVES.

THE TOTAL SYSTEM POWER FAILURE IS DETECTED BY THE "SYSMAC" SUBROUTINE .$POWER. WHEN THE POWER IS DETECTED TO BE GOING DOWN, THE REGISTERS ARE SAVED. WHEN THE POWER COMES BACK UP THE REGISTERS ARE RESTORED AND THE MESSAGE "POWER" IS PRINTED. THE PROGRAM THEN AUTOMATICALLY DOES A RESTART.

LOSS OF POWER IN THE RX11 CAUSES A RECALIBRATION OF ALL DRIVES. WHEN THIS HAPPENS, THE "INIT DONE" BIT IS SET IN THE THE RXES REGISTER ALONG WITH THE NORMAL "DONE" FLAG. AT EACH INTERRUPT THE PROGRAM TESTS FOR INIT DONE. IF IT IS FOUND TRUE, THE FUNCTION WAS NOT COMPLETED AND A POWER LOSS MUST HAVE BEEN DETECTED. WHEN THIS HAPPENS THE PROGRAM TYPES OUT "RX11 POWER" AND DOES AN AUTOMATIC RESTART.

IF THERE ARE REPEATED, NOTHING ELSE HAPPENS BUT, RX11 POWER MESSAGES THE INIT DONE FLAG MIGHT BE STUCK ON.

3.5 PROGRAM HUNG

THERE ARE MANY PLACES WHERE THE PROGRAM MUST WAIT FOR AN OPERATION TO BE COMPLETED IN THE RX01. THESE ARE WAITING FOR THE "DONE" FLAG TO INDICATE A FUNCTION IS COMPLETED, OR WAITING FOR A TRANSFER REQUEST "TR" FLAG TO SEND OR RECEIVE THE NEXT BYTE OF INFORMATION. IF THE RX11 DOES NOT COME BACK WITH EITHER OR BOTH OF THESE FLAGS THE PROGRAM WOULD HANG UP.

TO INHIBIT THIS FROM HAPPENING THERE ARE TWO SUBROUTINES USED TO CHECK FOR THESE TWO FLAGS.

THE "DONECK" LOOKS FOR THE DONE FLAG AND IF IT IS NOT SEEN WITHIN A SPECIFIC TIME THE "TEST HUNG" MESSAGE IS PRINTED AND THE PROGRAM HALTS. IN REGISTER 3 (177703) IS THE RETURN ADDRESS OF THE TEST THAT IS WAITING FOR THE DONE FLAG.

"TRCK" LOOKS FOR THE "TR" FLAG. IF IT IS NOT SEEN WITHIN A
SPECIFIC TIME IT ALSO PRINTS THE "TEST HUNG" MESSAGE AND
HALTS. IN THE SP (177706) IS THE RETURN ADDRESS OF THE
TEST WAITING FOR THE TR FLAG.

THE WAITING TIME FOR THE TWO FLAGS DEPENDS UPON THE HOST PROCESSOR AS
INDICATED BELOW:

"DONE" WAIT	11/45 BIPOL	14 SEC.
	11/05 CORE	62 SEC.
"TR" WAIT	11/45 BIPOL	.44 SEC.
	11/05 CORE	2 SEC.

4.0 ERROR REPORTING

ALL ERRORS DETECTED WILL BE REPORTED IF SW13 = 0 FOR
OPERATING CONDITIONS. IF SW12 = 1 THEN ONLY 10 DATA
ERRORS FOR 1 SECTOR WILL BE REPORTED, AND A TOTAL OF
DATA ERRORS FOR THAT SECTOR WILL BE REPORTED AT THE END
OF THE SECTOR.

THE END OF PASS INDICATOR TYPED,ALSO INDICATES WEITHER ANY ERRORS
OCCURED DURRING THAT PASS. IF THERE WERE NO ERRORS THEN A "*"
IS PRINTED. IF THERE WERE ERRORS THEN A "=" IS PRINTED.
THE TOTAL ACCUMULATED ERRORS AND SYSTEMS OPERATION IS
REPORTED BY THE "ERROR DUMP" PROGRAM.

5.0 HALTS

THERE ARE VARIOUS HALT LOCATIONS THROUGHOUT THE PROGRAM.
SOME ARE THE RESULTS OF "HARD" ERRORS OTHERS ARE WAITING FOR
INTERVENTION. THEY ARE LISTED BELOW:

HALT #	TYPE	DEFINITION
HLT1:	NO ERROR	;CAUTION - LOAD MEDIUM ON UNIT 0
HLT3:	ERROR	;ERRORS FOUND ON RECAL FUNCTIONS
HLT4:	ERROR	;HARD PARITY ERRORS
HLT5:	ERROR	;NO DRIVES READY
HLT6:	ERROR	;SW15 SET ON PARITY ERROR
HLT7:	ERROR	;SW15 SET ON SEEK ERROR
HLT10:	ERROR	;SW15 SET ON CRC GENERATOR ERROR (NO DATES ERROR)
HLT11:	ERROR	;HARD CRC GENERATOR ERROR (NO DATA ERRORS)
HLT12:	ERROR	;SW15 SET ON DATA CRC ERROR
HLT13:	ERROR	;SW15 SET ON MISSING DETECTED DATA ERROR
HLT14:	ERROR	;SW15 SET ON DATA NO CRC ERROR
HLT15:	ERROR	;SW15 SET ON MISSING INTERRUPT AT DONE
HLT16:	NO ERROR	;HALT AT END OF PASS
HLT17:	NO ERROR	;HALT AT END OF ERROR DUMP
HLT20:	ERROR	;PROGRAM HUNG WAITING FOR DONE
HLT21:	ERROR	;PROGRAM HUNG WAITING FOR TR FLAG


```
                                        .NLIST CND,MD,MC
                                        .LIST ME
                                        .ENABL ABS,AMA
                                        .MCALL .HEADER,.EQUAT,.SETUP,.STYPE,.STYPOCT,.$READ
```

```
                                .MCALL .$POWER,STARS,,$DB2D,,$SB2D,,$SAVE,,$TRAP,GETSWR
                        
                        
                        
                        .TITLE  MAINDEC-11-DZRXA-E
                        ;*COPYRIGHT (C) MARCH 21,1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY DAVID L ADAMS
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C0),MAR 21, 1976.
                        ;*
000001                  $TN=1
160000                  $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
                        
                        
                                ;COPYRIGHT (C) 1975,1976
                                ;THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY
                                ;ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH
                                ;THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
                                ;SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED
                                ;OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON
                                ;EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO
                                ;THESE LICENSE TERMS. TITLE TO OWNERSHIP OF THE
                                ;SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.
                                ;
                                ;THE INFOMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
                                ;WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
                                ;BY DIGITAL EQUIPMENT CORPORATION.
                                ;
                                ;DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
                                ;OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY
                                ;DEC.
                        
                        
                        .SBTTL  BASIC DEFINITIONS
                        
001200                  ;*INITIAL ADDRESS OF THE STACK POINTER *** 1200 ***
                        STACK=  1200
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
                        
                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
```



```
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                        
                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  R7,PC           ;;PROGRAM COUNTER
                        
                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7
                        
                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0
                        
                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
```

```
                100000                  BIT15=  100000
                040000                  BIT14=  40000
                020000                  BIT13=  20000
                010000                  BIT12=  10000
                004000                  BIT11=  4000
                002000                  BIT10=  2000
                001000                  BIT09=  1000
                000400                  BIT08=  400
                000200                  BIT07=  200
                000100                  BIT06=  100
                000040                  BIT05=  40
                000020                  BIT04=  20
                000010                  BIT03=  10
                000004                  BIT02=  4
                000002                  BIT01=  2
                000001                  BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

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


                                                ;SPECIAL EQUATES


                000013                          RD0STAT =13
                000033                          RD1STAT =33
                000017                          RUER    =17
                000040                          DONEBIT =40
                000101                          FBIE    =101
                000103                          EBIE    =103
                000105                          WRTIE   =105
                000107                          RDIE    =107
                000115                          WTDDIE  =115
                040001                          RECAL   =40001
                000200                          PR4     =200
```



```
                000340                          PR7     =340
                000000                          OPEN    =0

                000000                          .=0
000000  000000  000000                                  .WORD 0,0

                000024                          .=24
000024  013764                                          $PWRDN
000026  000340                                          340

                000034                          .=34
000034  013716                                          $TRAP                   ;ADDRESS OF TRAP SERVICE
000036  000340                                          340

                000046                          .=46
000046  011130                                          LOGICAL                 ;ACT11 EOP HOOK

                000052                          .=52
000052  000000                                          .WORD 0

                000174                          .=174
000174  000000                  DISPREG:               0
000176  000000                  SWREG:                 0


                                        ;;**************************************************************

                                                ;STARTING ADDRESSES
                                                ;
                                                ;INITIAL START  =200    /CLEARS ALL ERROR LOGS,RESETS COUNTERS,AND ALLOWS FOR
                                                ;                       /SELECTION OF DRIVES AND TEST CONDITIONS
                                                ;
                                                ;RESTART        =202    /USES PREVIOUS INITIAL START DRIVES AND TEST
                                                ;                       /SELECTION AND CONTINUES TO ACCUMULATE STATISTICAL DATA
                                                ;
                                                ;ERROR REPORT   =204    /PRINTS OUT ALL RUN AND ERROR CONDITIONS
                                                ;                       /ACUMULATED OVER THE RUN OF THE PROGRAM.
                                                ;


                000200                          .=200
000200  000402                                          BR 1$
000202  000403                                          BR 2$
000204  000404                                          BR 3$
000206  000137  001220          1$:                     JMP @#200               ;OPERATOR SELECTED CONDITIONS
000212  000137  002700          2$:                     JMP RESTART             ;RESTART PROGRAM WITH PREVIOUS CONDITIONS
000216  000137  021062          3$:                     JMP ERDUMP              ;STATISTICAL ERROR PRINT OUT


                                        ;;**************************************************************

                                                ;THE FOLLOWING LOCATIONS "OD","ID","FIRST",AND "LAST"
                                                ;MAY BE CHANGED BY THE OPERATOR MANUALLY HOWEVER FOLLOWING THESE RESTRICTIONS.
                                                ;(IF THESE LOCATIONS ARE LEFT CLEARED,MAX AND MIN VALUES ARE ASSUMED)
                                                ;
                                                ;       1. DEFINITATIONS:
```

```
                                        ;               OD=ADDRESS OF TRACK AT OUTTER DIAMETER (MIN VALUE=0)
                                        ;               ID=ADDRESS OF TRACK AT INNER DIAMETER (MAX VALUE=114 (OCTAL))
                                        ;               FIRST=ADDRESS OF FIRST SECTOR OF TRACK (MIN VALUE=1)
                                        ;               LAST=ADDRESS OF LAST SECTOR OF TRACK (MAX VALUE=32 (OCTAL))
                                        ;
                                        ;       2. LOCATIONS:
                                        ;               TRACKS  LOC, 1200 BITS 14-----8 6-----0
                                        ;                                              ID      OD
                                        ;
                                        ;               SECTORS LOC, 1202 BITS 12-----8 4-----0
                                        ;                                             LAST    FIRST
                                        ;
                                        ;       3. RESTRICTIONS:
                                        ;               THE CONTENTS OF "OD" MUST BE <= THE CONTENTS OF "ID"
                                        ;               THE CONTENTS OF FIRST MUST BE <= THE CONTENTS OF "LAST"


        001200                          .=1200
001200  000000                  OD:            0
        001201                  ID=OD+1
001202  000000                  FIRST:         0
        001203                  LAST=FIRST+1


                                ;;*************************************************************

                                        ;THE NEXT WORD IS THE LOCATION OF THE "INTERRUPT VECTOR" ADDRESS
                                        ;IF THE HARDWARE IS JUMPERED FOR OTHER THAN STANDARD (264) VECTOR
                                        ;ADDRESS, THEN LOAD INTO THIS LOCATION THE NEW VECTOR ADDRESS.
                                        ;
                                        ;IF THIS ADDRESS IS INCORRECT (DOES NOT MATCH THE HARDWARE) THE
                                        ;PROGRAM WILL HALT AT THE VECTOR ADDRESS THE HARDWARE IS JUMPERED
                                        ;TO, AS ALL OTHER VECTOR ADDRESSES WILL HAVE HALTS IN THEM.
                                        ;
                                        ;*****************************NOTE*****************************
                                        ;
                                        ;THE INTERRUPT VECTOR ADDRESS CAN NOT BE SET FOR ADDRESSES 200
                                        ;THRU 243, AS THESE ADDRESS ARE USED BY THE PROGRAM
                                        ;
                                        ;*************************************************************
001204  000264                  INTVEC: 264

                                ;;*************************************************************
                                        ;THE FOLLOWING TWO WORDS CONTAIN THE DEVICE CODES FOR
                                        ;THE RX11 INTERFACE REGISTERS
                                        ;       RXCS = COMMAND STATUS REGISTER
                                        ;       RXDB = DATA BUFFER REGISTER (USED AT VARIOUS TIMES FOR:)
                                        ;               (RXTA = TRACK ADDRESS)
                                        ;               (RXSA = SECTOR ADDRESS)
                                        ;               (RXES = ERROR STATUS)
                                        ;IF THE RX11 SYSTEM UNDER TEST IS JUMPERED FOR REGISTER
                                        ;ADDRESSES OTHER THAN STANDARD: RXCS = 177170
                                        ;                              RXDB = 177172
```



```
                                        ;PLACE IN THESE TWO WORDS THE CORRESPONDING NEW ADDRESSES FOR
                                        ;THE REGISTERS.  THE PROGRAM WILL TYPE OUT THE REGISTER
                                        ;ADDRESSES FOR VERIFICATION BY THE OPERATOR.  IF THE ADDRESSES
                                        ;ARE INCORRECT (I.E. - DON'T MATCH THE HARDWARE) AND THE PROGRAM
                                        ;IS RUN WITH THE INCORRECT ADDRESSES, A 'HUNG' CONDITION WILL BE
                                        ;REPORTED AND THE PROGRAM WILL 'HALT' AS THE RX11 WILL NOT BE
                                        ;ABLE TO RESPOND TO COMMANDS.
                                ;;*************************************************************
001206  177170                  RXCS:          177170
001210  177172                  RXDB:          177172

                                        ;BIT ASSIGNMENT IN THE RXCS REGISTER.
                                        ;
                                        ;KEY: R = READ ONLY BIT
                                        ;     W = WRITE ONLY BIT
                                        ;
                                        ;       15 - R - ERROR FLAG
                                        ;       14 - W - INITIALIZE (RECALIBRATE)
                                        ;       13 -
                                        ;       TO      NOT USED
                                        ;        8 -
                                        ;        7 - R - TRANSFER REQUEST (TR) FLAG
                                        ;        6 - R/W - INTERUPT ENABLE
                                        ;        5 - R - DONE FLAG
                                        ;        4 - W - UNIT SELECT
                                        ;        3 - W - FUNCTION
                                        ;        2 - W - FUNCTION
                                        ;        1 - W - FUNCTION
                                        ;        0 - W - GO
                                        ;
                                        ;FUNCTION CODES:
                                        ;
                                        ;       0 + GO = FILL BUFFER
                                        ;       2 + GO = EMPTY BUFFER
                                        ;       4 + GO = WRITE SECTOR
                                        ;       6 + GO = READ SECTOR
                                        ;       10 (NOT USED)
                                        ;       12 + GO = READ STATUS "A"
                                        ;       14 + GO = WRITE DELETED DATA
                                        ;       16 + GO = READ STATUS "B" (CODES)
                                        ;
                                        ;
                                        ;THE FOLLOWING BIT ASSIGNMENTS REPRESENTES THE STATUS AT THE END OF A
                                        ;FUNCTION (EXCEPT FUNCTION "READ STATUS B") DISPLAYED IN THE
                                        ;RX DATA BUFFER (RXDB).
                                        ;
                                        ;       15 -
                                        ;       TO    NOT USED
                                        ;       8 -
                                        ;       7 - SELECTED DRIVE READY *
                                        ;       6 - DELETED DATA
                                        ;       5 -
                                        ;       4 -
                                        ;       3 - WRITE PROTECT ERROR (WHEN AVAILABLE)
                                        ;       2 - INITIALIZE DONE **
```

```
                                        ;       1 = PARITY ERROR
                                        ;       0 = CRC ERROR
                                        ;
                                        ;*  VISIBLE ONLY IF THE FUNCTION WAS A #12 "READ STATUS A"
                                        ;
                                        ;**  VISIBLE ONLY AFTER AN INITIALIZE (KEY) OR (PROGRAMMED) WAS ISSUED.

001212  000000          DTESTP: .WORD 0                 ;TEST SELECTION WORD
001214  177570          SWR:    .WORD USWR              ;ADDRESS OF SWITCH REGISTER
001216  177570          DISPLAY: .WORD UDISP            ; ADDRESS OF DISPLAY REGISTER

                        ;;*************************************************************

                        ;START OF OPERATOR SELECTABLE TEST,DATA PATTERNS,AND DRIVE UNIT CONDITIONS
                        ;SET THE TEST CONDITIONS WANTED (OR LEAVE 0) IN LOCATION CALLED
                        ;"DTESTP" (LOC. 1212), BEFORE STARTING THE PROGRAM. WHEN THE PROGRAM
                        ;STARTS IT WILL TYPE OUT THE TEST CONDITIONS IT IS OPERATING UNDER.
                        ;

                        ;SWITCH REGISTER BITS
                        ;
                        ; 15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  0
                        ; U1  U0      NOT USED    D  P  P  P  T  T  T  S  S  S
                        ;
                        ; U1    SELECT DRIVE UNIT 1
                        ; U0    SELECT DRIVE UNIT 0
                        ;IF NEITHER DRIVE IS SPECIFIED,PROGRAM WILL SELECT ALL DRIVES
                        ;THAT ARE READY
                        ;
                        ; D =     DELETED DATA FUNCTIONS
                        ;
                        ;IF THIS SWITCH IS ON ALL READ AND WRITE FUNCTIONS WILL BE IN THE
                        ;DELETED DATA MODE
                        ;
                        ;
                        ; P =     DATA PATTERN SELECTION
                        ;       0       DO PATTERN 7 (RANDOM)
                        ;       1       ZEROS
                        ;       2       ONES
                        ;       3       FLOATING ZERO
                        ;       4       FLOATING ONE
                        ;       5       125
                        ;       6       314
                        ;       7       RANDOM
                        ;
                        ; T =     FUNCTIONAL TESTS
                        ;       0       DO TEST 7 (WRITE/READ/READ CHECK)
                        ;       1       WRITE ONLY
                        ;       2       WRITE/READ
                        ;       3       WRITE/READ CHECK
                        ;       4       READ CHECK ONLY
                        ;       5       READ ONLY
                        ;       6       WRITE/READ CHECK ON ALTERNATE DRIVES
                        ;       7       WRITE/READ/READ CHECK *
                        ;
                        ; * NOTE: THE FIRST HALF OF TEST 7 FORCES THE TRACK SEQUENCE
```



```
                        ;TO INCRIMENT UP THROUGH ALL THE TRACKS DOING WRITE / READ CHECK FUNCTIONS.
                        ;THIS VERIFIES THAT ALL THE TRACKS ARE ACCESSABLE.
                        ;THE SECOND HALF OF THE PASS WILL USE THE SEQUENCE SELECTED BY THE
                        ;OPERATOR AS INDICATED BELOW,AND ONLY READ CHECK THE DATA JUST WRITTEN.
                        ;THIS VERIFIES THAT THE DATA CAN BE READ FROM A TRACK,AFTER THE HEAD HAS BEEN
                        ;MOVED AWAY FROM AND BACK TO THAT TRACK, AT THE COMPLETION OF THE PASS
                        ;THE DELETED DATA BIT IN TEST CONDITIONS IS COMPLEMENTED AND THE NEXT
                        ;PASS WILL BE RUN UNDER THIS NEW CONDITION.
                        ;
                        ; S =     TRACK SEQUENCING
                        ;       0       DO SEQUENCE 7 (RANDOM)
                        ;       1       INCREMENT
                        ;       2       DECREMENT
                        ;       3       INCREMENT/DECREMENT
                        ;       4       BOUNCE
                        ;       5       DECREASING BOUNCE
                        ;       6       STROBE
                        ;       7       RANDOM


                        ;;*************************************************************

                        ;SET THE OPERATING CONDITIONS IN THE SWITCH REGISTER (HARDWARE),
                        ;OR THE SOFTWARE SWITCH REGISTER (LOC. 176) BEFORE STARTING THE PROGRAM.
                        ;
                        ;SWITCHES DO THE FOLLOWING WHEN SET TO "1".
                        ;
                        ;       SW 15   =HALT ON ERROR
                        ;       SW 14   =HALT AT END OF TEST
                        ;       SW 13   =DON'T PRINT ERROR MESSAGE
                        ;       SW 12   =TYPE ONLY 10 DATA ERRORS
                        ;       SW 11   =NO RETRY ON ERROR, LOG HARD ERROR
                        ;
                        ;       SW  8   =NO RECALIBRATION ON SEEK ERRORS
                        ;
                        ;       SW15 = SW0      =SELECT SOFTWARE SWITCH REGISTER


                        ;;*************************************************************


001220  000005          SA200:  RESET                   ;INITIALIZE THE RX01
001222  012737  177570  001214          MOV #177570,SWR         ;RESET TO HARDWARE SWR.
001230  012706  001200                  MOV #STACK,SP
001234  012746  000340                  MOV #PR7,-(SP)
001240  012746  001246                  MOV #2$,-(SP)
001244  000002                          RTI
001246  104400  016554          2$:     TYPE ,MREV              ;PRINT NAME AND REVISION
001252  012737  001272  000004          MOV #3$,4               ;SET TIME OUT VECTOR
001260  022777  177777  177726          CMP #177777,@SWR        ;IS SOFTWARE SWR SELECTED
001266  001402                          BEQ 4$                  ;YES, INSERT IT'S ADDRESS
001270  000423                          BR 5$                   ;BRANCH IF NO TIMEOUT TRAP OCCURS
001272  022626                  3$:     CMP (SP)+,(SP)+         ;RESTORE STACK AFTER TRAP
001274  012737  000176  001214  4$:     MOV #SWREG,SWR          ;POINT TO SOFTWARE SWITCH REG.
001302  012737  000174  001216          MOV #DISPREG,DISPLAY    ;POINT TO SOFTWARE DISP. REG.
```

```
                                        .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
001310  005737  000042                        TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
001314  001006                                BNE     64$             ;;BRANCH IF YES
001316  023727  001214  000176                CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
001324  001005                                BNE     65$             ;;BRANCH IF NO
001326  104404                                GTSWR                   ;;GET SOFT-SWR SETTINGS
001330  000403                                BR      65$
001332  112737  000001  013714  64$:   MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
001340                                65$:
001340  012737  000006  000004  5$:            MOV #6,4                ;RESTORE ERROR VECTOR
001346  013703  001204                        MOV INTVEC,R3           ;SET INTERUPT VECTOR LOCATIONS
001352  012723  007250                        MOV #INTSERV,(R3)+
001356  012713  000340                        MOV #340,(R3)
001362  012737  001234  012210                MOV #001234,RAN1        ;INITIALIZE CONSTANTS IN RANDOM NUMBER GENERATOR
001370  012737  000765  012212                MOV #000765,RAN2
001376  012705  017024                        MOV #PARLOG,R5          ;SET UP TO CLEAR ALL COUNTERS
001402  005025                        1$:     CLR (R5)+               ;OF ERRORS,ACCESSES,AND PASSES
001404  022705  021044                        CMP #PASCNTR+2,R5       ;HAVE ALL COUNTERS BEEN CLEARED
001410  001374                                BNE 1$
001412  012705  017004                START1: MOV #RUNRETRY,R5        ;SET ALL RETRY COUNTERS TO 10
001416  012725  000012                3$:     MOV #10.,(R5)+
001422  022705  017024                        CMP #SHETRY+2,R5
001426  001373                                BNE 3$
001430  104400  015167                        TYPE ,MRXCS             ;TYPE OUT DEVICE REGISTER ADDRESSES
001434  013746  001206                MOV     RXCS,-(SP)      ;;SAVE RXCS FOR TYPEOUT
001440  104402                        TYPOS                   ;;GO TYPE--OCTAL ASCII
001442     006                        .BYTE   6               ;;TYPE 6 DIGITS
001443     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
001444  104400  015177                        TYPE ,MRXDB
001450  013746  001210                MOV     RXDB,-(SP)      ;;SAVE RXDB FOR TYPEOUT
001454  104402                        TYPOS                   ;;GO TYPE--OCTAL ASCII
001456     006                        .BYTE   6               ;;TYPE 6 DIGITS
001457     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
001460  104400  015150                        TYPE ,MINTVEC           ;AND VECTOR ADDRESS
001464  013746  001204                MOV     INTVEC,-(SP)    ;;SAVE INTVEC FOR TYPEOUT
001470  104402                        TYPOS                   ;;GO TYPE--OCTAL ASCII
001472     003                        .BYTE   3               ;;TYPE 3 DIGIT(S)
001473     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
001474  004737  007122                        JSR PC,DONECK           ;WAIT FOR DONE BIT AFTER RECAL
001500  032777  000004  177502                BIT #BIT2,@RXDB         ;IS INIT DONE SET
001506  001002                                BNE 1$                  ;YES SKIP NEXT INSTRUCTION
001510  104400  015535                        TYPE ,MINIT1            ;NO,PRINT NO INIT DONE ERROR
001514  005777  177466                1$:     TST @RXCS               ;WAS THERE AN ERROR ON START RECAL
001520  100002                                BPL TESTDR              ;IF NOT CONTINUE
001522  004737  002554                        JSR PC,HOME             ;YES,DO RECAL AGAIN
001526  012737  000012  017014        TESTDR: MOV #10.,P2RETRY        ;SET THE PARITY RETRY COUNTER
001534  005037  010704                        CLR UNITSEL             ;CLEAR UNIT SELECTION WORD
001540  105777  177444                        TSTB @RXDB              ;IS DRY BIT SET FOR UNIT 0 AFTER RECAL
001544  100003                                BPL 2$                  ;IF NOT SKIP NEXT INSTRUCTION
001546  052737  000200  010704                BIS #BIT7,UNITSEL       ;YES,SET UNIT 0 READY BIT
001554  032737  140000  001212  2$:            BIT #140000,DTESTP      ;WERE ANY DRIVES SPECIFIED
001562  001445                                BEQ NOSEL               ;NO,SEE IF UNIT 1 IS READY
001564  005737  001212                        TST DTESTP              ;WAS UNIT 1 SELECTED
001570  100044                                BPL OPCOND              ;NO,IT MUST BE UNIT 0
001572  004737  001614                        JSR PC,DRY1             ;SEE IF UNIT 1 IS READY
001576  032737  040000  001212                BIT #40000,DTESTP       ;WAS UNIT 0 SELECTED
```



```
001604  001036                                BNE OPCOND              ;YES,CONTINUE
001606  105037  010704                        CLRB UNITSEL            ;NO,CLEAR UNIT 0 READY BIT
001612  000433                                BR OPCOND
                                        
001614  012777  000033  177364  DRY1:         MOV #RD1STAT,@RXCS      ;READ STATUS OF UNIT 1
001622  004737  007122                        JSR PC,DONECK           ;WAIT FOR DONE BIT
001626  132777  000002  177354                BITB #2,@RXDB           ;IS PARITY ERROR BIT SET
001634  001411                                BEQ 2$                  ;NO,CONTINUE
001636  004737  002046                        JSR PC,STATER           ;YES,GO HANDLE PARITY ERROR
001642  000764                                BR DRY1                 ;REISSUE FUNCTION
001644  032777  000004  177336                BIT #BIT2,@RXDB         ;INIT DONE SHOULD NOT BE SET
001652  001402                                BEQ 2$
001654  104400  015567                        TYPE ,MINIT2            ;INIT DONE WAS SET PRINT ERROR
001660  105777  177324                2$:     TSTB @RXDB              ;IS UNIT 1 DRY BIT SET
001664  100003                                BPL 3$                  ;IF NOT SKIP NEXT INSTRUCTION
001666  052737  100000  010704                BIS #BIT15,UNITSEL      ;YES,SET UNIT 1 READY BIT
001674  000207                        3$:     RTS PC
                                        
001676  004737  001614        NOSEL:  JSR PC,DRY1             ;TEST UNIT 1 FOR DRY SET
001702  105737  010704        OPCOND: TSTB UNITSEL            ;IS UNIT 0 SELECTED
001706  100016                                BPL 1$                  ;NO
                                        
                                        ;TEST ACT11 LOAD MEDIA INDICATOR
                                        
001710  123727  000041  000010                CMPB @#41,#10           ;DOES LOCATION 41 CONTAIN THE NUMBER 10?
001716  001012                                BNE 1$                  ;BRANCH IF NOT
001720  005737  000042                        TST @#42                ;CHECK FOR RXDP OPERATION
001724  001404                                BEQ 2$
001726  042737  000200  010704                BIC #200,UNITSEL        ;IN CHAIN MODE, DESELECT UNIT 0
001734  000403                                BR 1$                   ;AND DO NOT HALT
001736  104400  014643                2$:     TYPE,   D0LOAD          ;INFORM USER TO REMOVE LOAD MEDIUM
                                                                        ;FROM UNIT 0 AND REPLACE WITH
                                                                        ;A 'SCRATCH' DISKETTE IF HE
001742  000000                                HALT                    ;WISHES TO TEST UNIT 0
001744  000241                        1$:     CLC                     ;CLEAR C BIT FOR ROR'S THAT FOLLOW
001746  013737  001212  011370                MOV DTESTP,SEQUEN       ;PUT INITIAL CONDITIONS IN SEQUENCE
001754  042737  177770  011370                BIC #177770,SEQUEN      ;CLEAR ALL BUT SEQUENCE BITS
001762  013737  001212  002140                MOV DTESTP,TEST         ;PUT INITIAL CONDITIONS IN TEST
001770  042737  177707  002140                BIC #177707,TEST        ;CLEAR ALL BUT TEST BITS
001776  006037  002140                        ROR TEST                ;RIGHT JUSTIFY TEST BITS
002002  006037  002140                        ROR TEST
002006  006037  002140                        ROR TEST
002012  013737  001212  010420                MOV DTESTP,PAT          ;PUT INITIAL CONDITIONS IN PATTERN
002020  006137  010420                        ROL PAT                 ;RIGHT JUSTIFY PATTERN BITS
002024  006137  010420                        ROL PAT
002030  000337  010420                        SWAB PAT
002034  042737  177770  010420                BIC #177770,PAT         ;CLEAR ALL BUT PATTERN BITS
002042  000137  002722                        JMP XSTART
                                        
002046  005237  017024        STATER: INC PARLOG              ;INC PARITY ERROR COUNTER
002052  104400  015167                        TYPE ,MRXCS             ;PRINT THE RXCS CONTENTS
002056  017746  177124                MOV     @RXCS,-(SP)     ;;SAVE @RXCS FOR TYPEOUT
002062  104402                        TYPOS                   ;;GO TYPE--OCTAL ASCII
002064     003                        .BYTE   3               ;;TYPE 3 DIGIT(S)
002065     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
002066  005337  017014                        DEC P2RETRY             ;HAVE 10 ERRORS OCCURED
```

```
561  002072  001405                          BEQ 1$                  ;YES,LOG HARD ERROR
562  002074  104400  016017                  TYPE ,MPAR              ;NO,TYPE PARITY ERROR
563  002100  104400  015052                  TYPE ,MCRLF
564  002104  000207                          RTS PC                  ;RETURN
565  002106  005237  017026          1$:     INC HPARLOG             ;INC HARD PARITY ERROR LOG
566  002112  104400  015755                  TYPE ,MUNREC            ;PRINT UNRECOVERABLE PARITY ERROR
567  002116  104400  016017                  TYPE ,MPAR
568  002122  104400  015052                  TYPE ,MCRLF
569  002126  104400  015702                  TYPE ,MHALT4
570  002132  000000          HLT4:           HALT                    ;HALT BECAUSE OF HARD PARITY ERROR
571  002134  000137  001220                  JMP SA200               ;IF CONTINUE IS PRESSED GO TO START
572  002140  000000          TEST:           0
573
574  002142  005002          TSTLIMITS:      CLR R2                  ;MESSAGE FLAG
575  002144  005737  001200                  TST OD                  ;TEST FOR NO SELECTION OF TRACKS
576  002150  001441                          BEQ 1$                  ;IT WILL BE STANDARD LIMITS
577  002152  123727  001201  000114          CMPB ID,#114            ;TEST FOR MAXIMUM ID LIMIT
578  002160  101112                          BHI 14$
579  002162  001431                          BEQ 2$
580  002164  123737  001200  001201  3$:     CMPB OD,ID              ;OD MUST BE EQUAL OR LESS THAN ID
581  002172  101105                          BHI 14$
582  002174  004737  002432                  JSR PC,TSTMSG           ;HAS INITIAL MESSAGE BEEN TYPED
583  002200  104400  015510                  TYPE ,MOD               ;TYPE BOTH ID AND OD LIMITS
584  002204  113737  001200  002216          MOVB OD,7$
585  002212  004537  014446                  JSR R5,SGLDEC           ;TYPE SINGLE DECIMAL WORD
586  002216  000000          7$:             OPEN
587  002220  104400  015047                  TYPE ,DBLSP
588  002224  104400  015514                  TYPE ,MID
589  002230  113737  001201  002242          MOVB ID,10$
590  002236  004537  014446                  JSR R5,SGLDEC
591  002242  000000          10$:            OPEN
592  002244  000403                          BR 1$
593  002246  105737  001200          2$:     TSTB OD                 ;IS OD = 0
594  002252  001344                          BNE 5$                  ;NO,GO PRINT IT
595  002254  005737  001202          1$:     TST FIRST               ;TEST FOR NO SELECTION OF SECTORS
596  002260  001451                          BEQ 13$                 ;IT WILL BE STANDARD LIMITS
597  002262  123727  001203  000032          CMPB LAST,#32           ;TEST FOR NORMAL LAST SECTOR LIMIT
598  002270  101053                          BHI 15$
599  002272  001440                          BEQ 4$
600  002274  105737  001202          5$:     TSTB FIRST              ;THERE IS NO 0 SECTOR
601  002300  001447                          BEQ 15$
602  002302  123737  001202  001203          CMPB FIRST,LAST         ;FIRST MUST BE EQUAL TO OR LESS
603  002310  101043                          BHI 15$                 ;THAN LAST
604  002312  004737  002432                  JSR PC,TSTMSG
605  002316  104400  015047                  TYPE ,DBLSP
606  002322  104400  015520                  TYPE ,MFIRST            ;TYPE BOTH FIRST AND LAST LIMITS
607  002326  113737  001202  002340          MOVB FIRST,11$
608  002334  004537  014446                  JSR R5,SGLDEC
609  002340  000000          11$:            OPEN
610  002342  104400  015047                  TYPE ,DBLSP
611  002346  104400  015527                  TYPE ,MLAST
612  002352  113737  001203  002364          MOVB LAST,12$
613  002360  004537  014446                  JSR R5,SGLDEC
614  002364  000000          12$:            OPEN
615  002366  104400  015240                  TYPE ,DBLLF
616  002372  000404                          BR 15$
```



```
617  002374  123727  001202  000001  4$:     CMPB FIRST,#1           ;IS FIRST SET TO 1
618  002402  001334                          BNE 5$                  ;NO, GO PRINT IT
619  002404  000207          13$:            RTS PC
620
621  002406  104400  016363          14$:    TYPE ,OD2BIG            ;PRINT TRACK LIMITS OUT OF RANGE
622  002412  005037  001200                  CLR OD
623  002416  000716                          BR 1$
624  002420  104400  016455          15$:    TYPE ,S2BIG             ;PRINT SECTOR LIMITS OUT OF RANGE
625  002424  005037  001202                  CLR FIRST
626  002430  000765                          BR 15$
627
628  002432  005702          TSTMSG:         TST R2                  ;TEST MESSAGE FLAG
629  002434  001005                          BNE 1$
630  002436  104400  015446                  TYPE ,MNONSTD           ;MESSAGE HEADING HAD NOT BEEN TYPED
631  002442  104400  015052                  TYPE ,MCRLF
632  002446  005202                          INC R2                  ;SET THE MESSAGE FLAG
633  002450  000207          1$:             RTS PC
634
635  002452  104400  015052          ICOND:  TYPE ,MCRLF
636  002456  104400  015417                  TYPE ,MICON             ;TYPE INITIAL CONDITIONS
637  002462  105737  010704                  TSTB UNITSEL            ;TEST FOR DRIVE SELECTION
638  002466  100002                          BPL 1$
639  002470  104400  015106                  TYPE ,MUNIT0            ;TYPE UNIT 0 SELECTED
640  002474  005737  010704          1$:     TST UNITSEL
641  002500  100002                          BPL 2$
642  002502  104400  015116                  TYPE ,MUNIT1            ;TYPE UNIT 1 SELECTED
643  002506  104400  015040          2$:     TYPE ,TAB
644  002512  004737  010012                  JSR PC,TYPSEL           ;TYPE TEST SELECTIONS
645  002516  104400  015052                  TYPE ,MCRLF
646  002522  005737  010704                  TST UNITSEL             ;WERE ANY DRIVES READY
647  002526  001005                          BNE CONT12              ;YES,CONTINUE
648  002530  104400  015016                  TYPE ,MNODRY            ;NO,TYPE NO DRIVES MESSAGE
649  002534  000000          HLT5:           HALT                    ;NO DRIVES CAN'T CONTINUE
650  002536  000137  001220                  JMP SA200               ;IF CONTINUE IS PRESSED RESTART
651  002542  004737  002142          CONT12: JSR PC,TSTLIMITS        ;TYPE NONSTANDARD TRK AND SEC LIMITS
652  002546  000207                          RTS PC
653
654  002550  104405          HOME:           CKSWR
655  002552  032777  000400  176434          BIT #SWB,@SWR           ;TEST THE NO RECAL SWITCH
656  002560  001046                          BNE RTN                 ;RETURN IF THE SWITCH IS SET
657  002562  012737  000012  017014          MOV #10.,P2RETRY        ;SET UP THE PARITY RETRY COUNTER
658  002570  012737  000012  017010          MOV #10.,UDRETRY        ;USE THE UD RETRY COUNTER FOR RECAL RETRIES
659  002576  012777  040001  176402  2$:     MOV #RECAL,@RXCS        ;ISSUE RECAL FUNCTION
660  002604  004737  007122                  JSR PC,DONECK           ;WAIT FOR DONE FLAG
661  002610  032777  000002  176372          BIT #2,@RXDB            ;WAS THERE A PARITY ERROR
662  002616  001403                          BEQ 1$                  ;NO,CHECK FOR ANY OTHER ERROR
663  002620  004737  002046                  JSR PC,STATER           ;YES,GO REPORT IT
664  002624  000764                          BR 2$                   ;RETRY RECAL
665  002626  005777  176354          1$:     TST @RXCS               ;IS THE ERROR FLAG SET
666  002632  100016                          BPL XHOME               ;IF NOT RETURN
667  002634  004737  010200                  JSR PC,RDCODE           ;YES,PRINT STATUS REGISTERS
668  002640  123727  010072  000040          CMPB BSTAT,#40          ;IS THE B CODE LESS THAN CODE 40
669                                           ;(RECAL CODES ARE 10,20,AND 30)
670  002646  002010                          BGE XHOME               ;IF NOT RETURN
671  002650  005337  017010                  DEC UDRETRY             ;HAVE 10 ERRORS OCCURED
672  002654  001350                          BNE 2$                  ;NO,RETRY
```

```
673  002656  104400  015713                          TYPE ,MHALT3            ;YES,CAN'T CONTINUE AS RX11 WILL NOT RECAL
674  002662  000000                        HLT3:       HALT
675  002664  000137  001220                          JMP SA200               ;IF CONT. SWITCH IS PRESSED GO TO START
676  002670  012737  000001  011362        XHOME:      MOV #1,PRESTRK          ;RESET THE PRESENT TRACK TO TRACK 1
677  002676  000207                        RTN:        RTS PC                  ;RETURN
678
679                                          ;;**************************************************************
680
681                                                  ;DECODE TEST BITS OF INITIAL CONDITIONS
682                                                  ;
683                                                  ;BITS 3,4,AND 5 OF THE INITIAL SWR SELECTED THE TEST WHICH
684                                                  ;IS TO BE PREFORMED. THEY ARE AS FOLLOWS:
685                                                  ;
686                                                  ;   BITS                TESTS
687                                                  ; 5  4  3
688                                                  ; 0  0  0               NO TEST SELECTED (DEFAULT TO TEST 7)
689                                                  ; 0  0  1               WRITE ONLY
690                                                  ; 0  1  0               WRITE FOLLOWED BY READ
691                                                  ; 0  1  1               WRITE FOLLOWED BY READ AND VERIFY
692                                                  ; 1  0  0               READ AND VERIFY ONLY
693                                                  ; 1  0  1               READ ONLY (CRC CHECK)
694                                                  ; 1  1  0               READ AND VERIFY DELETED DATA
695                                                  ; 1  1  1               WRITE FOLLOWED BY READ FOLLOWED BY ANOTHER
696                                                  ;                       READ AND VERIFY
697
698                                          ;;**************************************************************
699
700  002700  012706  001200        RESTART:    MOV #STACK,SP           ;RESET THE STACK POINTER
701  002704  012746  000340                    MOV #PR7,-(SP)          ;AND INTERUPT LEVEL
702  002710  012746  002716                    MOV #1$,-(SP)
703  002714  000002                            RTI
704  002716  005237  021040        1$:         INC RESTCNTR            ;INC RESTART COUNTER
705  002722  004737  002452        XSTART:     JSR PC,ICOND            ;TYPE OUT INITIAL CONDITIONS
706  002726  005037  011142        TESTSEL:    CLR CHARCT              ;CLEAR EOP CHARACTER COUNTER
707  002732  012737  000012  017004            MOV #10.,RDRETRY        ;SET UP READ AND WRITE RETRY COUNTERS
708  002740  012737  000012  017006            MOV #10.,WTRETRY
709  002746  005037  004124                    CLR ERWRT               ;CLEAR WRITE CAUSED BY ERROR FLAG
710  002752  005037  004122                    CLR RDAFTWT             ;CLEAR READ AFTER WRITE FLAG
711  002756  142737  000377  003014            BICB #377,@#BRONTEST    ;CLEAR OUT BRANCH OFFSET
712  002764  005737  002140                    TST TEST                ;IF NO TEST SPECIFIED FORCE TEST 7
713  002770  001003                            BNE 1$
714  002772  012737  000007  002140            MOV #7,TEST
715  003000  013704  002140        1$:         MOV TEST,R4
716  003004  005304                            DEC R4                  ;ADJUST TEST BITS FOR CURRECT
717  003006  006304                            ASL R4                  ;BRANCH OFFSET
718  003010  150437  003014                    BISB R4,@#BRONTEST      ;INSERT OFFSET TO BRANCH INST.
719  003014  000777        BRONTEST:           BR .                    ;BRANCH BY TEST OFFSET
720  003016  000137  003052                    JMP WRTONLY             ;WRITE ONLY FUNCTION
721  003022  000137  004130                    JMP WRTRD               ;WRITE THEN READ FUNCTION
722  003026  000137  005456                    JMP WTRDCK              ;WRITE THEN READ VERIFY
723  003032  000137  006444                    JMP RDCHK               ;READ VERIFY
724  003036  000137  006504                    JMP RDONLY              ;READ ONLY FUNCTION
725  003042  000137  006544                    JMP DRVSWP              ;WRITE,AND READ VERIFY ON ALTERNATING DRIVES
726  003046  000137  006630                    JMP TEST7               ;WRITE,READ,FOLLOWED BY READ VERIFY
727
728
```



```
729
730                                          ;;**************************************************************
731
732                                                  ;THIS IS A WRITE ONLY FUNCTION USING DATA PATTERN AND
733                                                  ;TRACK SEQUENCE SPECIFIED BY INITIAL SWR SETTINGS
734                                                  ;  T = 1
735
736  003052  004737  010312        WRTONLY:    JSR PC,GETPATTERN       ;SET UP SOFTWARE DATA BUFFER
737  003056  004737  011144        XWRTONLY:   JSR PC,INITTRACKS       ;SET UP ID,OD,AND TRACK COUNTER
738  003062  004737  010614                    JSR PC,GETUNIT          ;SET UP DRIVE UNIT SELECTION
739  003066  004737  011254        1$:         JSR PC,GETTRACK         ;PICK UP NEXT TRACK
740  003072  004737  003112                    JSR PC,WRITE            ;DO THE WRITE FUNCTION
741  003076  005337  011356                    DEC TRKCNTR             ;TEST TRACK COUNTER
742  003102  001371                            BNE 1$
743  003104  004737  010732                    JSR PC,STOP             ;CHECK FOR LAST DRIVE AND EOP
744  003110  000762                            BR XWRTONLY             ;NEXT PASS
745
746  003112  004737  012216        WRITE:      JSR PC,INITSECTOR       ;SET UP FIRST,LAST,AND SECTOR COUNTER
747  003116  004737  012316        XWRITE:     JSR PC,GETSECTOR        ;PICK UP NEXT SECTOR
748  003122  012737  000012  017022            MOV #10.,SRETRY
749  003130  012737  000012  017016  ONEWRT:   MOV #10.,PRETRY         ;SET RETRY COUNTER
750  003136  004737  005656        FILLBUF:    JSR PC,ADJSUM           ;ADJUST DATA BUFFER AND CHECK SUM FOR ADDRESSES
751  003142  012746  003372                    MOV #FILLDONE,-(SP)     ;PUT GOOD RETURNON STACK
752  003146  012746  003214                    MOV #FILLER,-(SP)       ;PUT ERROR RETURN ON STACK
753  003152  005037  006434                    CLR BYTECNTR
754  003156  012746  000200                    MOV #PR4,-(SP)
755  003162  012746  003170                    MOV #1$,-(SP)
756  003166  000002                            RTI
757  003170  012777  000101  176010  1$:       MOV #FBIE,@RXCS         ;EXICUTE FILLBUFER COMMAND
758  003176  004737  007014        FILLFLAG:   JSR PC,TRCK             ;TEST FOR TRANSFER REQUEST FLAG
759  003202  112077  176002        XFRBYTE:    MOVB (R0)+,@RXDB        ;TRANSFER DATA BYTE
760  003206  005237  006434                    INC BYTECNTR
761  003212  000771                            BR FILLFLAG             ;WAIT FOR NEXT TR FLAG
762
763  003214  005726                FILLER:     TST (SP)+               ;REMOVE THE DONE RETURN FROM THE STACK
764  003216  012737  015366  003264            MOV #MFIL,PTYP1+2       ;PUT ADDR OF FILLBUF MESSAGE IN PAR ERR TYPOUT 1
765  003224  012737  015366  003354            MOV #MFIL,PTYP2+2       ;PUT ADDR OF FILLBUF MESSAGE IN PAR ERR TYPOUT 2
766  003232  012737  003136  003334            MOV #FILLBUF,PCONT+2    ;IF NOT HARD ERR RETURN THROUGH PCONT TO FILLBUF
767  003240  000137  003244                    JMP PARTEST             ;PRINT OUT PAR ERR AND TEST CONDITIONS FOR RETRY
768
769
770  003244  005237  017024        PARTEST:    INC PARLOG              ;INCREMENT PARITY ERROR
771  003250  104405                            CKSWR
772  003252  032777  020000  175734            BIT #SW13,@SWR          ;TEST DON'T PRINT ERROR SWITCH
773  003260  001006                            BNE CONT4
774  003262  104400  000000        PTYP1:      TYPE ,UPEN              ;PRINT THE PARITY ERROR MESSAGE
775  003266  104400  016017                    TYPE ,MPAR
776  003272  104400  015052                    TYPE ,MCRLF
777  003276  104405                CONT4:      CKSWR
778  003300  005777  175710                    TST @SWR                ;TEST HALT ON ERROR SWITCH
779  003304  100001                            BPL CONT13
780  003306  000000                HLT6:       HALT                    ;HALT ON ERROR
781  003310  032777  004000  175676  CONT13:   BIT #SW11,@SWR          ;TEST NO RETRY SWITCH
782  003316  001027                            BNE CONT5               ;IF SET LOG HARD ERROR
783  003320  005337  017016                    DEC PRETRY              ;DECREMENT RETRY COUNTER
784  003324  005737  017016                    TST PRETRY              ;HAVE 10 ERRORS OCCURED
```

```
                                                BEQ CONT5               ;IF CLEARED LOG HARD ERROR
003332  001402
003332  000137  003136          PCONT:        JMP FILLBUF             ;RETURN TO RETRY TEST THROUGH HERE
003336  005237  017026          CONT5:        INC HPARLOG             ;INC.HARD PARITY ERROR  COUNTER
                                              TYPE ,DBLLF
003342  104400  015240
                                              TYPE ,MUNREC            ;TYPE HARD PARITY ERROR
003346  104400  015755
003352  104400  000000          PTYP2:        TYPE ,OPEN
                                              TYPE ,MPAR
003356  104400  016017
                                              TYPE ,MCRLF
003362  104400  015052
                                              JMP DELUNIT             ;GO DELETE THE UNIT,CAN NOT CONTINUE
003366  000137  010074

                                      ;SWITCH 9 OF INITIAL TEST CONDITIONS SWITCH SETTINGS IS THE DELETED DATA
                                      ;FUNCTION INDICATOR, WHEN THIS BIT IS SET ALL WRITE / READ FUNCTIONS WILL
                                      ;SET AND CHECK THE DELETED DATA BIT ON THE DISKETTE.
                                      ;;**************************************************************

003372  012737  000012  017016  FILLDONE:     MOV #10.,PRETRY         ;SET UP RETRY COUNTER
003400  012746  003462          REWRITE:      MOV #WRTDONE,-(SP)      ;SET GOOD RETURN ON STACK
003404  012746  003510                        MOV #WRTER,-(SP)        ;SET ERROR RETURN ON STACK
003410  112737  000105  004120                MOVB #WRTIE,FUNCTION    ;SET FUNCTION WORD TO WRITE
003416  032737  001000  001212                BIT #BIT9,DTESTP        ;TEST FOR WRITE DELETED DATA
003424  001403                                BEQ 1$
003426  112737  000115  004120                MOVB #WTDDIE,FUNCTION
003434  062737  000001  021030  1$:           ADD #1,WTCNTR           ;INC TOTAL WRITE FUNCTIONS COUNTER
003442  005537  021032                        ADC WTCNTR+2            ;DOUBLE PRECISION COUNTER
003446  004737  004022                        JSR PC,COMMWORD         ;TRANSFER COMMAND TO DRIVE
003452  004737  007122                        JSR PC,DONECK           ;TEST FOR DONE FLAG
003456  000137  007204                        JMP NOINTER             ;NO INTERUPT ERROR

003462  005737  004124          WRTDONE:      TST ERWRT               ;IS THIS A REWRITE FROM A DATA ERROR
003466  001004                                BNE 1$                  ;YES GO REREAD THIS SECTOR
003470  005337  012306                        DEC SECCNTR             ;NO,TEST SECTOR COUNTER
003474  001003                                BNE 2$                  ;NOT LAST SECTOR GO TO NEXT ONE
003476  000207                                RTS PC
003500  000137  004246          1$:           JMP REREAD              ;REREAD THE SECTOR
003504  000137  005116          2$:           JMP XWRITE

003510  005726                  WRTER:        TST (SP)+               ;REMOVE THE DONE RETURN FROM THE STACK
003512  032737  000002  010070                BIT #BIT1,ASTAT         ;IS THIS A PARITY ERROR
003520  001413                                BEQ WRTSEK              ;NO, IT MUST BE A SEEK ERROR
                                      ;PARITY ERROR DURING A WRITE FUNCTION
003522  012737  016010  003264                MOV #MWRITE,PTYP1+2     ;PUT ADDR OF WRITE MESSAGE IN PAR ER TYPOUT 1
003530  012737  016010  003354                MOV #MWRITE,PTYP2+2     ;PUT ADDR OF WRITE MESSAGE IN PAR ER TYPOUT 2
003536  012737  003400  003334                MOV #REWRITE,PCONT+2    ;IF NOT HARD ERR RETURN THROUGH PCONT TO REWRITE
003544  000137  003244                        JMP PARTEST             ;GO INC LOG AND TEST FOR RETRY

                                      ;SEEK ERROR DURING A WRITE FUNCTION
003550  012737  003130  003674  WRTSEK:       MOV #ONEWRT,SEKRTY+2    ;SETUP FOR WRT RETRY ON SEEK ERROR
                                              ;(AFTER A RECAL. THE CONTENTS OF SECTOR 1,
                                              ;TRACK 1 ARE LOADED INTO THE SECTOR BUFFER.
                                              ;TO REWRITE THE CORRECT DATA THE PROGRAM
                                              ;MUST REFILL THE SECTOR BUFFER.
003556  012737  016010  003630                MOV #MWRITE,STYP1+2     ;PUT ADDR OF WRITE MESSAGE IN SEEK ER TYPEOUT 1
003564  012737  016010  003722                MOV #MWRITE,STYP2+2     ;PUT ADDR OF WRITE MESSAGE IN SEEK ER TYPEOUT 2
003572  004737  003576                        JSR PC,SEEKER           ;RECORD SEEK ERROR

003576  012703  017036          SEEKER:       MOV #ZSEKLOG,R3         ;SETUP INC INSTRUCTION FOR UNIT 0 LOG
```



```
003602  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 LOG
003606  005213                                INC (R3)                ;INCREMENT SEEK ERROR LOG
003610  004737  003770                        JSR PC,TRKERR           ;INC ERROR PER TRACK COUNTERS
003614  104405                                CKSWR
003616  032777  020000  175370                BIT #SW13,@SWR          ;CHECK DON'T PRINT ERROR SWITCH
003624  001004                                BNE SWHLT1
003626  104400  016010          STYP1:        TYPE ,MWRITE            ;PRINT WRITE (READ) SEEK ERROR
003632  004737  003736                        JSR PC,SEKTYP
003636  104405                  SWHLT1:       CKSWR
003640  005777  175350                        TST @SWR                ;TEST THE HALT ON ERROR SWITCH
003644  100001                                BPL CONT14
003646  000000                  HLT7:         HALT                    ;HALT ON THE ERROR
003650  032777  004000  175336  CONT14:       BIT #SW11,@SWR          ;CHECK THE NO RETRY SWITCH
003656  001007                                BNE HARDSK              ;IF SET LOG HARD SEEK ERROR
003660  005337  017022                        DEC SRETRY              ;HAVE 10 ERRORS BEEN LOGED
003664  001404                                BEQ HARDSK              ;YES LOG HARD ERROR
003666  004737  002550                        JSR PC,HOME             ;RECALIBRATE DRIVES ON SEEK ERROR
003672  000137  005130          SEKRTY:       JMP ONEWRT              ;NO,RETRY WRITE COMMAND (READ COMAND)
003676  012703  017076          HARDSK:       MOV #ZMSEKLOG,R3        ;SET INC. INST.FOR UNIT 0 ERR LOG
003702  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOG
003706  005213                                INC (R3)                ;HARD SEEK ERROR
003710  104400  015240                        TYPE ,DBLLF
003714  104400  015755                        TYPE ,MUNREC            ;TYPE UNRECOVERABLE SEEK ERROR
003720  104400  016010          STYP2:        TYPE ,MWRITE            ;ON WRITE (READ) COMMAND
003724  004737  003736                        JSR PC,SEKTYP
003730  062706  000002                        ADD #2,SP               ;REMOVE SEEK ERROR FROM STACK POINTER
003734  000207                                RTS PC                  ;RETURN TO NEXT SECTOR BY DONE RETURN ON STACK

003736  104400  015775          SEKTYP:       TYPE ,MSEEK             ;TYPE SEEK ERROR
003742  104400  014567                        TYPE ,MPRES             ;TYPE ADDRESS OF TRACK MOVED FROM
003746  013737  011362  003760                MOV PRESTRK,1$
003754  004537  014446                        JSR R5,SGLDEC
003760  000000                  1$:           OPEN
003762  104400  015052                        TYPE ,MCRLF
003766  000207                                RTS PC

003770  013705  011360          TRKERR:       MOV TARGET,R5           ;SET UP TO INC ERROR PER TRACK COUNTER
003774  006305                                ASL R5                  ;ADJUST FOR EVEN ADDRESS
003776  062705  020344                        ADD #U0TRK,R5           ;ADDIN ADDERSS OF UNIT 0 LOG
004002  032737  000020  010704                BIT #BIT4,UNITSEL       ;CHECK THE UNIT SELECTION BIT
004010  001402                                BEQ 1$                  ;IF CLEARED UNIT 0 IS ACTIVE
004012  062705  020576                        ADD #U1TRK,R5           ;ADJUST FOR UNIT 1 LOG
004016  005215                  1$:           INC (R5)                ;INC THE CORRECT COUNTER
004020  000207                                RTS PC

004022  013705  011360          COMMWORD:     MOV TARGET,R5           ;GET TRACK NUMBER
004026  006305                                ASL R5                  ;MULTIPLY BY 4 TO DOUBLE PRECISION
004030  006305                                ASL R5                  ;INTERLEAVE COUNTER LOCATIONS
004032  062705  017174                        ADD #TKACC,R5           ;ADD ON ADDRESS OF TRACK ACCESS COUNTER
004036  062715  000001                        ADD #1,(R5)             ;INCREMENT THE COUNTER
004042  005565  000002                        ADC 2(R5)               ;ADD CARRY TO HIGH ORDER WORD
004046  153737  010704  004120                BISB UNITSEL,FUNCTION   ;SET UNIT SELECTION BIT IN COMMAND WORD
004054  013777  004120  175124                MOV FUNCTION,@RXCS      ;SEND OUT COMMAND TO DRIVE
004062  004737  007014                        JSR PC,TRCK             ;WAIT FOR TR FLAG
004066  113777  012310  175114                MOVB TSECTOR,@RXDB      ;SEND OUT TARGET SECTOR
004074  004737  007014                        JSR PC,TRCK             ;WAIT FOR TR FLAG
```

```
                                        MOVB TARGET,@RXDB       ;SEND OUT TARGET TEACK
                                        CLR -(SP)               ;LOWER INTERUPT LEVEL TO ALLOW AN INTERUPT
                                        MOV #1S,-(SP)
                                        RTI
                                1S:     RTS PC

                                FUNCTION:       0
                                RDAFTWT:        0               ;READ AFTER WRITE FUNCTION FLAG
                                ERWRT:          0               ;WRITE CAUSED BY DATA ERROR FLAG
                                DATACK:         0               ;DATA CHECK ON CRC ERROR FLAG

                                ;;*************************************************************

                                        ;THIS IS A WRITE ALL SECTORS FOLLOWED BY A READ ALL SECTORS
                                        ;WITH DATA PATTERNS AND HEAD SEQUENCING SET BY INITIAL SWR
                                        ;   T = 2

                                WRTRD:          COM RDAFTWT             ;SET READ AFTER WRITE FLAG
                                                JSR PC,GETPATTERN
                                XWRTRD:         JSR PC,INITTRACKS
                                                JSR PC,GETUNIT
                                1$:             JSR PC,GETTRACK
                                                JSR PC,WRITE
                                                JSR PC,READ
                                                DEC TRKCNTR
                                                BNE 1$
                                                JSR PC,STOP
                                                BR XWRTRD

                                ;;*************************************************************

                                        ;READ DATA FROM THE DISKETTE

                                READ:           JSR PC,INITSECTOR
                                XREAD:          JSR PC,GETSECTOR
                                                MOV #10.,RDRETRY        ;SET UP RETRY COUNTERS FOR DELETED DATA
                                                MOV #10.,DATARETRY      ;DATA ERROR
                                                MOV #10.,PRETRY         ;PARITY
                                                MOV #10.,SRETRY         ;SEEK
                                                MOV #10.,CRETRY         ;AND CRC ERRORS
                                REREAD:         CLR DATACK              ;CLEAR CRC DATA CHECK FLAG
                                                MOV #RDDONE,-(SP)       ;SET GOOD RETURN ON STACK
                                                MOV #RDERR,-(SP)        ;SET READ ERROR RETURN ON STACK
                                                MOVB #RDIE,FUNCTION
                                                ADD #1,RDCNTR           ;INC TOTAL READ FUNCTIONS COUNTER
                                                ADC RDCNTR+2            ;DOUBLE PRECISION COUNTER
                                                JSR PC,COMMWORD
                                                JSR PC,DONECK           ;TEST FOR DONE
                                                JMP NOINTER             ;NO INTERUPT ON DONE


                                RDDONE:         CMP #10.,RDRETRY        ;IS READ RETRY EQUAL TO 10
                                                BEQ CONT1               ;YES,NO ERRORS OCCURED
                                                MOV #ZRDLOG,R3          ;SET INC. INST. FOR UNIT 0 ERROR LOG
```



```
                                                JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOG
                                                INC (R3)                ;INC RECOVERABLE READ LOG
                                                MOV #10.,RDRETRY        ;RESET READ RETRY COUNTER
                                                TYPE ,DBLLF
                                                TYPE ,MREC              ;TYPE RECOVERABLE READ ERROR
                                                TYPE ,MREAD
                                                TYPE ,MCRLF
                                CONT1:          CMP #10.,WTRETRY        ;IS WRITE RETRY EQUAL TO 10
                                                BEQ CONT2               ;YES,NO ERRORS OCCURED
                                                MOV #ZWRTLOG,R3         ;SET INC.INST.FOR UNIT 0 ERROR LOG
                                                JSR PC,U1LOG            ;TEST FOR UNIT 1 LOG
                                                INC (R3)                ;INC RECOVERABLE WRITE LOG
                                                MOV #10.,WTRETRY        ;RESET THE WRITE RETRY COUNTER
                                                TYPE ,DBLLF
                                                TYPE ,MREC              ;TYPE RECOVERABLE WRITE ERROR
                                                TYPE ,MWRITE
                                                TYPE ,MCRLF
                                CONT2:          JSR PC,ODDCHK           ;CHECK FOR DELETED DATA INDICATOR
                                                TST R1                  ;BIT 15 OF R1 IS READ 1 SECTOR FLAG
                                                BPL NEXTRD
                                                RTS PC                  ;IF SET,GO VERIFY DATA JUST READ
                                NEXTRD:         DEC SECCNTR
                                                BNE XREAD
                                                RTS PC                  ;READ FUNCTION IS DONE

                                RDERR:          TST (SP)+               ;REMOVE THE DONE RETURN FROM THE STACK
                                                BIT #BIT1,ASTAT         ;IS THIS A PARITY ERROR
                                                BEQ 1$                  ;NO, SEE IF ITS A CRC ERROR
                                        ;PARITY ERROR DURING A READ FUNCTION
                                                MOV #MREAD,PTYP1+2      ;PUT ADDR OF READ MESSAGE IN PAR ERR TYPEOUT 1
                                                MOV #MREAD,PTYP2+2      ;PUT ADDR OF READ MESSAGE IN PAR ER TYPEOUT 2
                                                MOV #REREAD,PCONT+2     ;IF HARD ERR RETURN THROUGH PCONT TO REREAD
                                                JMP PARTEST             ;RECORD PARITY ERROR AND RETRY FUNCTION
                                1$:             BIT #BIT0,ASTAT         ;IS THIS A CRC ERROR
                                                BNE CRCER               ;YES GO TEST AND LOG IT
                                        ;SEEK ERROR DURING A READ FUNCTION
                                                MOV #REREAD,SEKRTY+2    ;SET SEEK CONTINUE FOR READ RETRY
                                                MOV #MREAD,STYP1+2      ;SET ADDR OF READ MESSAGE IN SEEK ER TYPEOUT 1
                                                MOV #MREAD,STYP2+2      ;SET ADDR OF READ MESSAGE IN SEEK ER TYPEOUT 2
                                                JSR PC,SEEKER           ;RECORD SEEK ERROR
                                                BR NEXTRD               ;GO TO NEXT SECTOR,CAN'T READ THIS ONE
                                        ;CRC ERROR DETECTED WHILE READING
                                CRCER:          JSR PC,TRKERR           ;INC ERROR PER TRACK COUNTER
                                                TST R1                  ;IF READ ONLY, REPORT DATA CRC ERROR
                                                BPL DATACRC
                                                INC DATACK              ;SET DATA CHECK FLAG
                                                JSR PC,EMPBUFF          ;CHECK FOR A DATA ERROR
                                                TST ERCNTR              ;WAS THERE A DATA ERROR
                                                BNE DATACRC             ;YES,REREAD AND/OR REWRITE THE DATA
                                                MOV #ZCRCBAD,R3         ;SET INC.INST.FOR UNIT 0 ERROR LOG
                                                JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
                                                INC (R3)                ;NO,INC BAD CRC GENERATOR ERROR
                                                CKSWR
                                                BIT #SW13,@SWR          ;TEST DON'T SWITCH
                                                BNE 2$
                                                TYPE ,MBADCRC           ;TYPE CRC GENERATOR ERROR
```

```
1009  004636  104400  015052                        TYPE ,MCRLF
1010  004642  104405                    2$:         CKSWR
1011  004644  005777  174344                        TST @SWR                ;TEST HALT ON ERROR SWITCH
1012  004650  100001                                BPL CONT15
1013  004652  000000                    HLT10:      HALT                    ;HALT ON ERROR
1014  004654  032777  004000  174332    CONT15:     BIT #SW11,@SWR          ;CHECK NO RETRY SIWTCH
1015  004662  001005                                BNE 3$                  ;IF SET LOG HARD ERROR
1016  004664  005337  017020                        DEC CRETRY              ;HAVE 10 ERRORS BEEN LOGED
1017  004670  001402                                BEQ 3$                  ;YES,LOG HARD ERROR
1018  004672  000137  004246                        JMP REREAD              ;NO,GO REREAD DATA
1019  004676  012703  017106            3$:         MOV #ZMCRCBAD,R3        ;SET INC,INST,FOR UNIT 0 ERROR LOG
1020  004702  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1021  004706  005213                                INC (R3)
1022  004710  104400  015240                        TYPE ,DBLLF
1023  004714  104400  015644                        TYPE ,MBADCRC           ;TYPE HARD CRC GENERATOR ERROR
1024  004720  104400  015052                        TYPE ,MCRLF
1025  004724  104400  015724                        TYPE ,MHALT11           ;AND HALT AS THE CRC GENERATOR DOESN'T WORK
1026  004730  000000                    HLT11:      HALT
1027                                          ;DATA CRC ERROR REREAD AND/OR REWRITE TO GET GOOD DATA
1028  004732  012703  017042            DATACRC:    MOV #ZCRCLOG,R3         ;SET INC,INST,FOR UNIT 0 ERROR LOG
1029  004736  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1030  004742  005213                                INC (R3)                ;TRUE DATA CRC ERROR
1031  004744  104405                                CKSWR
1032  004746  032777  020000  174240                BIT #SW13,@SWR          ;TEST DON'T PRINT ERROR SWITCH
1033  004754  001004                                BNE 4$
1034  004756  104400  015736                        TYPE ,MCRC              ;TYPE DATA CRC ERROR
1035  004762  104400  015052                        TYPE ,MCRLF
1036  004766  104405                    4$:         CKSWR
1037  004770  005777  174220                        TST @SWR                ;TEST HALT ON ERROR SWITCH
1038  004774  100001                                BPL CONT16
1039  004776  000000                    HLT12:      HALT                    ;HALT ON ERROR
1040  005000  032777  004000  174206    CONT16:     BIT #SW11,@SWR          ;TEST NO RETRY SWITCH
1041  005006  001005                                BNE 5$                  ;IF SET LOG HARD ERROR
1042  005010  005337  017004                        DEC RDRETRY             ;HAVE 10 ERRORS OCCURED
1043  005014  001402                                BEQ 5$                  ;YES LOG HARD ERROR OR REWRITE DATA
1044  005016  000137  004246                        JMP REREAD              ;NO,GO REREAD THIS SECTOR
1045  005022  012703  017102            5$:         MOV #ZMCRCLOG,R3        ;SET INC,INST,FOR UNIT 0 ERROR LOG
1046  005026  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIY 1 ERROR LOGS
1047  005032  005213                                INC (R3)                ;INC HARD CRC ERROR LOG
1048  005034  012737  000012  017004                MOV #10.,RDRETRY        ;RESET READ RETRY COUNTER
1049  005042  005737  004122                        TST RDAFTWT             ;IS THIS A READ AFTER WRITE FUNCTION
1050  005046  001021                                BNE CONT6               ;YES,GO REWRITE IT
1051  005050  012703  017112                        MOV #ZHRDLOG,R3         ;NO,SET INC,INST,FOR UNIT 0 ERROR LOG
1052  005054  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1053  005060  005213                                INC (R3)                ;READ ONLY, HARD READ ERROR
1054  005062  104400  015240                        TYPE ,DBLLF
1055  005066  104400  015755                        TYPE ,MUNREC            ;TYPE UNRECOVERABLE READ ERROR
1056  005072  104400  015674                        TYPE ,MREAD
1057  005076  104400  015052            CONT7:      TYPE ,MCRLF
1058  005102  062706  000002                        ADD #2,SP               ;REMMOVE READ DONE ADDRESS FROM STACK
1059  005106  000137  004450                        JMP NEXTRD              ;READ NEXT SECTOR CAN'T READ THIS ONE
1060  005112  104405            CONT6:      CKSWR
1061  005114  032777  004000  174072                BIT #SW11,@SWR          ;TEST NO RETRY SWITCH
1062  005122  001011                                BNE 7$                  ;IF SET LOG HARD ERROR
1063  005124  005337  017006                        DEC WTRETRY             ;HAVE 10 REWRITES BEEN DONE
1064  005130  001406                                BEQ 7$                  ;YES,LOG HARD WRITE ERROR
```



```
1065  005132  005137  004124                        COM ERWRT               ;NO,SET THE WRITE CAUSED BY ERROR FLAG
1066  005136  062706  000002                        ADD#2,SP                ;REMOVE RUDONE ADDRESS FROM STACK
1067  005142  000137  003130                        JMP UNEWRT              ;REWRITE SECTOR THEN REREAD TO CHECK FOR ERROR
1068  005146  012703  017116            7$:         MOV #ZHWRTLOG,R3        ;SET INC,INST,FOR UNIT 0 ERROR LOG
1069  005152  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1070  005156  005213                                INC (R3)                ;HARD WRITE ERROR
1071  005160  012737  000012  017006                MOV #10.,WTRETRY        ;RESET WRITE RETRY COUNTER
1072  005166  104400  015240                        TYPE ,DBLLF
1073  005172  104400  015755                        TYPE ,MUNREC            ;TYPE UNRECOVERABLE WRITE ERROR
1074  005176  104400  016010                        TYPE ,MWRITE
1075  005202  000735                                BR CONT7
1076
1077
1078
1079  005204  032737  001000  001212    DDCHK:      BIT #BIT9,DTESTP        ;TEST BIT 9 AS TO DELETED DATA TRANSFER
1080  005212  001472                                BEQ CONT10
1081  005214  132737  000100  010070                BITB #BIT6,ASTAT        ;THIS IS A DELETED DATA FUNCTION
1082  005222  001114                                BNE RETURN              ;DD BIT SHOULD BE SET
1083  005224  012703  017066                        MOV #ZDDMIS,R3          ;SET INC INST FOR UNIT 0 ERROR LOG
1084  005230  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1085  005234  005213                                INC (R3)                ;INC MISSING DELETED DATA LOG
1086  005236  004737  003770                        JSR PC,TRKERR           ;INC ERROR PER TRACK COUNTER
1087  005242  104405                                CKSWR
1088  005244  032777  020000  173742                BIT #SW13,@SWR          ;TEST DON'T PRINT ERROR SWITCH
1089  005252  001011                                BNE CONT11
1090  005254  104400  014506                        TYPE ,MDDMIS            ;TYPE MISSING DELETED DATA BIT
1091  005260  052737  000400  010704    DDERR:      BIS #BIT8,UNITSEL       ;SET HAD ERROR FLAG
1092  005266  004737  007716                        JSR PC,TYPADR           ;TYPE ADDRESS OF ERROR
1093  005272  104400  015052                        TYPE ,MCRLF
1094  005276  104405            CONT11:     CKSWR
1095  005300  005777  173710                        TST @SWR                ;TEST HALT ON ERROR SWITCH
1096  005304  100001                                BPL CONT17
1097  005306  000000                    HLT13:      HALT                    ;HALT ON DELETED DATA ERROR
1098  005310  032777  004000  173676    CONT17:     BIT #SW11,@SWR          ;TEST NO RETRY SWITCH
1099  005316  001005                                BNE 4$                  ;IF SET LOG HARD ERROR
1100  005320  005337  017010                        DEC DDRETRY             ;HAVE 10 ERRORS BEEN LOGED
1101  005324  001402                                BEQ 4$                  ;YES LOG HARD ERROR
1102  005326  000137  004246                        JMP REREAD              ;NO, REREAD SECTOR
1103  005332  012703  017126            4$:         MOV #ZMDDLOG,R3         ;SET INC INST,FOR UNIT 0 ERROR LOG
1104  005336  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1105  005342  005213                                INC (R3)
1106  005344  104400  015240                        TYPE ,DBLLF
1107  005350  104400  015755                        TYPE ,MUNREC            ;TYPE UNRECOVERABLE DELETED DATA ERROR
1108  005354  104400  015615                        TYPE ,MODER
1109  005360  104400  015052                        TYPE ,MCRLF
1110  005364  004737  007716                        JSR PC,TYPADR
1111  005370  104400  015052                        TYPE ,MCRLF
1112  005374  000137  004450                        JMP NEXTRD              ;READ NEXT SECTOR
1113  005400  032737  000100  010070    CONT10:     BIT #BIT6,ASTAT         ;THIS IS NOT A DELETED DATA TRANSFER
1114  005406  001422                                BEQ RETURN
1115  005410  012703  017072                        MOV #ZUNXDD,R3          ;SET INC INST,FOR UNIT 0 ERROR LOG
1116  005414  004737  021044                        JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1117  005420  005213                                INC (R3)                ;UNEXPECTED DD BIT SET
1118  005422  052737  000400  010704                BIS #BIT8,UNITSEL       ;SET HAD ERROR FLAG
1119  005430  004737  003770                        JSR PC,TRKERR           ;INC ERROR PER TRACK COUNTER
1120  005434  104405                                CKSWR
```

```
 1121  005436  032777  020000  173550          BIT #SW13,@SWR          ;TEST DON'T PRINT ERROR SWITCH
 1122  005444  001314                          BNE CONT11
 1123  005446  104400  014462                  TYPE ,MUNXDD            ;TYPE UNEXPECTED DELETED DATA BIT
 1124  005452  000702                          BR DDERR
 1125  005454  000207              RETURN:     RTS PC
 1126
 1127
 1128
 1129                              ;;*************************************************************
 1130
 1131                                      ;WRITE ALL SECTORS,READ AND VERIFY ALL SECTORS
 1132                                      ;   T = 3
 1133
 1134  005456  005137  004122      WTRDCK:     COM RDAFTWT             ;SET READ AFTER WRITE FLAG
 1135  005462  004737  010312                  JSR PC,GETPATTERN
 1136  005466  004737  011144      XWTRDCK:    JSR PC,INITTRACKS
 1137  005472  004737  010614                  JSR PC,GETUNIT
 1138  005476  004737  011254      1$:         JSR PC,GETTRACK
 1139  005502  004737  003112                  JSR PC,WRITE
 1140  005506  004737  005526                  JSR PC,READCHK
 1141  005512  005337  011356                  DEC TRKCNTR
 1142  005516  001367                          BNE 1$
 1143  005520  004737  010732                  JSR PC,STOP
 1144  005524  000760                          BR XWTRDCK
 1145
 1146
 1147                              ;;*************************************************************
 1148
 1149                                      ;READ A SECTOR,EMPTY THE SECTOR BUFFER AND VERIFY
 1150                                      ;THE DATA READ AGAINST CORE DATA BUFFER
 1151
 1152  005526  052701  100000      READCHK:    BIS #BIT15,R1           ;SET READ ONE SECTOR FLAG
 1153  005532  004737  004200                  JSR PC,READ             ;GO READ ONE SECTOR
 1154  005536  005737  012306      EMPBUFF:    TST SECCNTR             ;IF CLEARED NO SECTOR WAS FOUND
 1155  005542  001002                          BNE 2$
 1156  005544  000137  006432                  JMP EXIT                ;GO TO NEXT TRACK CAN'T READ THIS ONE
 1157  005550  005037  006434      2$:         CLR BYTECNTR            ;CLEAR THE BYTE AND ERROR COUNTERS
 1158  005554  005037  006442                  CLR ERCNTR
 1159  005560  052701  000200                  BIS #BIT7,R1            ;R1 BIT 7 IS USED AS FIRST ERROR FLAG
 1160  005564  004737  005656                  JSR PC,ADJSUM           ;ADJUST DATA AND CK SUM FOR ADDRESSES
 1161  005570  005037  005746                  CLR CKSUM               ;SET UP FOR CHECK SUM ACCUMULATION
 1162  005574  012746  006212                  MOV #EMPDONE,-(SP)      ;SET UP RETURN ADDRESSES
 1163  005600  012746  005750                  MOV #EMPER,-(SP)
 1164  005604  005046                          CLR -(SP)               ;LOWER INTERUPT LEVEL
 1165  005606  012746  005614                  MOV #1$,-(SP)
 1166  005612  000002                          RTI
 1167  005614  012777  000103  173364  1$:     MOV #EBIE,@RXCS         ;LOAD EMPTY BUFFER FUNCTION
 1168  005622  004737  007014      EMPFLAG:    JSR PC,TRCK             ;TEST FOR TR FLAG
 1169
 1170  005626  117737  173356  006436  CKBYTE:  MOVB @RXDB,BADBYTE     ;SAVE BYTE FROM DISKETTE
 1171  005634  063737  006436  005746          ADD BADBYTE,CKSUM       ;ACCUMULATE CHECK SUM
 1172  005642  123720  006436                  CMPB BADBYTE,(R0)+      ;COMPARE AGAINST GOOD BYTE
 1173  005646  001054                          BNE DATAER              ;IF NOT EQUAL GO TO DATAER
 1174  005650  005237  006434                  INC BYTECNTR
 1175  005654  000762                          BR EMPFLAG              ;GET NEXT BYTE
 1176
```



```
 1177  005656  113737  011360  016604  ADJSUM:     MOVB TARGET,BUFADR      ;SET FIRST AND SECOND BYTES WITH ADDRESSES
 1178  005664  113737  012310  016605              MOVB TSECTOR,BUFADR+1
 1179  005672  013737  010612  005746              MOV SUM,CKSUM           ;GET THE PATTERN SUM
 1180  005700  063737  011360  005746              ADD TARGET,CKSUM        ;ADD TRACK ADDRESS TO CHECK SUM
 1181  005706  063737  012310  005746              ADD TSECTOR,CKSUM       ;ADD SECTOR ADDRESS TO CHECK SUM
 1182  005714  113737  005746  017002              MOVB CKSUM,BUFADR+176   ;INSERT CHECK SUM TO DATA BUFFER
 1183  005722  106337  005746                      ASLB CKSUM              ;GENERATE NEGITIVE CHECK SUM
 1184  005726  105437  005746                      NEGB CKSUM
 1185  005732  113737  005746  017003              MOVB CKSUM,BUFADR+177   ;INSERT NEG,SUM INTO DATA BUFFER
 1186  005740  012700  016604                      MOV #BUFADR,R0          ;SET ADDRESS OF BYTE IN R0
 1187  005744  000207                              RTS PC                  ;RETURN
 1188
 1189  005746  000000              CKSUM:      0
 1190
 1191  005750  005726              EMPER:      TST (SP)+               ;REMOVE THE DONE RETURN FROM THE STACK
 1192  005752  012737  015402  003264          MOV #MEMPTY,PTYP1+2     ;PUT ADDR OF EMPTYBUF MESSAGE IN PAR ER TYPOUT 1
 1193  005760  012737  015402  003354          MOV #MEMPTY,PTYP2+2     ;PUT ADDR OF EMPTYBUF MESSAGE IN PAR ER TYPOUT 2
 1194  005766  012737  005536  003334          MOV #EMPBUFF,PCONT+2    ;NOT HARD ERR RETURN THROUGH PCONT TO EMPTYBUF
 1195  005774  000137  003244                  JMP PARTEST             ;REPORT PARITY ERROR
 1196
 1197  006000  052737  000400  010704  DATAER:    BIS #BIT8,UNITSEL       ;SET THE HAD ERROR FLAG
 1198  006006  005237  006442                  INC ERCNTR              ;INC THE BYTE ERROR COUNTER
 1199  006012  104405                          CKSWR
 1200  006014  032777  020000  173172          BIT #SW13,@SWR          ;TEST PRINT ERROR SW IN SWR
 1201  006022  001062                          BNE NOERTYP             ;DON'T PRINT THE ERROR
 1202  006024  032777  010000  173162          BIT #SW12,@SWR          ;TEST PRINT 10 ERRORS SWITCH
 1203  006032  001404                          BEQ 1$                  ;PRINT ALL ERRORS
 1204  006034  023727  006442  000012          CMP ERCNTR,#10.         ;HAVE 10 ERRORS BEEN TYPED
 1205  006042  003052                          BGT NOERTYP             ;YES,DON'T PRINT ANY MORE
 1206  006044  005737  004126          1$:     TST DATACK              ;WAS THIS A READ CHECK DUE TO CRC ERROR
 1207  006050  001403                          BEQ 2$                  ;NO,REPORT DATA ERRORS
 1208  006052  105701                          TSTB R1                 ;YES,TEST FIRST ERROR FLAG
 1209  006054  100017                          BPL TYPERR              ;TYPE THE DATA CRC ERROR
 1210  006056  000412                          BR 3$
 1211  006060  105701                  2$:     TSTB R1                 ;TEST FIRST ERROR FLAG
 1212  006062  100014                          BPL TYPERR
 1213  006064  104400  015052                  TYPE ,MCRLF
 1214  006070  104400  014527                  TYPE ,MDERHDR           ;PRINT ERROR HEADER
 1215  006074  104400  015052                  TYPE ,MCRLF
 1216  006100  004737  007716                  JSR PC,TYPADR           ;PRINT TRACK AND SECTOR LOCATIONS
 1217  006104  104400  014616          3$:     TYPE ,MCOLMUN           ;SET UP COLMUN HEADINGS
 1218  006110  042701  000200                  BIC #BIT7,R1            ;CLEAR FIRST ERROR FLAG
 1219  006114  013737  006434  006126  TYPERR:    MOV BYTECNTR,1$         ;PRINT BYTE NUMBER
 1220  006122  004537  014446                  JSR R5,SGLDEC
 1221  006126  000000              1$:         OPEN
 1222  006130  104400  015047                  TYPE ,DBLSP
 1223  006134  013746  006436                  MOV BADBYTE,-(SP)       ;PRINT BYTE READ FROM DISKETTE
 1224  006140  104402                          TYPOS
 1225  006142  000003                          .WORD 3
 1226  006144  104400  015047                  TYPE ,DBLSP
 1227  006150  114037  006440                  MOVB -(R0),GOODBYTE     ;GET GOOD BYTE
 1228  006154  005200                          INC R0                  ;RETURN R0 TO NEXT BYTE IN BUFFER
 1229  006156  013746  006440                  MOV GOODBYTE,-(SP)
 1230  006162  104403                          TYPON                   ;PRINT GOOD DATA
 1231  006164  104400  015052                  TYPE ,MCRLF
 1232  006170  104405              NOERTYP:    CKSWR
```

```
1233  006172  005777  173016                    TST @SWR                ;TEST HALT ON ERROR SWITCH
1234  006176  100001                            BPL CONT20
1235  006200  000000                  HLT14:    HALT
1236  006202  005237  006434          CONT20:   INC BYTECNTR
1237  006206  000137  005622                    JMP EMPFLAG
1238
1239  006212  005737  004126          EMPDONE:  TST DATACK              ;WAS THIS READCHECK CAUSED BY CRC ERROR
1240  006216  001401                            BEQ 1$                  ;NO,CONTINUE
1241  006220  000207                            RTS PC                  ;YES,RETURN TO CRC ERROR HANDLER
1242  006222  005737  006442          1$:       TST ERCNTR              ;WAS THERE ERRORS
1243  006226  001471                            BEQ CONT3               ;NO ERRORS
1244  006230  104405                            CKSWR
1245  006232  032777  020000  172754            BIT #SW13,@SWR          ;YES,TEST DON'T PRINT SWITCH
1246  006240  001024                            BNE 2$                  ;DON'T PRINT THE ERROR
1247  006242  104400  015333                    TYPE ,MERCT             ;PRINT THE TOTAL DATA ERROR COUNT
1248  006246  013737  006442  006260            MOV ERCNTR,3$
1249  006254  004537  014446                    JSR R5,SGLDEC
1250  006260  000000                  3$:       OPEN
1251  006262  104400  016327                    TYPE ,MSUM              ;INDICATE IF CHECK SUM WAS GOOD OR HAD ERRORS
1252  006266  105737  005746                    TSTB CKSUM
1253  006272  001403                            BEQ 4$
1254  006274  104400  016067                    TYPE ,MERS
1255  006300  000402                            BR 5$
1256  006302  104400  016322          4$:       TYPE,MGOOD
1257  006306  104400  015052          5$:       TYPE ,MCRLF
1258  006312  012703  017062          2$:       MOV #ZUATALOG,R3        ;SET INC,INST,FOR UNIT 0 ERROR LOG
1259  006316  004737  021044                    JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1260  006322  005213                            INC (R3)                ;INC DATA ERROR LOG
1261  006324  004737  003770                    JSR PC,TRKERR           ;INC ERROR PER TRACK COUNTER
1262  006330  104405                            CKSWR
1263  006332  032777  004000  172654            BIT #SW11,@SWR          ;TEST NO RETRY SWITCH
1264  006340  001007                            BNE 6$                  ;IF SET LOG HARD ERROR
1265  006342  005337  017012                    DEC DATARETRY           ;HAVE 10 ERRORS BEEN LOGED
1266  006346  001404                            BEQ 6$
1267  006350  004737  004246                    JSR PC,REREAD           ;NO,GO REREAD THE DATA
1268  006354  000137  005536                    JMP EMPBUFF             ;GO RECHECK THE DATA
1269  006360  012703  017122          6$:       MOV #ZMDATALOG,R3       ;YES,SET INC,INST,FOR UNIT 0 ERROR LOG
1270  006364  004737  021044                    JSR PC,U1LOG            ;TEST FOR UNIT 1 ERROR LOGS
1271  006370  005213                            INC (R3)                ;INC HARD DATA NO STATUS ERROR
1272  006372  104400  015240                    TYPE ,UBELLF
1273  006376  104400  015755                    TYPE ,MUNREC            ;TYPE UNRECOVERABLE DATA NO
1274  006402  104400  014527                    TYPE ,MDERMDR           ;STATUS ERROR
1275  006406  104400  015052                    TYPE ,MCRLF
1276  006412  005337  012306          CONT3:    DEC SECCNTR
1277  006416  001405                            BEQ EXIT
1278  006420  004737  004204                    JSR PC,XREAD            ;READ THE NEXT SECTOR
1279  006424  000137  005536                    JMP EMPBUFF
1280  006430  005001                  3$:       CLR R1                  ;CLEAR THE ONE READ FLAG
1281  006432  000207                  EXIT:     RTS PC
1282
1283  006434  000000                  BYTECNTR: 0
1284  006436  000000                  BADBYTE:  0
1285  006440  000000                  GOODBYTE: 0
1286  006442  000000                  ERCNTR:   0
1287
1288                                  ;;*****************************************************************
```



```
1289
1290                                          ;READ AND VERIFY ALL SECTORS WRITTEN BY
1291                                          ;PREVIOUS WRITE FUNCTION
1292
1293                                          ;   T = 4
1294
1295  006444  004737  010312          RDCHK:    JSR PC,GETPATTERN
1296  006450  004737  011144          XRDCHK:   JSR PC,INITTRACKS
1297  006454  004737  010614                    JSR PC,GETUNIT
1298  006460  004737  011254          1$:       JSR PC,GETTRACK
1299  006464  004737  005526                    JSR PC,READCHK
1300  006470  005337  011356                    DEC TRKCNTR
1301  006474  001371                            BNE 1$
1302  006476  004737  010732                    JSR PC,STOP
1303  006502  000762                            BR XRDCHK
1304
1305
1306                                  ;;*****************************************************************
1307
1308                                          ;DO A READ ONLY FUNCTION ON ALL SECTORS
1309                                          ;THIS DOES NOT VERIFY THE DATA, ONLY TESTS FOR CRC ERRORS
1310                                          ;   T = 5
1311
1312  006504  004737  010312          RDONLY:   JSR PC,GETPATTERN
1313  006510  004737  011144          XRDONLY:  JSR PC,INITTRACKS
1314  006514  004737  010614                    JSR PC,GETUNIT
1315  006520  004737  011254          1$:       JSR PC,GETTRACK
1316  006524  004737  004200                    JSR PC,READ
1317  006530  005337  011356                    DEC TRKCNTR
1318  006534  001371                            BNE 1$
1319  006536  004737  010732                    JSR PC,STOP
1320  006542  000762                            BR XRDONLY
1321
1322
1323
1324                                  ;;*****************************************************************
1325
1326                                          ;WRITE AND READ CHECK ANY DATA PATTERN,AND ANY TRACK SEQUENCE
1327                                          ;BUT ALTERNATE BETWEEN DRIVE UNITS ON EACH TRACK SELECTED.
1328                                          ;   T = 6
1329
1330  006544  005137  004122          DRVSWP:   COM RDAFTWT             ;SET THE READ AFTER WRITE FLAG
1331  006550  004737  010312                    JSR PC,GETPATTERN
1332  006554  004737  011144          2$:       JSR PC,INITTRACKS
1333  006560  004737  010614          1$:       JSR PC,GETUNIT          ;SET UP FOR UNIT 0
1334  006564  004737  011254                    JSR PC,GETTRACK         ;GET THE TRACK TO BE ACCESSED
1335  006570  004737  003112                    JSR PC,WRITE
1336  006574  004737  005526                    JSR PC,READCHK
1337  006600  004737  010614                    JSR PC,GETUNIT          ;NOW GO TO UNIT 1 ON THE SAME TRACK
1338  006604  004737  003112                    JSR PC,WRITE
1339  006610  004737  005526                    JSR PC,READCHK
1340  006614  005337  011356                    DEC TRKCNTR
1341  006620  001357                            BNE 1$
1342  006622  004737  010732                    JSR PC,STOP
1343  006626  000752                            BR 2$
1344
```

```
                                    ;;*************************************************************
                                    ;THE FIRST HALF OF TEST 7 FORCES THE TRACK SEQUENCE
                                    ;TO INCRIMENT UP THROUGH ALL THE TRACKS DOING WRITE / READ CHECK FUNCTIONS.
                                    ;THIS VERIFIES THAT ALL THE TRACKS ARE ACCESSABLE.
                                    ;THE SECOND HALF OF THE PASS WILL USE THE SEQUENCE SELECTED BY THE
                                    ;OPERATOR AS INDICATED BELOW,AND ONLY READ CHECK THE DATA JUST WRITTEN.
                                    ;THIS VERIFIES THAT THE DATA CAN BE READ FROM A TRACK,AFTER THE HEAD HAS BEEN
                                    ;MOVED AWAY FROM AND BACK TO THAT TRACK. AT THE COMPLETION OF THE PASS
                                    ;THE DELETED DATA BIT IN TEST CONDITIONS IS COMPLEMENTED AND THE NEXT
                                    ;PASS WILL BE RUN UNDER THIS NEW CONDITION.
                                    ;   T = 0 OR 7
006630  004737  010312          TEST7:      JSR PC,GETPATTERN
006634  013737  011370  007012              MOV SEQUEN,SEQSAV       ;SAVE THE SELECTED SEQUENCE
006642  012737  000001  011370  TEST7X:      MOV #1,SEQUEN           ;FORCE INCREMENT SEQUENCE
006650  005137  004122                      COM RDAFTWT             ;SET READ AFTER WRITE FLAG FOR ERROR TESTING
006654  004737  011144                      JSR PC,INITTRACKS
006660  004737  010614                      JSR PC,GETUNIT
006664  004737  011254          1$:         JSR PC,GETTRACK
006670  004737  003112                      JSR PC,WRITE
006674  004737  005526                      JSR PC,READCHK
006700  005337  011356                      DEC TRKCNTR             ;IS THE FIRST HALF OF THE PASS DONE
006704  001367                              BNE 1$                  ;NO,CONTINUE WRITTING AND READING
006706  005037  004122                      CLR RDAFTWT             ;YES,NOW DO A READ ONLY
006712  013737  007012  011370              MOV SEQSAV,SEQUEN       ;USING SELECTED SEQUENCE
006720  004737  011144                      JSR PC,INITTRACKS
006724  004737  011254          2$:         JSR PC,GETTRACK
006730  004737  005526                      JSR PC,READCHK
006734  005337  011356                      DEC TRKCNTR             ;IS THIS HALF OF THE PASS DONE
006740  001371                              BNE 2$                  ;NO,CONTINUE READ CHECKING
006742  004737  010732                      JSR PC,STOP             ;YES,TEST FOR END OF PASS CONDITIONS
006746  032737  040100  010704              BIT #40100,UNITSEL      ;HAVE ALL SELECTED UNITS BEEN USED
006754  001332                              BNE TEST7X              ;IF A USED BIT IS STILL SET GO TO NEXT DRIVE
006756  012706  001200                      MOV #1200,SP            ;RESET STACK ADDRESS FOR ACT11 EOP
006762  032737  001000  001212              BIT #BIT9,DTESTP        ;NEXT PASS,COMPLEMENT D U BIT
006770  001404                              BEQ 3$
006772  042737  001000  001212              BIC #BIT9,DTESTP        ;IT WAS ON CLEAR IT
007000  000720                              BR TEST7X
007002  052737  001000  001212  3$:         BIS #BIT9,DTESTP        ;IT WAS OFF SET IT
007010  000714                              BR TEST7X

007012  000000              SEQSAV:         0
                                    ;;*************************************************************

                                    ;TEST FOR TRANSFER FLAG AND PROGRAM NOT HUNG

007014  005037  007200          TRCK:       CLR HCNTR1              ;CLEAR HUNG COUNTER
007020  012746  000340                      MOV #PR7,-(SP)          ;RAISE INTERUPT LEVEL TO INHIBIT INTERUPTS
007024  012746  007032                      MOV #3$,-(SP)
007030  000002                              RTI
007032  105777  172150          3$:         TSTB @RXCS              ;TEST FOR TR FLAG
007036  100001                              BPL 1$                  ;SKIP NEXT INST,IF NOT SET
007040  000207                              RTS PC                  ;RETURN TO TRANSFER DATA
007042  023727  006434  000200  1$:         CMP BYTECNTR,#128.      ;HAVE 128 BYTES BEEN TRANSFERED
007050  001403                              BEQ 2$                  ;ALLOW AN INTERUPR IF EQUAL
```



```
007052  005237  007200                      INC HCNTR1              ;IF THE HUNG COUNTER OVERFLOWES EITHER
                                            ;THERE WAS NO TR FLAG OR NO INTERUPT OCCURED
                                            ;AFTER 128 BYTES TRANSFERED
007056  001365                              BNE 5$                  ;NO OVERFLOW RETEST FOR TR FLAG
007060  132777  000040  172120  2$:         BITB #DONEBIT,@RXCS     ;TEST FOR A DONE FLAG WHICH WOULD INDICATE A
                                            ;POSSABLE ERROR CONDITION EXISTS WHILE
                                            ;TRANSFERING COMMANDS OR DATA
007066  001407                              BEQ 4$                  ;IF DONE NOT SET,TYPE HUNG
007070  005037  006434                      CLR BYTECNTR            ;CLEAR BYTE COUNTER SO IT  WON'T INTERFERE
                                            ;WITH TESTING FOR DONE.
007074  005726                              TST (SP)+               ;REMOVE THIS RETURN ADDRESS FROM STACK
007076  005046                              CLR -(SP)               ;LOWER INTERUPT LEVEL TO ACCEPT AN INTERUPT
007100  012746  007106                      MOV #4$,-(SP)           ;THE INTERUPT WILL RETURN THROUGH PREVIOUSLY
007104  000002                              RTI                     ;SET ADDRESSES
007106  005237  007200          4$:         INC HCNTR1              ;ALLOW SOME TIME FOR FLAG TO SET
007112  001362                              BNE 2$
007114  104400  015432                      TYPE ,MHUNG
007120  000000              HLT21:          HALT                    ;PROGRAM HUNG, SP (177706) CONTAINS RETURN
                                            ;ADDRESS OF WAITING LOOP

                                    ;TEST FOR DONE FLAG AND PROGRAM HUNG
007122  012603              DONECK:         MOV (SP)+,R3            ;TAKE THE RETURN ADDRESS OF THE STACK INCASE
                                            ;THE PROGRAM IS EXPECTING AN INTERUPT AT DONE
007124  005037  007200                      CLR HCNTR1              ;CLEAR THE LOWER HUNG COUNTER
007130  012737  177740  007202              MOV #177740,HCNTR2      ;SET UPER COUNTER FOR 31 FULL LOOPS
007136  032777  000040  172042  2$:         BIT #DONEBIT,@RXCS      ;TEST FOR DONE FLAG
007144  001402                              BEQ 1$                  ;IF NOT SET SKIP RETURN
007146  010346                              MOV R3,-(SP)            ;PUT RETURN ADDRESS BACK ON THE STACK
007150  000207                              RTS PC                  ;RETURN
007152  062737  000001  007200  1$:         ADD #1,HCNTR1           ;WHEN THE DOUBLE WORD COUNTER OVERFLOWES
007160  005537  007202                      ADC HCNTR2              ;THE PROGRAM MUST BE HUNG
007164  005737  007202                      TST HCNTR2
007170  001362                              BNE 2$                  ;IF NO OVERFLOW RETEST FOR DONE
007172  104400  015432                      TYPE ,MHUNG
007176  000000              HLT20:          HALT                    ;PROGRAM HUNG,CONTENTS OF R3 (177703) HOLDES
                                            ;ADDRESS OF WAITING TEST.

007200  000000              HCNTR1:         0
007202  177740              HCNTR2:         177740

                                    ;AN INTERUPT DID NOT OCCURE AT A FUNCTION DONE FLAG.

007204  005237  017034          NOINTER:    INC INTER               ;INC INTERUPT ERROR COUNTER
007210  104405                              CKSWR
007212  032777  020000  171774              BIT #SW13,@SWR          ;TEST DON'T PRINT ERROR SWITCH
007220  001004                              BNE 1$
007222  104400  015260                      TYPE ,MINTER            ;TYPE NO INTERUPT ON DONE ERROR
007226  104400  015052                      TYPE ,MCRLF
007232  104405              1$:             CKSWR
007234  005777  171754                      TST @SWR                ;TEST HALT ON ERROR SWITCH
007240  100001                              BPL CONT21
007242  000000              HLT15:          HALT                    ;HALT ON ERROR
007244  004737  007250          CONT21:     JSR PC,INTSERV          ;JSR TO INTSERV AS IF IT WAS AN INTERUPT

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

```
                                        ;INTERUPT SERVICE
                                        ;
007250  117737  171734  010070  INTSERV: MOVB @RXDB,ASTAT       ;SAVE THE ERROR AND STATUS WORD
007256  005777  171724                   TST @RXCS              ;TEST THE ERROR FLAG
007262  100450                           BMI RXERROR            ;THERE WAS AN ERROR GO REPORT IT
007264  032737  000004  010070           BIT #BIT2,ASTAT        ;IS INIT DONE SET
007272  001402                           BEQ 2$                 ;NO,CONTINUE
007274  000137  007702                   JMP RXPWR              ;YES,REPORT POWER FAILED AND RESTART
007300  032737  000003  010070  2$:      BIT #3,ASTAT           ;ARE PAR OR CRC BITS SET
007306  001023                           BNE 1$                 ;YES GO LOG ERROR
007310  132777  000040  171670           BITB #DONEBIT,@RXCS    ;IS DONE SET
007316  001014                           BNE 3$                 ;IF SET RETURN TO TEST
007320  005237  017052                   INC UKNINT             ;INC UNKNOWN INTERUPT ERROR LOG
007324  104405                           CK$WR
007326  032777  020000  171660           BIT #SW13,@SWR         ;TEST DON'T PRINT ERROR SWITCH
007334  001004                           BNE 4$                 ;DON'T PRINT
007336  104400  015312                   TYPE ,MUKNINT          ;TYPE UNKNOWN INTERUPT
007342  104400  015052                   TYPE ,MCRLF
007346  000002          4$:              RTI                    ;RETURN FROM THE INTERUPT
007350  062706  000006          3$:      ADD #6,SP              ;BYPASS INTERUPT POINTERS ON STACK
007354  000207                           RTS PC                 ;RETURN TO PROGRAM
007356  005237  017030          1$:      INC NOERLOG            ;NO STATUS ERROR FLAG ERROR
007362  104405                           CK$WR
007364  032777  020000  171622           BIT #SW13,@SWR         ;TEST DON'T PRINT ERROR SWITCH
007372  001004                           BNE RXERROR
007374  104400  016034                   TYPE ,MNOFLAG          ;TYPE NO STATUS ERROR ERROR
007400  104400  015052                   TYPE ,MCRLF
                                        ;
007404  052737  000400  010704  RXERROR: BIS #BIT8,UNITSEL      ;SET HAD ERROR FLAG
007412  012737  000012  017014           MOV #10.,P2RETRY       ;SET RETRY COUNTER
007420  012777  000017  171560  2$:      MOV #RUER,@RXCS        ;GET THE ERROR CODE
007426  004737  007122                   JSR PC,DONECK          ;TEST FOR DONE FLAG
007432  032777  000002  171550           BIT #2,@RXDB           ;WAS THERE A PARITY ERROR
007440  001403                           BEQ 1$                 ;NO,CONTINUE
007442  004737  002046                   JSR PC,STATER          ;YES,GO REPORT THE PARITY ERROR
007446  000764                           BR 2$                  ;REISSUE THE FUNCTION
007450  117737  171534  010072  1$:      MOVB @RXDB,BSTAT       ;SAVE THE ERROR CODE IN B STATUS
007456  005737  010072                   TST BSTAT              ;IS THERE A DEFINITE CODE
007462  001407                           BEQ NOPRNT             ;NO , CONTINUE
007464  013705  010072                   MOV BSTAT,R5           ;ADJUST ERROR CODE TO PRODUCE AN EVEN ADDR
007470  006005                           ROR R5                 ;OF THE CORRESPONDING COUNTER
007472  006005                           ROR R5
007474  062705  017130                   ADD #ERCODE-2,R5       ;ADD ON ADDR OF ERROR LOG -2,AS THERE IS NO 0
                                        ;ERROR CODE. THE CONTENTS OF R5 WILL READJUST
                                        ;ADDRESS TO CORRECT LOG.
007500  005215                           INC (R5)               ;INC THE CORRECT ERROR CODE COUNTER
007502  104405          NOPRNT:         CK$WR
007504  032777  020000  171502           BIT #SW13,@SWR         ;TEST PRINT ERROR SWITCH IN SWR
007512  001032                           BNE 2$
007514  104400  015052                   TYPE ,MCRLF
007520  104400  015063                   TYPE ,MERHEADER        ;TYPE ERROR HEADER
007524  104400  016142                   TYPE ,MPASS            ;TYPE PASSES COMPLETED AT ERROR
007530  013737  021042  007542           MOV PASCNT,1$
007536  004537  014446                   JSR R5,SGLDEC
007542  000000          1$:              OPEN
007544  104400  015052                   TYPE ,MCRLF
```



```
007550  104400  015167                   TYPE ,MRXCS            ;TYPE COMMAND STATUS REGISTER
007554  013746  004120          MOV      FUNCTION,-(SP)  ;;SAVE FUNCTION FOR TYPEOUT
007560  104402                  TYPOS                    ;;GO TYPE--OCTAL ASCII
007562     003                  .BYTE    3               ;;TYPE 3 DIGIT(S)
007563     000                  .BYTE    0               ;;SUPPRESS LEADING ZEROS
007564  004737  007716                   JSR PC,TYPADR          ;TYPE ADDRESSES AND RUN CONDITIONS
007570  104400  015052                   TYPE ,MCRLF
007574  004737  010250                   JSR PC,TYPCODE         ;PRINT THE STATUS REGISTERS
007600  032737  000020  004120  2$:      BIT #BIT4,FUNCTION     ;WHAT DRIVE IS BEING USED
007606  001006                           BNE 6$
007610  012777  000013  171370           MOV #RU0STAT,@RXCS     ;DRIVE 0 BEING USED
007616  004737  007122          5$:      JSR PC,DONECK          ;TEST FOR DONE FLAG
007622  000404                           BR 7$
007624  012777  000033  171354  6$:      MOV #RU1STAT,@RXCS     ;DRIVE 1 BEING USED
007632  000771                           BR 5$
007634  032777  000002  171346  7$:      BIT #2,@RXDB           ;WAS THERE A PARITY ERROR
007642  001403                           BEQ 3$                 ;NO,CONTINUE
007644  004737  002046                   JSR PC,STATER          ;YES,REPORT THE ERROR
007650  000753                           BR 2$                  ;REISSUE THE COMMAND
007652  105777  171332          3$:      TSTB @RXDB             ;WAS DRIVE READY SET
007656  100406                           BMI 4$                 ;YES RETURN
007660  104400  015016                   TYPE ,MNODRY
007664  104400  015052                   TYPE ,MCRLF
007670  004737  010074                   JSR PC,DELUNIT         ;NO GO DELETE THAT UNIT
007674  062706  000004          4$:      ADD #4,SP              ;MOVE ERROR RETURN TO TOP OF STACK
007700  000207                           RTS PC
                                        ;
007702  104400  016344          RXPWR:   TYPE ,MRX11            ;ONLY THE RX11 POWER HAS FAILED
007706  104400  014136                   TYPE ,SPOWER           ;PRINT POWER FAILED
007712  000137  002700                   JMP RESTART            ;GO TO RESTART
                                        ;
                                        ;
007716  104400  014555          TYPADR:  TYPE ,MTRK             ;TYPE TRACK ADDRESS
007722  013737  011360  007734           MOV TARGET,3$
007730  004537  014446                   JSR R5,SGLDEC
007734  000000          3$:              OPEN
007736  104400  014604                   TYPE ,MSECT            ;TYPE SECTOR ADDRESS
007742  013737  012310  007762           MOV TSECTOR,2$
007750  042737  177740  007762           BIC #177740,2$         ;CLEAR ALL BUT SECTOR ADDRESS
007756  004537  014446                   JSR R5,SGLDEC
007762  000000          2$:              OPEN
007764  104400  015047                   TYPE ,DBLSP
007770  032737  000020  010704           BIT #BIT4,UNITSEL      ;WHITCH DRIVE IS BEING USED
007776  001003                           BNE 1$
010000  104400  015106                   TYPE ,MUNIT0           ;TYPE UNIT 0
010004  000402                           BR TYPSEL
010006  104400  015116          1$:      TYPE ,MUNIT1           ;TYPE UNIT 1
010012  104400  015244          TYPSEL:  TYPE ,MPAT             ;TYPE PATTERN CODE
010016  013746  010420                   MOV PAT,-(SP)   ;;SAVE PAT FOR TYPEOUT
010022  104402                  TYPOS                    ;;GO TYPE--OCTAL ASCII
010024     001                  .BYTE    1               ;;TYPE 1 DIGIT(S)
010025     000                  .BYTE    0               ;;SUPPRESS LEADING ZEROS
010026  104400  015047                   TYPE ,DBLSP
010032  104400  015250                   TYPE ,MTEST            ;TYPE TEST CODE
010036  013746  002140                   MOV TEST,-(SP)  ;;SAVE TEST FOR TYPEOUT
010042  104402                  TYPOS                    ;;GO TYPE--OCTAL ASCII
```

```
1569  010044  001                             .BYTE   1               ;;TYPE 1 DIGIT(S)
1570  010045  000                             .BYTE   0               ;;SUPPRESS LEADING ZEROS
1571  010046  104400  015047                  TYPE ,DBLSP
1572  010052  104400  015254                  TYPE ,MSEQ              ;TYPE SEQUENCE CODE
1573  010056  013746  011370          MOV     SEQUEN,-(SP)    ;;SAVE SEQUEN FOR TYPEOUT
1574  010062  104402                   TYPOS                   ;;GO TYPE--OCTAL ASCII
1575  010064  001                             .BYTE   1               ;;TYPE 1 DIGIT(S)
1576  010065  000                             .BYTE   0               ;;SUPPRESS LEADING ZEROS
1577  010066  000207                          RTS PC
1578
1579  010070  000000          ASTAT:          0
1580  010072  000000          BSTAT:          0
1581
1582  010074  132737  000020  004120  DELUNIT:        BITB #BIT4,FUNCTION     ;TEST FOR PRESENTLY USED UNIT
1583  010102  001016                           BNE 3$
1584  010104  104400  015106                  TYPE ,MUNIT0            ;UNIT 0 HAS BEEN DELETED FROM USE
1585  010110  104400  015126                  TYPE ,MDELET
1586  010114  104400  015240                  TYPE ,DBLLF
1587  010120  042737  000200  010704          BIC #BIT7,UNITSEL       ;CLEAR UNIT 0 SELECTION BIT
1588  010126  005737  010704                  TST UNITSEL             ;WAS UNIT 1 SELECTED FOR USE
1589  010132  100020                           BPL 1$                  ;UNIT 1 NOT SELECTED GO DUMP ERROR REPORT
1590  010134  000137  002726          2$:     JMP TESTSEL             ;CONTINUE ON OTHER UNIT AT BEGINING OF TEST
1591  010140  104400  015116          3$:     TYPE ,MUNIT1            ;UNIT 1 HAS BEEN DELETED FROM USE
1592  010144  104400  015126                  TYPE,MDELET
1593  010150  104400  015240                  TYPE ,DBLLF
1594  010154  042737  100000  010704          BIC #BIT15,UNITSEL      ;CLEAR UNIT 1 SELECTION BIT
1595  010162  105737  010704                  TSTB UNITSEL            ;WAS UNIT 0 SELECTED FOR USE
1596  010166  100762                           BMI 2$                  ;YES CONTINUE ON UNIT 0
1597  010170  005137  022514                  COM SHORTRPT            ;SET SHORT REPORT FLAG
1598  010174  000137  021062          1$:     JMP ERDUMP              ;CAN'T CONTINUE TYPE OUT STATISTICAL REPORT
1599
1600
1601  010200  117737  171004  010070  RDCODE:         MOVB @RXDB,ASTAT        ;SAVE THE A STATUS
1602  010206  012777  000017  170772  2$:             MOV #RDER,@RXCS         ;READ THE B STATUS REGISTER
1603  010214  004737  007122                          JSR PC,DONECK           ;WAIT FOR DONE FLAG
1604  010220  032777  000002  170762                  BIT #2,@RXDB            ;WAS THERE A PARITY ERROR
1605  010226  001403                                  BEQ 1$                  ;NO,CONTINUE
1606  010230  004737  002046                          JSR PC,STATER           ;YES,REPORT THE PARITY ERROR
1607  010234  000764                                  BR 2$                   ;RETRY READING STATUS B
1608  010236  117737  170746  010072  1$:             MOVB @RXDB,BSTAT        ;SAVE THE B STATUS CODES
1609  010244  104400  015052                          TYPE ,MCRLF
1610  010250  104400  015210          TYPCODE:        TYPE ,MASTAT            ;TYPE THE CONTENTS OF THE TWO STATUS REGISTERS
1611  010254  013746  010070                  MOV     ASTAT,-(SP)     ;;SAVE ASTAT FOR TYPEOUT
1612  010260  104402                           TYPOS                   ;;GO TYPE--OCTAL ASCII
1613  010262  003                              .BYTE   3               ;;TYPE 3 DIGIT(S)
1614  010263  000                              .BYTE   0               ;;SUPPRESS LEADING ZEROS
1615  010264  104400  015040                  TYPE ,TAB
1616  010270  104400  015224                  TYPE ,MBSTAT
1617  010274  013746  010072                  MOV     BSTAT,-(SP)     ;;SAVE BSTAT FOR TYPEOUT
1618  010300  104402                           TYPOS                   ;;GO TYPE--OCTAL ASCII
1619  010302  003                              .BYTE   3               ;;TYPE 3 DIGIT(S)
1620  010303  000                              .BYTE   0               ;;SUPPRESS LEADING ZEROS
1621  010304  104400  015052                  TYPE ,MCRLF
1622  010310  000207                          RTS PC
1623
1624
```



```
1625                                    ;;*****************************************************************
1626
1627                                            ;DECODES THE DATA PATERN SELECTED IN THE SWR
1628                                            ;
1629                                            ;BITS 6,7,AND 8 OF THE INITIAL SWR SELECTED THE DATA PATTERN
1630                                            ;TO BE USED AS FOLLOWS:
1631                                            ;
1632                                            ;   BITS                    DATA PATTERN
1633                                            ; 8  7  6
1634                                            ; 0  0  0               NO PATTERN SPECIFIED (FORCE RANDOM DATA)
1635                                            ; 0  0  1               ALL ZEROS
1636                                            ; 0  1  0               ALL ONES
1637                                            ; 0  1  1               FLOATING ZERO
1638                                            ; 1  0  0               FLOATING ONE
1639                                            ; 1  0  1               ALTERNATING BITS
1640                                            ; 1  1  0               ALTERNATING PAIRS OF BITS
1641                                            ; 1  1  1               RANDOM
1642                                            ;
1643                                            ;NOTE: ALL DATA PATTERNS WILL BE MODIFIED SO THE FIRST BYTE WILL
1644                                            ;CONTAIN THE TRACK ADDRESS, THE SECOND BYTE WILL CONTAIN THE UNIT
1645                                            ;NUMBER AND SECTOR ADDRESS IN WHICH THE DATA IS WRITTEN, THE LAST
1646                                            ;TWO BYTES CONTAIN THE CHECK SUM NUMBERS.
1647                                    ;;*****************************************************************
1648
1649
1650
1651  010312  142737  000377  010360  GETPATTERN:     BICB #377,@#BRONPAT     ;CLEAR BRANCH OFFSET
1652  010320  005037  010612                          CLR SUM                 ;SET UP FOR ACCUMULATION OF CHECK SUM
1653  010324  005737  010420                          TST PAT                 ;IF NO PATTERN SPECIFIED FORCE PATTERN 7
1654  010330  001003                                  BNE 1$
1655  010332  012737  000007  010420                  MOV #7,PAT
1656  010340  013704  010420          1$:             MOV PAT,R4              ;GET PATTERN BITS
1657  010344  005304                                  DEC R4                  ;ADJUST FOR CORRECT OFFSET
1658  010346  006304                                  ASL R4
1659  010350  150437  010360                          BISB R4,@#BRONPAT       ;INSERT OFFSET
1660  010354  012704  016606                          MOV #BUFADR+2,R4        ;SET UP ADDRESS OF FIRST BYTE
1661  010360  000777          BRONPAT:        BR .                    ;BRANCH BY OFFSET SELECTED
1662  010362  000137  010422                          JMP DATA0               ;000 DATA BYTE
1663  010366  000137  010434                          JMP DATA1               ;377 DATA BYTE
1664  010372  000137  010444                          JMP FLOAT0              ;FLOAT A 0 THROUGH ALL 1'S
1665  010376  000137  010506                          JMP FLOAT1              ;FLOAT A 1 THROUGH ALL 0'S
1666  010402  000137  010514                          JMP PAT125              ;125/052 DATA WORD
1667  010406  000137  010534                          JMP PAT314              ;314/063 DATA WORD
1668  010412  000137  010544                          JMP RANDATA             ;RANDOM DATA BYTE
1669
1670
1671  010416  000000          DATABYTE:       0
1672  010420  000000          PAT:            0
1673
1674
1675                                    ;;*****************************************************************
1676
1677                                            ;LOAD SOFTWARE BUFFER WITH ALL ZEROS
1678                                            ;   P = 1
1679
1680  010422  005037  010416          DATA0:          CLR DATABYTE
```

```
1681  010426  004737  010564          PATGEN:         JSR PC,LOAD             ;GO LOAD THE DATA BUFFER
1682  010432  000775                                  BR PATGEN
1683
1684                                  ;;**************************************************************
1685
1686                                          ;LOAD SOFTWARE BUFFER WITH ALL ONES
1687                                          ;   P = 2
1688
1689
1690  010434  112737  000377  010416  DATA1:          MOVB #377,DATABYTE
1691  010442  000771                                  BR PATGEN
1692
1693                                  ;;**************************************************************
1694
1695                                          ;FLOAT A 0 THROUGH ONES IN SOFTWARE BUFFER
1696                                          ;   P = 3
1697
1698
1699  010444  112737  000376  010416  FLOAT0:         MOVB #376,DATABYTE      ;SET UP A ONES FIELD
1700  010452  000261          XPATGEN:        SEC                     ;SET THE C BIT TO ROTATE THROUGH THE DATA
1701  010454  012702  000000          1$:             MOV #0,R2               ;CLR R2 (CAN'T USE "CLR" AS IT CLEARS "C" BIT)
1702  010460  103001                                  BCC 2$                  ;BR IF THE "C" BIT IS CLEARED
1703  010462  005202                                  INC R2                  ;SET R2 IF NOT
1704  010464  004737  010564          2$:             JSR PC,LOAD             ;GO LOAD THE DATA BUFFER
1705  010470  000241                                  CLC
1706  010472  005702                                  TST R2                  ;IS R2 NONZERO
1707  010474  001401                                  BEQ 3$
1708  010476  000261                                  SEC                     ;YES, SET THE "C" BIT
1709  010500  106137  010416          3$:             ROLB DATABYTE
1710  010504  000763                                  BR 1$
1711
1712                                  ;;**************************************************************
1713
1714                                          ;FLOAT A 1 THROUGH ALL ZEROS IN SOFTWARE BUFFER
1715                                          ;   P = 4
1716
1717  010506  005037  010416          FLOAT1:         CLR DATABYTE
1718  010512  000757                                  BR XPATGEN
1719
1720                                  ;;**************************************************************
1721
1722                                          ;LOAD SOFTWARE BUFFER WITH ALTERNATING 1 AND 0 FOR
1723                                          ;ONE BYTE AND THE COMPLIMENT INTO THE NEXT
1724                                          ;   P = 5
1725
1726  010514  112737  000125  010416  PAT125:         MOVB #125,DATABYTE
1727  010522  004737  010564          XXPATGEN:       JSR PC,LOAD
1728  010526  105137  010416                          COMB DATABYTE           ; * (SEE NOTE BELOW )
1729  010532  000773                                  BR XXPATGEN
1730
1731                                          ; * NOTE:   TO MAKE PATTERN 5 AND 6 COMPATABLE WITH THE RX8 PATTERNS
1732                                          ;NOP THIS INSTRUCTION. (CHANGE THESE 2 LOCATIONS TO 000240)
1733                                          ;THIS CHANGE IS FOR INTERPROCESSOR RX** COMPATABILITY TESTING ONLY.
1734
1735                                  ;;**************************************************************
1736
```



```
1737                                          ;LOAD SOFTWARE BUFFER WITH ALTERNATING PAIRS OF 1 AND 0 AND
1738                                          ;COMPLIMENT INTO THE NEXT
1739                                          ;   P = 6
1740
1741  010534  112737  000314  010416  PAT314:         MOVB #314,DATABYTE
1742  010542  000767                                  BR XXPATGEN
1743
1744                                  ;;**************************************************************
1745
1746                                          ;LOAD SOFTWARE BUFFER WITH RANDOM DATA PATTERN
1747                                          ;   P = 0 OR 7
1748
1749  010544  004737  012122          RANDATA:        JSR PC,RANGEN           ;GET RANDOM NUMBER
1750  010550  113737  012214  010416                  MOVB RANUM,DATABYTE
1751  010556  004737  010564                          JSR PC,LOAD
1752  010562  000770                                  BR RANDATA
1753
1754  010564  063737  010416  010612  LOAD:           ADD DATABYTE,SUM        ;ACCUMULATE THE PATTERN CHECK SUM
1755  010572  113724  010416                          MOVB DATABYTE,(R4)+     ;LOAD THE DATA BUFFER
1756  010576  022704  017002                          CMP #BUFADR+176,R4      ;HAVE 124 BYTES BEEN GENERATED
1757  010602  001401                                  BEQ 1$                  ;IF YES,RETURN TO TEST
1758  010604  000207                                  RTS PC                  ;IF NO,RETURN TO PATTERN GENERATOR
1759  010606  005726          1$:             TST (SP)+               ;TAKE PATTERN RETURN ADDRESS OF STACK
1760  010610  000207                                  RTS PC                  ;RETURN TO TEST
1761
1762  010612  000000          SUM:            0
1763
1764                                  ;;**************************************************************
1765
1766                                          ;TEST FOR SELECTED UNITS,DRIVE READY,AND USED CONDITIONS
1767                                          ;ALSO CONTAINS A "HAD ERROR" FLAG TO BE TESTED AT EOP.
1768                                          ;THE BITS IN UNITSEL ARE USED AS FOLLOWS
1769                                          ;
1770                                          ;BIT15  =UNIT 1 SELECTED VIA SWR
1771                                          ;BIT14  =UNIT 1 USED BIT
1772                                          ;BIT8   =THIS PASS HAD AN ERROR
1773                                          ;BIT7   =UNIT 0 SELECTED VIA SWR
1774                                          ;BIT6   =UNIT 0 USED BIT
1775                                          ;BIT4   =UNIT SELECTION FOR FUNCTION WORD
1776
1777                                  ;;**************************************************************
1778
1779  010614  032737  000100  010704  GETUNIT:        BIT #BIT6,UNITSEL       ;WAS UNIT 0 JUST USED
1780  010622  001012                                  BNE 1$                  ;UNIT 0 USED CHECK UNIT 1
1781  010624  105737  010704                          TSTB UNITSEL            ;WAS UNIT 0 SELECTED
1782  010630  100007                                  BPL 1$                  ;NO GO TO UNIT 1
1783  010632  042737  040020  010704                  BIC #40020,UNITSEL      ;CLEAR UNIT 1 USED BIT AND FUNCTION UNIT BIT
1784  010640  052737  000100  010704                  BIS #BIT6,UNITSEL       ;SET UNIT 0 USED BIT
1785  010646  000207                                  RTS PC
1786  010650  005737  010704          1$:             TST UNITSEL             ;WAS UNIT 1 SELECTED
1787  010654  100012                                  BPL 2$                  ;NO RETURN
1788  010656  032737  040000  010704                  BIT #BIT14,UNITSEL      ;HAS UNIT 1 BEEN USED
1789  010664  001006                                  BNE 2$                  ;YES RETURN
1790  010666  042737  000100  010704                  BIC #BIT6,UNITSEL       ;CLEAR UNIT 0 USED BIT
1791  010674  052737  040020  010704                  BIS #40020,UNITSEL      ;SET UNIT 1 USED BIT AND FUNCTION UNIT BIT
1792  010702  000207          2$:             RTS PC
```

```
010704  000000                          UNITSEL:        0

                                        ;TEST THAT ALL UNITS HAVE BEEN ACCESSED

010706  005737  010704                  DONE:           TST UNITSEL             ;IS UNIT 1 SELECTED
010712  100006                                          BPL 1$                  ;NO RETURN
010714  032737  040000  010704                          BIT #BIT14,UNITSEL      ;YES HAS IT BEEN USED
010722  001002                                          BNE 1$                  ;YES RETURN
010724  062706  000002                                  ADD #2,SP               ;BYPASS NOT DONE RETURE ON STACK
010730  000207                          1$:             RTS PC


010732  004737  010706                  STOP:           JSR PC,DONE             ;HAVE ALL DRIVES BEEN USED
010736  005037  007200                                  CLR HCNTR1
010742  005237  021042                                  INC PASCNTR             ;KEEP TOTAL NUMBER OF PASSES
010746  005737  011142                                  TST CHARCT              ;HAVE 72 EOP INDICATORS BEEN TYPED
010752  001005                                          BNE 3$
010754  104400  015052                                  TYPE ,MCRLF             ;YES DO A CRLF
010760  012737  177670  011142                          MOV #-72.,CHARCT        ;RESET CHARACTER COUNTER
010766  005237  011142                  3$:             INC CHARCT
010772  032737  000400  010704                          BIT #BIT8,UNITSEL       ;TEST HAD ERROR FLAG
011000  001403                                          BEQ 2$                  ;NO ERROR PRINT D
011002  104400  015057                                  TYPE ,MEREOP            ;HAD ERROR PRINT ?
011006  000404                                          BR 1$
011010  104400  015055                  2$:             TYPE ,MEOP              ;PRINT EOP INDICATOR
011014  104400  015061                                  TYPE ,MABELL
011020  104405                          1$:             CKSWR
011022  032777  040000  170164                          BIT #SW14,@SWR          ;TEST EOP HALT SWITCH
011030  001417                                          BEQ CONT22
011032  104400  015052                                  TYPE ,MCRLF
011036  104400  016142                                  TYPE ,MPASS             ;AT HALT ON PASS PRINT NUMBER OF PASSES COMPLETE
011042  013737  021042  011054                          MOV PASCNTR,6$
011050  004537  014446                                  JSR R5,SGLDEC
011054  000000                          6$:             OPEN
011056  104400  015052                                  TYPE ,MCRLF
011062  000000                          HLT16:          HALT
011064  005037  011142                                  CLR CHARCT              ;THIS CAUSES A CRLF AFTER HALT AT E.O.P.
011070  042737  040500  010704  CONT22:                 BIC #40500,UNITSEL      ;CLEAR USED BITS, AND "HAD ERROR" BIT
011076  022737  000007  010420                          CMP #7,PAT              ;IF RANDOM DATA GET NEW DATA FOR BUFFER
011104  001002                                          BNE 2$
011106  004737  010312                                  JSR PC,GETPATTERN
011112  005237  007200                  2$:             INC HCNTR1              ;WAIT FOR THE EOP INDICATOR TO BE PRINTED
011116  001375                                          BNE 2$
011120  013705  000042                                  MOV @#42,R5             ;ACT11 END OF PASS HOOKS
011124  001405                                          BEQ HERE
011126  000005                                          RESET
011130  004715                          LOGICAL:        JSR PC,(R5)
011132  000240                                          NOP
011134  000240                                          NOP
011136  000240                                          NOP
011140  000207                          HERE:           RTS PC

011142  000000                          CHARCT:         0
```



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

                                                ;INITIALIZE TRACK SEQUENCE


011144  042737  100200  001200  INITTRACK:      BIC #100200,OD          ;CLEAR FIRST USED BITS
011152  005737  001200                          TST OD                  ;TEST CONTENTS OF ID,OD FOR 0
011156  001005                                  BNE 1$                  ;ID,OD SPECIFIED USE THEM
011160  112737  000114  001201                  MOVB #114,ID            ;NONE SPECIFIED SET ID TO MAXIMUM
011166  105037  001200                          CLRB OD                 ;SET OD TO MINIMUM
011172  113737  001200  011360  1$:             MOVB OD,TARGET          ;INIT OD AS PRESENT TRACK
011200  005037  011366                          CLR XID                 ;INIT WORKING ID AND OD LOCATIONS
011204  113737  001201  011366                  MOVB ID,XID
011212  005037  011364                          CLR XOD
011216  113737  001200  011364                  MOVB OD,XOD
011224  013737  011366  011356                  MOV XID,TRKCNTR         ;SET UP NUMBER OF TRACK MOVEMENTS
011232  163737  011364  011356                  SUB XOD,TRKCNTR
011240  005237  011356                          INC TRKCNTR
011244  052737  100200  001200                  BIS #100200,OD          ;SET FIRST TIME BITS IN ID,OD
011252  000207                                  RTS PC



                                        ;;*************************************************************

                                                ;TEST FOR HEAD SEQUENCE SELECTED BY INITIAL SWR SETTING
                                                ;
                                                ;BITS 0,1,AND 2 OF THE INITIAL SWR SELECTED THE TRACK SEQUENCING
                                                ;TO BE FOLLOWED AS INITICATED BELOW
                                                ;
                                                ;   BITS                SEQUENCE
                                                ; 0  0  0               NO SEQUENCE SPECIFIED (DEFAULT TO SEQ 7)
                                                ; 0  0  1               INCREMENT FROM OD TO ID
                                                ; 0  1  0               DECREMENT FROM ID TO OD
                                                ; 0  1  1               DO PREVIOUS 2 SEQUENCES
                                                ; 1  0  0               BOUNCE BETWEEN ID AND OD
                                                ; 1  0  1               DECREASING BOUNCE
                                                ; 1  1  0               STROBE BETWEEN OD AND DECREMENTING ID
                                                ; 1  1  1               RANDOM TRACK SELECTION

                                        ;;*************************************************************

011254  113737  011360  011362  GETTRACK:       MOVB TARGET,PRESTRK     ;RESET TO PRESENT TRACK
011262  142737  000377  011320                  BICB #377,@#BRONTRK     ;CLEAR OUT BRANCH OFFSET
011270  005737  011370                          TST SEQUEN              ;IF NO SEQUENCE SPECIFIED FORCE SEQUENCE 7
011274  001003                                  BNE 1$
011276  012737  000007  011370                  MOV #7,SEQUEN
011304  013704  011370          1$:             MOV SEQUEN,R4           ;GET SEQUENCE BITS
011310  005304                                  DEC R4                  ;ADJUST FOR CORRECT OFFSET
011312  006304                                  ASL R4
011314  150437  011320                          BISB R4,@#BRONTRK       ;THIS BR INST,IS MODIFIED BY THE TEST CONDITIONS
                                                                        ;SELECTED ,TO BRANCH TO ONE OF THE SEQ. BELOW

011320  000777                  BRONTRK:        BR .                    ;BRANCH BY OFFSET SELECTED
011322  000137  011372                          JMP SEQ1                ;INCREMENT
```

```
                                        JMP SEQ2                ;DECREMENT
                                        JMP SEQ3                ;INCREMENT/DECREMENT
                                        JMP SEQ4                ;BOUNCE ID TO OD
                                        JMP SEQ5                ;DECREASING BOUNCE
                                        JMP SEQ6                ;STROBE
                                        JMP SEQ7                ;RANDOM

                        TRKCNTR:        0
                        TARGET:         0
                        PRESTRK:        0
                        XOD:            0
                        XID:            0
                        SEQUEN:         0

                        ;;**************************************************************

                                ;INCREMENT FROM OD+1 TO ID AND RETURN TO OD
                                ;   S = 1

                        SEQ1:           BIC #100200,OD          ;CLEAR FIRST TIME BITS
                                        CMPB XID,PRESTRK        ;PRESENT TRACK EQUAL TO ID
                                        BNE 1$                  ;NO GET NEW TRACK
                                        MOVB OD,TARGET          ;YES RETURN TO OD
                                        BR NEWTRK               ;NEWTRK INCREMENTS THE TRACK MOVED TO COUNTER
                                                                ;AND RETURNS TO NEXT TRACK
                        1$:             INC TARGET              ;ADD 1 TO TARGET TRACK
                                        BR NEWTRK

                        ;;**************************************************************

                                ;DECREMENT FROM ID TO OD
                                ;   S = 2

                        SEQ2:           TSTB ID                 ;FIRST TIME BIT SET
                                        BPL 1$                  ;NO GET NEXT TRACK
                                        BIC #100200,OD          ;YES CLEAR FIRST TIME BITS
                                        MOVB XID,TARGET         ;GO TO ID
                                        BR NEWTRK
                        1$:             DEC TARGET              ;MOVE TO NEXT TRACK
                                        BR NEWTRK

                        ;;**************************************************************

                                ;INCREMENT THEN DECREMENT TRACKS
                                ;   S = 3

                        SEQ3:           TSTB OD                 ;CHECK FIRST TIME BIT
                                        BPL 1$                  ;NOT FIRST TIME THROUGH
                                        BIC #100200,OD          ;CLEAR FIRST TIME BITS
                                        DEC TRKCNTR
                                        ASL TRKCNTR             ;RESET TRACK COUNTER TO DOUBLE THE COUNT
                        1$:             MOV XID,R0
                                        SUB XOD,R0              ;GET DIFFERENCE BETWEEN ID AND OD
                                        SUB TRKCNTR,R0          ;IS TRACK COUNTER HALF DONE
                                        BGE SEQ2                ;YES GO DECREMENT TRACKS
                                        BR SEQ1                 ;NO CONTINUE INCREMENTING TRACKS
```



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

                                ;BOUNCE BETWEEN ID AND OD ONLY
                                ;   S = 4

                        SEQ4:           BIC #100200,OD          ;CLEAR THE FIRST TIME BITS
                                        CMPB PRESTRK,OD         ;DID IT JUST DO OD
                                        BEQ IDNEXT              ;YES
                                        MOVB OD,TARGET          ;NO GO TO OD TRACK
                                        BR NEWTRK
                        IDNEXT:         MOVB ID,TARGET          ;GO DO ID TRACK
                                        BR NEWTRK

                        ;;**************************************************************

                                ;INCREMENT THE "HEAD MOVED TO" COUNTERS AND RETURN
                                ;THROUGH HERE FROM ALL TRACK SEQUENCES

                        NEWTRK:         MOV TARGET,R5           ;GET TRACK ADDRESS
                                        ASL R5                  ;MULTIPLY BY 4 TO DOUBLE PRECISION
                                        ASL R5                  ;INTERLEAVE COUNTER ADDRESSES
                                        ADD #HOMOVE,R5          ;ADD ON ADDRESS OF HEAD MOVE COUNTER
                                        ADD #1,(R5)             ;INC COUNTER
                                        ADC 2(R5)               ;ADD CARRY TO HIGH ORDER WORD
                                        RTS PC                  ;RETURN TO ACCESS NEXT TRACK

                        ;;**************************************************************

                                ;BOUNCE BETWEEN DECREASING ID AND INCREASING OD
                                ;   S = 5

                        SEQ5:           TSTB ID                 ;TEST FIRST TIME BIT OF ID
                                        BPL 1$
                                        INC TRKCNTR             ;ADJUST FOR ONE EXTRA MOVEMENT TO TRK 00
                                        BICB #200,ID            ;CLEAR FIRST TIME BIT OF ID
                                        MOVB XID,TARGET         ;MOVE TO ID
                                        BR NEWTRK
                        1$:             TSTB OD                 ;TEST FIRST TIME BIT OF OD
                                        BPL 2$
                                        BICB #200,OD            ;CLEAR FIRST TIME BIT OF OD
                                        MOVB XOD,TARGET         ;MOVE TO OD
                                        BR NEWTRK
                        2$:             BITB #1,TRKCNTR         ;TEST COUNTER FOR ODD OR EVEN
                                        BNE 3$                  ;ODD MOVE TO NEW ID
                                        INC XOD                 ;EVEN ADD 1 TO OD
                                        MOVB XOD,TARGET
                                        BR LASTTRK              ;SEE IF LAST TRACK IS BEING ACCESSED
                        3$:             DEC XID                 ;SUBTRACT 1 FROM ID
                                        MOVB XID,TARGET
                        LASTTRK:        CMPB TRKCNTR,#1
                                        BNE XRETURN
                                        MOVB OD,TARGET          ;THIS IS LAST TRACK RETURN TO OD
                        XRETURN:        BR NEWTRK
```

```
                                        ;;*****************************************************************
                                                ;STROBE BETWEEN OD AND DECREASING ID
                                                ;   S = 6
                                        
011742  105737  001200          SEQ6:         TSTB OD                 ;CHECK FIRST TIME BIT
011746  100007                                BPL 1$                  ;NOT FIRST TIME
011750  042737  100200  001200                BIC #100200,OD          ;WAS FIRST TIME,CLEAR THE BITS
011756  005337  011356                        DEC TRKCNTR             ;RESET COUNTER TO DOUBLE THE NUMBER OF TRACKS
011762  006337  011356                        ASL TRKCNTR
011766  123737  011362  001200  1$:           CMPB PRESTRK,OD         ;WAS OD JUST ACCESSED
011774  001006                                BNE ODNEXT              ;NO GO TO OD
011776  113737  011366  011360                MOVB XID,TARGET         ;DECREASING ID IS NEXT TRACK
012004  005337  011366                        DEC XID
012010  000664                                BR NEWTRK
012012  113737  001200  011360  ODNEXT:       MOVB OD,TARGET          ;OD IS NEXT TRACK
012020  000660                                BR NEWTRK
                                        
                                        
                                        
                                        ;;*****************************************************************
                                                ;RANDOM SEQUENCING OF TRACKS
                                                ;FOR PROPER USE OF RANDOM TRACK SEQUENCING, THE OD/ID LIMITS
                                                ;SHOULD NOT BE SET FOR LESS THAN HALF THE TRACKS.
                                                ;   S = 0 OR 7
                                        
012022  042737  100200  001200  SEQ7:         BIC #100200,OD          ;CLEAR THE FIRST TIME BITS
012030  004737  012122                        JSR PC,RANGEN           ;GET A RANDOM NUMBER
012034  042737  177600  012214                BIC #177600,RANUM       ;CLEAR ALL BUT LOW 7 BITS
012042  123737  012214  011366  IDCOMP:       CMPB RANUM,XID          ;IS RANUM LARGER THAN ID ADDRESS
012050  003404                                BLE ODCOMP              ;NO,RANUM IS LESS OR EQUAL TO ID
012052  163737  011366  012214                SUB XID,RANUM           ;YES,SUBTRACT ID FROM IT
012060  000770                                BR IDCOMP               ;SEE IF RANUM IS NOW OK
012062  123737  012214  011364  ODCOMP:       CMPB RANUM,XOD          ;IS RANUM SMALLER THAN OD ADDRESS
012070  002004                                BGE PRESCHK             ;NO,RANUM IS GREATER OR EQUAL TO OD
012072  063737  011364  012214                ADD XOD,RANUM           ;YES,ADD OD TO RANUM
012100  000770                                BR ODCOMP               ;SEE IF RANUM IS NOW OK
012102  123737  012214  011362  PRESCHK:       CMPB RANUM,PRESTRK      ;IF RANUM EQUALS PRESENT TRACK
012110  001744                                BEQ SEQ7                ;GET ANOTHER RANDOM NUMBER
012112  013737  012214  011360                MOV RANUM,TARGET        ;RANUM OK PUT IT IN TARGET TRACK
012120  000620                                BR NEWTRK
                                        
012122  012700  000001          RANGEN:       MOV #1,R0
012126  063700  012210                        ADD RAN1,R0
012132  063700  012212                        ADD RAN2,R0
012136  042700  170000                        BIC #170000,R0
012142  000241                                CLC
012144  006100                                ROL R0
012146  006100                                ROL R0
012150  310037  012210                        MOV R0,RAN1
012154  005000                                CLR R0
012156  013700  012212                        MOV RAN2,R0
012162  006000                                ROR R0
012164  006000                                ROR R0
012166  063700  012210                        ADD RAN1,R0
```



```
012172  042700  170000                        BIC #170000,R0
012176  010037  012212                        MOV R0,RAN2
012202  010037  012214                        MOV R0,RANUM
012206  000207                                RTS PC
                                        
012210  000000                  RAN1:         0
012212  000000                  RAN2:         0
012214  000000                  RANUM:        0
                                        
                                        ;;*****************************************************************
                                                ;SECTOR INITIALIZATION AND SELECTION
                                        
012216  005737  001202          INITSECTOR:   TST FIRST               ;TEST FIRST AND LAST FOR 0
012222  001005                                BNE 1$                  ;SECTORS SPECIFIED USE THEM
012224  005237  001202                        INC FIRST               ;NONE SPECIFIED SET FIRST TO 1
012230  112737  000032  001203                MOVB #32,LAST           ;SET LAST TO MAXIMUM
012236  113737  001203  012306  1$:           MOVB LAST,SECCNTR       ;SET UP SECTOR COUNTER
012244  163737  001202  012306                SUB FIRST,SECCNTR
012252  005237  012306                        INC SECCNTR
012256  105037  012307                        CLRB SECCNTR+1
012262  113737  001202  012310                MOVB FIRST,TSECTOR      ;PUT FIRST SECTOR IN TARGET SECTOR
012270  163737  012314  012310                SUB THREE,TSECTOR       ;SUB 3 FROM TSECTOR AS FIRST TIME THROUGH
                                                                        ;IT GETS ADDED BACK ON.
012276  012737  000001  012312                MOV #1,INTLEAV          ;SET INTERLEAVE OFFSET
012304  000207                                RTS PC
                                        
012306  000000                  SECCNTR:      0
012310  000000                  TSECTOR:      0
012312  000000                  INTLEAV:      0
012314  000003                  THREE:        3
                                        
012316  063737  012314  012310  GETSECTOR:    ADD THREE,TSECTOR       ;ADD 3 FOR INTERLEAVING
012324  123737  001203  012310                CMPB LAST,TSECTOR
012332  002010                                BGE 1$                  ;NEW SECTOR IS WITHIN LIMITS
012334  113737  001202  012310                MOVB FIRST,TSECTOR      ;RESET TARGET SECTOR TO INTERLEAVE
012342  063737  012312  012310                ADD INTLEAV,TSECTOR     ;ADD ON INTERLEAVE OFFSET VALUE
012350  005237  012312                        INC INTLEAV             ;UP DATE THE OFFSET VALUE
012354  000207                  1$:           RTS PC
                                        
                                        
                                        .SBTTL  TYPE ROUTINE
                                        
                                        ;;*****************************************************************
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
```

```
2129                                    ;*
2130
2131  012356  105737  012605          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
2132  012362  100002                          BPL     1$              ;;BR IF YES
2133  012364  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
2134  012366  000407                          BR      3$              ;;LEAVE
2135  012370  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
2136  012372  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
2137  012376  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
2138  012400  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
2139  012402  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
2140  012404  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
2141  012406  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
2142  012412  000002                          RTI                     ;;RETURN
2143  012414  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
2144  012420  001430                          BEQ     8$
2145  012422  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
2146  012426  001006                          BNE     5$
2147  012430  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
2148  012432  104400                          TYPE                    ;;TYPE A CR AND LF
2149  012434  012607                          $CRLF
2150  012436  105037  012572                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
2151  012442  000755                          BR      2$              ;;GET NEXT CHARACTER
2152  012444  004737  012526          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
2153  012450  123726  012604          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
2154  012454  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
2155  012456  013746  012602                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
2156                                                                  ;;AND THE NULL CHAR.
2157  012462  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
2158  012466  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
2159  012470  004737  012526                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
2160  012474  105337  012572                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2161  012500  000770                          BR      7$              ;;LOOP
2162
2163                                    ;HORIZONTAL TAB PROCESSOR
2164
2165  012502  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2166  012506  004737  012526          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2167  012512  132737  000007  012572          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2168  012520  001372                          BNE     9$              ;;TAB STOP
2169  012522  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2170  012524  000724                          BR      2$              ;;GET NEXT CHARACTER
2171  012526  105777  000044          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2172  012532  100375                          BPL     $TYPEC
2173  012534  116677  000002  000036          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2174  012542  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2175  012550  001003                          BNE     1$              ;;BRANCH IF NO
2176  012552  105037  012572                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2177  012556  000406                          BR      $TYPEX          ;;EXIT
2178  012560  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2179  012566  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2180  012570  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2181  012572  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
2182  012574  000207          $TYPEX: RTS     PC
2183
2184  012576  177564          $TPS:   .WORD   177564          ;;TTY PRINTER STATUS REG. ADDRESS
```



```
2185  012600  177566          $TPB:   .WORD   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
2186  012602      000         $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
2187  012603      002         $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
2188  012604      012         $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
2189  012605      000         $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
2190  012606      077         $QUES:  .ASCII  "?"             ;;QUESTION MARK
2191  012607      015         $CRLF:  .ASCII  <15>            ;;CARRAIGE RETURN
2192  012610  000012          $LF:    .ASCIZ  <12>            ;;LINEFEED
2193                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
2194
2195                                  ;;*************************************************************
2196                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2197                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
2198                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2199                                  ;*CALL:
2200                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2201                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
2202                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2203                                  ;*      .BYTE   M               ;;M=1 OR 0
2204                                  ;*                                      ;;1=TYPE LEADING ZEROS
2205                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
2206                                  ;*
2207                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2208                                  ;*$TYPOS OR $TYPOC
2209                                  ;*CALL:
2210                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2211                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
2212                                  ;*
2213                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2214                                  ;*CALL:
2215                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2216                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
2217
2218  012612  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
2219  012616  116637  000001  013035          MOVB    1(SP),$0FILL    ;;LOAD ZERO FILL SWITCH
2220  012624  112637  013037                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
2221  012630  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
2222  012634  000406                          BR      $TYPON
2223  012636  112737  000001  013035  $TYPOC: MOVB    #1,$0FILL       ;;SET THE ZERO FILL SWITCH
2224  012644  112737  000006  013037          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
2225  012652  112737  000005  013034  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
2226  012660  010346                          MOV     R3,-(SP)        ;;SAVE R3
2227  012662  010446                          MOV     R4,-(SP)        ;;SAVE R4
2228  012664  010546                          MOV     R5,-(SP)        ;;SAVE R5
2229  012666  113704  013037                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
2230  012672  005404                          NEG     R4
2231  012674  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
2232  012700  110437  013036                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
2233  012704  113704  013035                  MOVB    $0FILL,R4       ;;GET THE ZERO FILL SWITCH
2234  012710  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
2235  012714  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
2236  012716  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
2237  012720  000404                          BR      3$              ;;GO DO MSB
2238  012722  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
2239  012724  006105                          ROL     R5
2240  012726  006105                          ROL     R5
```

```
2241  012730  010503                        MOV     R5,R3
2242  012732  006103                3$:     ROL     R3                      ;;GET LSB OF THIS DIGIT
2243  012734  105337  013036                DECB    $OMODE                  ;;TYPE THIS DIGIT?
2244  012740  100016                        BPL     7$                      ;;BR IF NO
2245  012742  042703  177770                BIC     #177770,R3              ;;GET RID OF JUNK
2246  012746  001002                        BNE     4$                      ;;TEST FOR 0
2247  012750  005704                        TST     R4                      ;;SUPPRESS THIS 0?
2248  012752  001403                        BEQ     5$                      ;;BR IF YES
2249  012754  005204                4$:     INC     R4                      ;;DON'T SUPPRESS ANYMORE 0'S
2250  012756  052703  000060                BIS     #'0,R3                  ;;MAKE THIS DIGIT ASCII
2251  012762  052703  000040        5$:     BIS     #' ,R3                  ;;MAKE ASCII IF NOT ALREADY
2252  012766  110337  013032                MOVB    R3,@$                   ;;SAVE FOR TYPING
2253  012772  104400  013032                TYPE    ,@$                     ;;GO TYPE THIS DIGIT
2254  012776  105337  013034        7$:     DECB    $OCNT                   ;;COUNT BY 1
2255  013002  003347                        BGT     2$                      ;;BR IF MORE TO DO
2256  013004  002402                        BLT     6$                      ;;BR IF DONE
2257  013006  005204                        INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
2258  013010  000744                        BR      2$                      ;;GO DO THE LAST DIGIT
2259  013012  012605                6$:     MOV     (SP)+,R5                ;;RESTORE R5
2260  013014  012604                        MOV     (SP)+,R4                ;;RESTORE R4
2261  013016  012603                        MOV     (SP)+,R3                ;;RESTORE R3
2262  013020  016666  000002  000004        MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
2263  013026  012616                        MOV     (SP)+,(SP)
2264  013030  000002                        RTI                             ;;RETURN
2265  013032     000                8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
2266  013033     000                        .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
2267  013034     000                $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
2268  013035     000                $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
2269  013036  000000                $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
2270                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
2271
2272                                ;;**************************************************************
2273                                ;*SAVE R0-R5
2274                                ;*CALL:
2275                                ;*      SAVREG
2276                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
2277                                ;*
2278                                ;*TOP---(+16)
2279                                ;* +2---(+16)
2280                                ;* +4---R5
2281                                ;* +6---R4
2282                                ;* +8---R3
2283                                ;*+10---R2
2284                                ;*+12---R1
2285                                ;*+14---R0
2286
2287  013040                        $SAVREG:
2288  013040  010046                        MOV     R0,-(SP)                ;;PUSH R0 ON STACK
2289  013042  010146                        MOV     R1,-(SP)                ;;PUSH R1 ON STACK
2290  013044  010246                        MOV     R2,-(SP)                ;;PUSH R2 ON STACK
2291  013046  010346                        MOV     R3,-(SP)                ;;PUSH R3 ON STACK
2292  013050  010446                        MOV     R4,-(SP)                ;;PUSH R4 ON STACK
2293  013052  010546                        MOV     R5,-(SP)                ;;PUSH R5 ON STACK
2294  013054  016646  000022                MOV     22(SP),-(SP)            ;;SAVE PS OF MAIN FLOW
2295  013060  016646  000022                MOV     22(SP),-(SP)            ;;SAVE PC OF MAIN FLOW
2296  013064  016646  000022                MOV     22(SP),-(SP)            ;;SAVE PS OF CALL
```



```
2297  013070  016646  000022                MOV     22(SP),-(SP)            ;;SAVE PC OF CALL
2298  013074  000002                        RTI
2299
2300                                ;*RESTORE R0-R5
2301                                ;*CALL:
2302                                ;*      RESREG
2303  013076                        $RESREG:
2304  013076  012666  000022                MOV     (SP)+,22(SP)            ;;RESTORE PC OF CALL
2305  013102  012666  000022                MOV     (SP)+,22(SP)            ;;RESTORE PS OF CALL
2306  013106  012666  000022                MOV     (SP)+,22(SP)            ;;RESTORE PC OF MAIN FLOW
2307  013112  012666  000022                MOV     (SP)+,22(SP)            ;;RESTORE PS OF MAIN FLOW
2308  013116  012605                        MOV     (SP)+,R5                ;;POP STACK INTO R5
2309  013120  012604                        MOV     (SP)+,R4                ;;POP STACK INTO R4
2310  013122  012603                        MOV     (SP)+,R3                ;;POP STACK INTO R3
2311  013124  012602                        MOV     (SP)+,R2                ;;POP STACK INTO R2
2312  013126  012601                        MOV     (SP)+,R1                ;;POP STACK INTO R1
2313  013130  012600                        MOV     (SP)+,R0                ;;POP STACK INTO R0
2314  013132  000002                        RTI
2315                                .SBTTL  TTY INPUT ROUTINE
2316
2317                                ;;**************************************************************
2318  013134  177560                $TKS:   .WORD   177560                  ;;TTY KBD STATUS
2319  013136  177562                $TKB:   .WORD   177562                  ;;TTY KBD BUFFER
2320                                .ENABL  LSB
2321
2322                                ;;**************************************************************
2323                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
2324                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
2325                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
2326                                ;*WHEN OPERATING IN TTY FLAG MODE.
2327  013140  022737  000176  001214 $CKSWR: CMP     #$WREG,$WR              ;;IS THE SOFT-SWR SELECTED?
2328  013146  001074                        BNE     15$                     ;;BRANCH IF NO
2329  013150  105777  177760                TSTB    @$TKS                   ;;CHAR THERE?
2330  013154  100071                        BPL     15$                     ;;IF NO, DON'T WAIT AROUND
2331  013156  117746  177754                MOVB    @$TKB,-(SP)             ;;SAVE THE CHAR
2332  013162  042716  177600                BIC     #^C177,(SP)             ;;STRIP-OFF THE ASCII
2333  013166  022726  000007                CMP     #7,(SP)+                ;;IS IT A CONTROL G?
2334  013172  001062                        BNE     15$                     ;;NO, RETURN TO USER
2335  013174  123727  013714  000001        CMPB    $AUTOB,#1               ;;ARE WE RUNNING IN AUTO-MODE?
2336  013202  001456                        BEQ     15$                     ;;BRANCH IF YES
2337
2338  013204  104400  013665                TYPE    ,$CNTLG                 ;;ECHO THE CONTROL-G (^G)
2339  013210  104400  013672        $GTSWR: TYPE    ,$MSWR                  ;;TYPE CURRENT CONTENTS
2340  013214  013746  000176                MOV     $WREG,-(SP)             ;;SAVE SWREG FOR TYPEOUT
2341  013220  104401                        TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2342  013222  104400  013703                TYPE    ,$MNEW                  ;;PROMPT FOR NEW SWR
2343  013226  005046                19$:    CLR     -(SP)                   ;;CLEAR COUNTER
2344  013230  005046                        CLR     -(SP)                   ;;THE NEW SWR
2345  013232  105777  177676        7$:     TSTB    @$TKS                   ;;CHAR THERE?
2346  013236  100375                        BPL     7$                      ;;IF NOT TRY AGAIN
2347
2348  013240  117746  177672                MOVB    @$TKB,-(SP)             ;;PICK UP CHAR
2349  013244  042716  177600                BIC     #^C177,(SP)             ;;MAKE IT 7-BIT ASCII
2350
2351
2352
```

```
 2353  013250  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
 2354  013254  001005                          BNE     10$             ;;BRANCH IF NOT
 2355  013256  104400  013660                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
 2356  013262  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 2357  013266  000757                          BR      19$             ;;LET'S TRY IT AGAIN
 2358
 2359
 2360  013270  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
 2361  013274  001022                          BNE     16$             ;;BRANCH IF NO
 2362  013276  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 2363  013302  001403                          BEQ     11$             ;;BRANCH IF YES
 2364  013304  016677  000002  165702          MOV     2(SP),@SWR      ;;SAVE NEW SWR
 2365  013312  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
 2366  013316  104400  012607          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 2367  013322  123727  013715  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 2368  013330  001003                          BNE     15$             ;;BRANCH IF NOT
 2369  013332  012777  000100  177574          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 2370  013340  000002          15$:    RTI                             ;;RETURN
 2371  013342  004737  012526          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 2372  013346  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
 2373  013352  002420                          BLT     18$             ;;BRANCH IF YES
 2374  013354  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
 2375  013360  003015                          BGT     18$             ;;BRANCH IF YES
 2376  013362  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 2377  013366  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
 2378  013372  001403                          BEQ     17$             ;;BRANCH IF YES
 2379  013374  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
 2380  013376  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
 2381  013400  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
 2382  013402  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 2383  013406  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 2384  013412  000707                          BR      7$              ;;GET THE NEXT ONE
 2385  013414  104400  012606          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 2386  013420  000720                          BR      20$             ;;SIMULATE CONTROL-U
 2387                                  .DSABL  LSB
 2388
 2389
 2390                                  ;;*************************************************************
 2391                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 2392                                  ;*CALL:
 2393                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 2394                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 2395                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 2396                                  ;
 2397
 2398  013422  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 2399  013424  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
 2400  013432  105777  177476          1$:    TSTB    @$TKS           ;;WAIT FOR
 2401  013436  100375                          BPL     1$              ;;A CHARACTER
 2402  013440  117766  177472  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
 2403  013446  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 2404  013454  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 2405  013462  001013                          BNE     3$              ;;BRANCH IF NO
 2406  013464  105777  177444          2$:    TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 2407  013470  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
 2408  013472  117746  177440                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
```



```
 2409  013476  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 2410  013502  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 2411  013506  001366                          BNE     2$              ;;IF NOT DISCARD IT
 2412  013510  000750                          BR      1$              ;;YES, RESUME
 2413  013512  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
 2414  013520  002407                          BLT     4$              ;;BRANCH IF YES
 2415  013522  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 2416  013530  003003                          BGT     4$              ;;BRANCH IF YES
 2417  013532  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 2418  013540  000002          4$:     RTI                             ;;GO BACK TO USER
 2419                                  ;;*************************************************************
 2420                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 2421                                  ;*CALL:
 2422                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 2423                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 2424                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 2425
 2426  013542  010346          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 2427  013544  012703  013650          1$:    MOV     #$TTYIN,R3      ;;GET ADDRESS
 2428  013550  022703  013660          2$:    CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
 2429  013554  101405                          BLOS    4$              ;;BR IF YES
 2430  013556  104406                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 2431  013560  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 2432  013562  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 2433  013566  001003                          BNE     3$              ;;SKIP IF NOT
 2434  013570  104400  012606          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 2435  013574  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 2436  013576  111337  013646          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 2437  013602  104400  013646                  TYPE    ,9$
 2438  013606  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 2439  013612  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 2440  013614  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 2441  013620  104400  012610                  TYPE    ,$LF            ;;TYPE A LINE FEED
 2442  013624  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 2443  013626  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 2444  013630  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 2445  013636  012766  013650  000004          MOV     #$TTYIN,4(SP)
 2446  013644  000002                          RTI                     ;;RETURN
 2447  013646     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
 2448  013647     000                  .BYTE   0                       ;;TERMINATOR
 2449  013650  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
 2450  013660  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
 2451  013665     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
 2452  013672  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 2453  013700  020075     000
 2454  013703     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 2455  013710  036440  000040
 2456  013714     000          $AUTOB: .BYTE   0       ;;AUTO MODE FLAG
 2457  013715     000          $INTAG: .BYTE   0       ;;INTERRUPT MODE FLAG
 2458                                  .SBTTL  TRAP DECODER
 2459
 2460                                  ;;*************************************************************
 2461                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 2462                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 2463                                  ;*OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
 2464                                  ;*GO TO THAT ROUTINE.
```

```
                                ;
                        $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
                                MOV     2(SP),R0        ;;GET TRAP ADDRESS
                                TST     -(R0)           ;;BACKUP BY 2
                                MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
                                ASL     R0              ;;POSITION FOR INDEXING
                                MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
                                RTS     R0              ;;GO TO ROUTINE
                                ;
                        .SBTTL  TRAP TABLE
                                ;
                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                        ;*BY THE "TRAP" INSTRUCTION.
                                ;
                        ;       ROUTINE
                        ;       -------
                        $TRPAD:
                                $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
                                $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
                                $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
                                $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
                                ;
                                $GTSWR  ;;CALL=GTSWR    TRAP+4(104404)  GET SOFT-SWR SETTING
                                ;
                                $CKSWR  ;;CALL=CKSWR    TRAP+5(104405)  TEST FOR CHANGE IN SOFT-SWR
                                $RDCHR  ;;CALL=RDCHR    TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
                                $RDLIN  ;;CALL=RDLIN    TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
                                $SAVREG ;;CALL=SAVREG   TRAP+10(104410) SAVE R0-R5 ROUTINE
                                $RESREG ;;CALL=RESREG   TRAP+11(104411) RESTORE R0-R5 ROUTINE
                        .SBTTL  POWER DOWN AND UP ROUTINES
                                ;
                        ;;*************************************************************
                        ;POWER DOWN ROUTINE
                        $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
                                MOV     #340,@#PWRVEC+2 ;;PRIO:7
                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                MOV     R2,-(SP)        ;;PUSH R2 ON STACK
                                MOV     R3,-(SP)        ;;PUSH R3 ON STACK
                                MOV     R4,-(SP)        ;;PUSH R4 ON STACK
                                MOV     R5,-(SP)        ;;PUSH R5 ON STACK
                                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
                                MOV     SP,$SAVR6       ;;SAVE SP
                                MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
                                HALT
                                BR      .-2             ;;HANG UP
                                ;
                        ;;*************************************************************
                        ;POWER UP ROUTINE
                        $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
                                MOV     $SAVR6,SP       ;;GET SP
                                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
                        1$:     INC     $SAVR6          ;;WAIT FOR THE INC
                                BNE     1$              ;;OF  WORD
                                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
                                MOV     (SP)+,R5        ;;POP STACK INTO R5
```



```
                                MOV     (SP)+,R4        ;;POP STACK INTO R4
                                MOV     (SP)+,R3        ;;POP STACK INTO R3
                                MOV     (SP)+,R2        ;;POP STACK INTO R2
                                MOV     (SP)+,R1        ;;POP STACK INTO R1
                                MOV     (SP)+,R0        ;;POP STACK INTO R0
                                MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
                                MOV     #340,@#PWRVEC+2 ;;PRIO:7
                                TYPE                    ;REPORT THE POWER FAILURE
                        $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
                                MOV     (PC)+,(SP)      ;;RESTART AT RESTART
                        $PWRAD: .WORD   RESTART         ;;RESTART ADDRESS
                                RTI
                        $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
                                BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
                        $SAVR6: 0                       ;;PUT THE SP HERE
                        $POWER: .ASCIZ  <15><12>"POWER"
                                ;
                                .EVEN
                        .SBTTL  SINGLE LENGTH BINARY TO DECIMAL ASCIZ ROUTINE
                                ;
                        ;;*************************************************************
                        ;*THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
                        ;*UNSIGNED DECIMAL ASCIZ NUMBER.
                        ;*CALL
                        ;*      MOV     NUMBER,-(SP)    ;;PUT BINARY NUMBER ON THE STACK
                        ;*      JSR     PC,@#$B2D       ;;CALL
                        ;*      RETURN                  ;;ADDRESS OF THE 1ST ASCIZ CHAR,IS ON THE STACK
                                ;
                                ;
                        $B2D:   MOV     2(SP),1$        ;;SAVE BINARY NUMBER
                                MOV     #1$,-(SP)       ;;SET POINTER
                                JSR     PC,@#$D2D       ;;CALL DOUBLE LENGTH CONVERT
                                ADD     #5,(SP)         ;;ONLY ALLOW FIVE CHARACTERS
                                MOV     (SP)+,2(SP)     ;;PICKUP POINTER
                                RTS     PC              ;;RETURN
                        1$:     .WORD   0,0
                        .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
                                ;
                        ;;*************************************************************
                        ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
                        ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
                        ;*POSITIVE.
                        ;*CALL
                        ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                        ;*      JSR     PC,@#$D2D
                        ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
                                                        ;;IS ON THE STACK
                                ;
                        $D2D:   SAVREG                  ;;SAVE REGISTERS
                                MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
                                MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
                                MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
                                MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
                                MOV     (R2)+,R2
                                MOV     #10.,,45        ;;SET UP TO DO 10 CONVERSIONS
```

```
2577  014232  012704  014312                    MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
2578  014236  012705  014314                    MOV     #$TNPWR+2,R5
2579  014242  005003                  1$:       CLR     R3              ;;CLEAR PARTIAL
2580  014244  161401                  2$:       SUB     (R4),R1         ;;SUBTRACT TEN POWER
2581  014246  005602                            SBC     R2
2582  014250  161502                            SUB     (R5),R2
2583  014252  002402                            BLT     3$              ;;BR IF TEN POWER TO LARGE
2584  014254  005203                            INC     R3              ;;ADD 1 TO PARTIAL
2585  014256  000772                            BR      2$              ;;LOOP
2586  014260  062401                  3$:       ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
2587  014262  005502                            ADC     R2
2588  014264  062402                            ADD     (R4)+,R2
2589  014266  022525                            CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
2590  014270  052703  000060                    BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
2591  014274  110320                            MOVB    R3,(R0)+        ;;SAVE IT
2592  014276  005327                            DEC     (PC)+           ;;DONE?
2593  014300  000000                  4$:       .WORD   0
2594  014302  001357                            BNE     1$              ;;BR IF NO
2595  014304  105020                            CLRB    (R0)+           ;;TERMINATOR
2596  014306  104411                            RESREG                  ;;RESTORE REGISTERS
2597  014310  000207                            RTS     PC              ;;RETURN
2598  014312  145000                  $TNPWR: 145000                    ;;1.0E09
2599  014314  035632                          35632
2600  014316  160400                          160400                    ;;1.0E08
2601  014320  002765                          2765
2602  014322  113200                          113200                    ;;1.0E07
2603  014324  000230                          230
2604  014326  041100                          041100                    ;;1.0E06
2605  014330  000017                          17
2606  014332  103240                          103240                    ;;1.0E05
2607  014334  000001                          1
2608  014336  023420                          23420                     ;;1.0E04
2609  014340  000000                          0
2610  014342  001750                          1750                      ;;1.0E03
2611  014344  000000                          0
2612  014346  000144                          144                       ;;1.0E02
2613  014350  000000                          0
2614  014352  000012                          12                        ;;1.0E01
2615  014354  000000                          0
2616  014356  000001                          1                         ;;1.0E00
2617  014360  000000                          0
2618  014362  000014                  $DECVL: .BLKB   12.               ;;RESERVE STORAGE FOR ASCIZ STRING
2619
2620                                          ;;************************************************************
2621
2622                                          ;TYPE NUMERICAL ASCIZ STRING,RIGHT JUSTIFIED
2623                                          ;REPLACING LEADING ZEROS WITH SPACES.
2624                                          ;
2625                                          ;FIRST ADDRESS OF ASCIZ STRING MUST BE ON TOP OF THE STACK
2626
2627  014376  010046                  RTJUST:         MOV R0,-(SP)            ;SAVE R0
2628  014400  016600  000004                         MOV 4(SP),R0            ;PICK UP ADDRESS OF ASCIZ STRING
2629  014404  010037  014436                         MOV R0,3$               ;SAVE ADDRESS FOR TYPE OUT
2630  014410  105710                  1$:             TSTB (R0)               ;IS THIS THE TERMINATOR
2631  014412  001406                                 BEQ 2$                  ;IF YES TYPE IT OUT
2632  014414  122710  000060                         CMPB #'0,(R0)           ;IS IT A ZERO
```



```
2633  014420  001005                                 BNE 4$                  ;IF NO GO PRINT IT
2634  014422  112720  000040                         MOVB #' ,(R0)+          ;IF YES REPLACE IT WITH A SPACE
2635  014426  000770                                 BR 1$                   ;TEST NEXT CHAR.
2636  014430  112740  000060          2$:            MOVB #'0,-(R0)          ;STRING OFF ALL ZEROS,PUT BACK THE LAST ONE
2637  014434  104400                  4$:            TYPE                    ;TYPE THE STRING
2638  014436  000000                  3$:            OPEN
2639  014440  012600                                 MOV (SP)+,R0            ;RESTORE R0
2640  014442  012616                                 MOV (SP)+,(SP)          ;RESTORE THE STACK
2641  014444  000207                                 RTS PC                  ;RETURN
2642
2643                                          ;TYPES 16 BIT WORD IN DECIMAL
2644
2645  014446  012546                  SGLDEC:        MOV (R5)+,-(SP)         ;PUT NUMBER TO BE TYPED ON STACK
2646  014450  004737  014146                         JSR PC,$$$B2D           ;CONVERT NUMBER TO DECIMAL
2647  014454  004737  014376                         JSR PC,RTJUST           ;TYPE THE DECIMAL NUMBER
2648  014460  000205                                 RTS R5
2649
2650
2651  014462  047125  054105  042520  MUNXDD:       .ASCIZ "UNEXPECTED D D MARK"
2652  014470  052103  042105  042040
2653  014476  042040  046440  051101
2654  014504  000113
2655
2656  014506  020104  020104  040515  MDDMIS:       .ASCIZ "D D MARK MISSING"
2657  014514  045522  046440  051511
2658  014522  044523  043516     000
2659
2660
2661  014527     104  052101  026101  MDERHDR:      .ASCIZ "DATA, NO STATUS ERROR"
2662  014534  047040  020117  052123
2663  014542  052101  051525  042440
2664  014550  051122  051117     000
2665
2666  014555     040  047117  052040  MTRK:         .ASCIZ " ON TRACK"
2667  014562  040522  045503     000
2668
2669
2670  014567     040  043040  047522  MPRES:        .ASCIZ "  FROM TRACK"
2671  014574  020115  051124  041501
2672  014602  000113
2673
2674  014604  027440  051440  041505  MSECT:        .ASCIZ " / SECTOR"
2675  014612  047524  000122
2676
2677  014616  005015  041040  052131  MCOLMUN:      .ASCIZ <15><12>" BYTE  BAD  GOOD"<15><12>
2678  014624  020105  041040  042101
2679  014632  020040  047507  042117
2680  014640  005015     000
2681
2682  014643     015  041412  052501  D0LOAD: .ASCII  <15><12>"CAUTION - IF YOU DESIRE TO TEST UNIT 0"
2683  014650  044524  047117  026440
2684  014656  044440  020106  047531
2685  014664  020125  042504  044523
2686  014672  042522  052040  020117
2687  014700  042524  052123  052440
2688  014706  044516  020124     060
```

```
2689  014713    015  051012  050105          ,ASCII  <15><12>"REPLACE LOAD MEDIUM WITH A SCRATCH DISKETTE"
2690  014720  040514  042503  046040
2691  014726  040517  020104  042515
2692  014734  044504  046525  053440
2693  014742  052111  020110  020101
2694  014750  041523  040522  041524
2695  014756  020110  044504  045523
2696  014764  052105  042524
2697  014770  005015  044124  047105          ,ASCIZ  <15><12>"THEN PRESS CONTINUE"
2698  014776  050040  042522  051523
2699  015004  041440  047117  044524
2700  015012  052516  000105
2701
2702  015016  047516  042040  044522  MNODRY:         ,ASCIZ "NO DRIVES READY"<15><12>
2703  015024  042526  020123  042522
2704  015032  042101  006531  000012
2705
2706  015040  020040  020040  020040  TAB:            ,ASCIZ <40><40><40><40><40><40><40>
2707  015046    000
2708
2709  015047    040  000040          DBLSP:          ,ASCIZ <40><40>
2710
2711  015052  005015    000           MCRLF:          ,ASCIZ <15><12>
2712
2713  015055    104    000           MEOP:           ,ASCIZ "D"
2714
2715  015057    055    000           MEREOP:         ,ASCIZ "-"
2716
2717  015061    007    000           MABELL:         ,ASCIZ <07>
2718
2719  015063    105  051122  051117  MERHEADER:      ,ASCIZ "ERROR CONDITIONS  "
2720  015070  041440  047117  044504
2721  015076  044524  047117  020123
2722  015104  000040
2723
2724  015106  047125  052111  030040  MUNIT0:         ,ASCIZ "UNIT 0 "
2725  015114  000040
2726
2727  015116  047125  052111  030440  MUNIT1:         ,ASCIZ "UNIT 1 "
2728  015124  000040
2729
2730  015126  040510  020123  042502  MDELET:         ,ASCIZ "HAS BEEN DELETED."
2731  015134  047105  042040  046105
2732  015142  052105  042105  000056
2733
2734  015150  044440  052116  053040  MINTVEC:        ,ASCIZ " INT VECTOR = "
2735  015156  041505  047524  020122
2736  015164  020075    000
2737
2738  015167    122  041530  020123  MRXCS:          ,ASCIZ "RXCS = "
2739  015174  020075    000
2740
2741  015177    040  054122  041104  MRXDB:          ,ASCIZ " RXDB = "
2742  015204  036440  000040
2743
2744  015210  052123  052101  051525  MASTAT:         ,ASCIZ "STATUS A = "
```



```
2745  015216  040440  036440  000040
2746
2747  015224  052123  052101  051525  MBSTAT:         ,ASCIZ "STATUS B = "
2748  015232  041040  036440  000040
2749
2750  015240  005015  000012          DBLLF:          ,ASCIZ <15><12><12>
2751
2752  015244  036520  000040          MPAT:           ,ASCIZ "P= "
2753
2754  015250  036524  000040          MTEST:          ,ASCIZ "T= "
2755
2756  015254  036523  000040          MSEQ:           ,ASCIZ "S= "
2757
2758  015260  047516  044440  052116  MINTER:         ,ASCIZ "NO INTERUPT AT DONE ERROR"
2759  015266  051105  050125  020124
2760  015274  052101  042040  047117
2761  015302  020105  051105  047522
2762  015310  000122
2763
2764  015312  047125  047113  053517  MUKNINT:        ,ASCIZ "UNKNOWN INTERUPT"
2765  015320  020116  047111  042524
2766  015326  052522  052120    000
2767
2768  015333    124  052117  046101  MERCT:          ,ASCIZ "TOTAL READ CHECK ERRORS = "
2769  015340  051040  040505  020104
2770  015346  044103  041505  020113
2771  015354  051105  047522  051522
2772  015362  036440  000040
2773
2774  015366  044506  046114  052502  MFIL:           ,ASCIZ "FILLBUFFER "
2775  015374  043106  051105  000040
2776
2777  015402  046505  052120  041131  MEMPTY:         ,ASCIZ "EMPTYBUFFER "
2778  015410  043125  042506  020122
2779  015416    000
2780
2781  015417    104  044522  042526  MICON:          ,ASCIZ "DRIVE(S)  "
2782  015424  051450  020051  000040
2783
2784  015432  042524  052123  044040  MHUNG:          ,ASCIZ "TEST HUNG"<15><12>
2785  015440  047125  006507  000012
2786
2787  015446  047516  051516  040524  MNONSTD:        ,ASCIZ "NONSTANDARD TRACK / SECTOR LIMITS"
2788  015454  042116  051101  020104
2789  015462  051124  041501  020113
2790  015470  020057  042523  052103
2791  015476  051117  046040  046511
2792  015504  052111  000123
2793
2794  015510  042117  000075          MOD:            ,ASCIZ "OD="
2795
2796  015514  042111  000075          MID:            ,ASCIZ "ID="
2797
2798  015520  044506  051522  036524  MFIRST:         ,ASCIZ "FIRST="
2799  015526    000
2800
```

```
2801  015527  000114  051501  036524  MLAST:          .ASCIZ "LAST="
2802  015534  000000
2803
2804  015535  000111  044516  027124  MINIT1:         .ASCIZ "INIT.DONE NOT SET ERROR"<15><12>
2805  015542  047504  042516  047040
2806  015550  052117  051440  052105
2807  015556  042440  051122  051117
2808  015564  005015  000009
2809
2810  015567  000111  044516  027124  MINIT2:         .ASCIZ "INIT.DONE SET ERROR" <15><12>
2811  015574  047504  042516  051440
2812  015602  052105  042440  051122
2813  015610  051117  005015  000000
2814
2815  015615  000104  042040  042440  MDDER:          .ASCIZ "D D ERROR"
2816  015622  051122  051117  000000
2817
2818  015627  000122  041505  053117  MREC:          .ASCIZ "RECOVERABLE "
2819  015634  051105  041101  042514
2820  015642  000040
2821
2822  015644  051103  020103  051105  MBADCRC:        .ASCIZ "CRC ERROR NO DATA ERROR"
2823  015652  047522  020122  047516
2824  015660  042040  052101  020101
2825  015666  051105  047522  000122
2826
2827  015674  042522  042101  000040  MREAD:         .ASCIZ "READ "
2828
2829  015702  040510  052114  032040  MHALT4:        .ASCIZ "HALT 4" <15><12>
2830  015710  005015  000000
2831
2832  015713  000110  046101  020124  MHALT3:        .ASCIZ "HALT 3" <15><12>
2833  015720  006463  000012
2834
2835  015724  040510  052114  030440  MHALT11:       .ASCIZ "HALT 11" <15><12>
2836  015732  006461  000012
2837
2838  015736  040504  040524  041440  MCRC:          .ASCIZ "DATA CRC ERROR"
2839  015744  041522  042440  051122
2840  015752  051117  000000
2841
2842
2843  015755  000040  047125  042522  MUNREC:        .ASCIZ " UNRECOVERABLE "
2844  015762  047503  042526  040522
2845  015770  046102  020105  000000
2846
2847  015775  000123  042505  020113  MSEEK:         .ASCIZ "SEEK ERROR"
2848  016002  051105  047522  000122
2849
2850  016010  051127  052111  020105  MWRITE:        .ASCIZ "WRITE "
2851  016016  000000
2852
2853  016017  000120  051101  052111  MPAR:          .ASCIZ "PARITY ERROR"
2854  016024  020131  051105  047522
2855  016032  000122
2856
```



```
2857  016034  051105  047522  020122  MNOFLAG:        .ASCIZ "ERROR FLAG ERROR"
2858  016042  046106  043501  042440
2859  016050  051122  051117  000000
2860
2861  016055  000122  030530  020061  MDUMP:         .ASCIZ "RX11 DUMP"
2862  016062  052504  050115  000000
2863
2864  016067  000105  051122  051117  MERS:          .ASCIZ "ERRORS"
2865  016074  000123
2866
2867  016076  044123  051117  020124  MSHORT:        .ASCIZ "SHORT DUMP"
2868  016104  052504  050115  000000
2869
2870  016111  000122  051505  040524  MRESTART:      .ASCIZ "RESTARTS = "
2871  016116  052122  020123  020075
2872  016124  000000
2873
2874  016125  000120  051501  020123  MNOPAS:        .ASCIZ "PASS ABORTED"
2875  016132  041101  051117  042524
2876  016140  000124
2877
2878  016142  040520  051523  051505  MPASS:         .ASCIZ "PASSES = "
2879  016150  036440  000040
2880
2881  016154  051127  052111  042524  MWRTEN:        .ASCIZ "WRITTEN "
2882  016162  020116  000000
2883
2884  016165  000057  000040          MSLASH:        .ASCIZ "/ "
2885
2886  016170  000040                  SPACE:         .ASCIZ <40>
2887
2888  016172  051105  047522  051522  MCODES:        .ASCIZ "ERRORS PER ERROR CODES"<15><12>
2889  016200  050040  051105  042440
2890  016206  051122  051117  041440
2891  016214  042117  051505  005015
2892  016222  000000
2893
2894  016223  000124  040522  045503  MTKLOG:        .ASCIZ "TRACK  ACCESSED   MOVED TO    UNIT 0  UNIT 1 ERRORS"<15><12>
2895  016230  020040  041501  042503
2896  016236  051523  042105  020040
2897  016244  046440  053117  042105
2898  016252  052040  020117  020040
2899  016260  052440  044516  020124
2900  016266  020060  052440  044516
2901  016274  020124  020061  051105
2902  016302  047522  051522  005015
2903  016310  000000
2904
2905  016311  000050  047516  042516  MNONE:         .ASCIZ "(NONE)"<15><12>
2906  016316  006451  000012
2907
2908  016322  047507  042117  000000  MGOOD:         .ASCIZ "GOOD"
2909
2910  016327  000040  041440  042510  MSUM:          .ASCIZ "  CHECK SUM "
2911  016334  045503  051440  046525
2912  016342  000040
```

```
                                                                
016344  005015  054122  030461  MRX11:          .ASCIZ <15><12>"RX11 / RXV11"
016352  027440  051040  053130
016360  030461     000
                                                                
016363     015  052012  040522  OD2BIG:         .ASCII <15><12> "TRACK LIMITS SELECTED OUT OF RANGE,"
016370  045503  046040  046511
016376  052111  020123  042523
016404  042514  052103  042105
016412  047440  052125  047440
016420  020106  040522  043516
016426  026105
016430  051525  047111  020107                  .ASCIZ "USING OD=0, ID=114"<15><12>
016436  042117  030075  020054
016444  042111  030475  032061
016452  005015     000
                                                                
016455     015  051412  041505  S2BIG:          .ASCII <15><12> "SECTOR LIMITS SELECTED OUT OF RANGE,"
016462  047524  020122  044514
016470  044515  051524  051440
016476  046105  041505  042524
016504  020104  052517  020124
016512  043117  051040  047101
016520  042527     054
016523     125  044523  043516                  .ASCIZ "USING FIRST=1, LAST=32"<15><12>
016530  043040  051111  052123
016536  030475  020054  040514
016544  052123  031475  006462
016552  000012
                                                                
016554  005015  040515  047111  MREV:          .ASCIZ <15><12> "MAINDEC-11-DZRXA-E" <15><12>
016562  042504  026503  030461
016570  042055  051132  040530
016576  042455  005015     000
                                                                
        016604                          .EVEN
                                                                
                                ;;**************************************************************
                                                                
                                ;THE FOLLOWING LOCATIONS ARE USED FOR DATA STORAGE,RETRY COUNTERS
                                ;ACCESS COUNTERS ETC.
                                                                
016604  000200                  BUFADR:         .BLKW 200
                                                                
                                ;RETRY COUNTERS
017004  000012                  RDRETRY:        10.             ;FOR SETUP PURPOSE RDRETRY MUST BE FIRST ON RETRY LIST
017006  000012                  WTRETRY:        10.
017010  000012                  DDRETRY:        10.
017012  000012                  DATARETRY:      10.
017014  000012                  P2RETRY:        10.
017016  000012                  PRETRY:         10.
017020  000012                  CRETRY:         10.
017022  000012                  SRETRY:         10.             ;FOR SETUP PURPOSE SRETRY MUST BE LAST ON RETRY LIST
                                                                
                                ;GENERAL ERROR COUNTERS
```



```
017024  000000                  PARLOG:         0               ;FOR SETUP PURPOSE PARLOG MUST BE FIRST ON COUNTERS LIST
017026  000000                  MPARLOG:        0
017030  000000                  NDERLOG:        0
017032  000000                  UKNINT:         0
017034  000000                  INTER:          0
                                ;DRIVE RELATED ERROR COUNTERS
017036  000000                  ZSEKLOG:        0               ;"Z" LOGS ARE FOR DRIVE UNIT "Z"ERO
017040  000000                  SEKLOG:         0               ;NON"Z" LOGS ARE FOR DRIVE UNIT 1
017042  000000                  ZCRCLOG:        0
017044  000000                  CRCLOG:         0               ; * * * NOTE: * * *
017046  000000                  ZCRCBAD:        0               ;ERROR CODES MUST NOT BE CHANGED
017050  000000                  CRCBAD:         0               ;FROM THIS ORDER, ERROR DUMP
017052  000000                  ZRDLOG:         0               ;ASSUMES TAGS OF LOGS ARE IN ORDER SHOWN.
017054  000000                  RDLOG:          0
017056  000000                  ZWRTLOG:        0
017060  000000                  WRTLOG:         0
017062  000000                  ZDATALOG:       0
017064  000000                  DATALOG:        0
017066  000000                  ZDDMIS:         0
017070  000000                  DDMIS:          0
017072  000000                  ZUNXDD:         0
017074  000000                  UNXDD:          0
017076  000000                  ZHSEKLOG:       0
017100  000000                  HSEKLOG:        0
017102  000000                  ZHCRCLOG:       0
017104  000000                  HCRCLOG:        0
017106  000000                  ZHCRCBAD:       0
017110  000000                  HCRCBAD:        0
017112  000000                  ZHRDLOG:        0
017114  000000                  HRDLOG:         0
017116  000000                  ZHWRTLOG:       0
017120  000000                  HWRTLOG:        0
017122  000000                  ZHDATALOG:      0
017124  000000                  HDATALOG:       0
017126  000000                  ZHDDLOG:        0
017130  000000                  HDDLOG:         0
                                                                
                                ;ERROR CODES
017132  000021                  ERCODE:         .BLKW 17.
                                                                
                                ;THE FOLLOWING 2 BLOCKS OF WORDS ARE TRACK ACCESS COUNTER AND
                                ;HEAD MOVED TO TRACK COUNTERS, THEY ARE DOUBLE PRECISION
                                ;COUNTERS IN THE FOLLOWING FORMATT:
                                ;
                                ;LOC.X          LOW ORDER WORD TRACK 0
                                ;LOC.X+2        HIGH ORDER WORD TRACK 0
                                ;LOC.X+4        LOW ORDER WORD TRACK 1
                                ;LOC.X+6        HIGH ORDER WORD TRACK 1
                                ;ETC.
                                                                
                                ;TOTAL ACCESS / TRACK
017174  000232                  TKACC:          .BLKW 232
                                                                
                                ;TOTAL HEAD MOVEMENT TO TRACK
017660  000232                  HDMOVE:         .BLKW 232
```

```
                                        ;THE FOLLOWING 2 BLOCKS OF COUNTERS ARE SINGLE PRECISION

                                        ;ERROR PER TRACK ON UNIT 0
020344  000115              U0TRK:    .BLKW 115

                                        ;ERROR PER TRACK ON UNIT 1
020576  000115              U1TRK:    .BLKW 115

                                        ;TOTAL WRITE AND READ FUNCTIONS DOUBLE PRECISION COUNTERS
021030  000000  000000      WTCNTR:   .WORD 0,0
021034  000000  000000      RDCNTR:   .WORD 0,0

                                        ;RESTART AND PASSES COMPLTED COUNTERS
021040  000000              RESTCNTR: 0
021042  000000              PASCNTR:  0             ;FOR SETUP PURPOSE PASCNTR MUST BE PLACED LAST

021044  032737  000020  010704  U1LOG:   BIT #BIT4,UNITSEL    ;TEST FOR UNIT IN OPERATION
021052  001402                        BEQ 1$              ;IF BIT 4 IS 0 RETURN AND INC. UNIT 0 COUNTERS
021054  062703  000002                ADD #2,R3           ;ADJUST THE CONTENTS OF R3 FOR
                                                            ;THE ADDRESS OF UNIT 1 ERROR LOG COUNTERS.
021060  000207              1$:       RTS PC

                                        ;STATISTICAL ERROR REPORT ROUTEEN
                                        ;PRINTS ALL ERROR LOGS AND OPERATING CONDITIONS

                                        ;;*************************************************************

021062  012706  001200      ERDUMP:   MOV #STACK,SP
021066  104400  015240                TYPE ,CRLLF
021072  005737  022514                TST SHORTRPT        ;IS SHORT REPORT REQUESTED
021076  001405                        BEQ 1$              ;NO,PRINT LONG REPORT
021100  104400  015040                TYPE ,TAB
021104  104400  016076                TYPE ,MSHORT        ;YES,TYPE SHORT REPOR HEADER
021110  000404                        BR 2$
021112  104400  015040      1$:       TYPE ,TAB
021116  104400  016055                TYPE ,MDUMP         ;TYPE LONG REPORT HEADER
021122  104400  016554      2$:       TYPE ,MREV          ;PRINT NAME AND REVISION OF PROGRAM
021126  004737  002452                JSR PC,ICOND        ;TYPE OUT SELECTED DRIVES AND TESTS
021132  005737  021040                TST RESTCNTR        ;HAVE THERE BEEN ANY RESTARTS
021136  001412                        BEQ 5$              ;NO
021140  104400  016111                TYPE ,MRESTART      ;YES,PRINT OUT HOW MANY
021144  013737  021040  021156        MOV RESTCNTR,12$
021152  004537  014446                JSR R5,SGLDEC
021156  000000              12$:      OPEN
021160  104400  015240                TYPE ,CRLLF
021164  005737  021042      3$:       TST PASCNTR         ;HAVE ANY PASSES BEEN COMPLETED
021170  001005                        BNE 4$              ;YES,PRINT OUT HOW MANY
021172  104400  016125                TYPE ,MNOPAS        ;NO,TYPE PASS ABORTED
021176  104400  015240                TYPE ,CRLLF
021202  000412                        BR 5$
021204  104400  016142      4$:       TYPE ,MPASS         ;TYPE OUT NUMBER OF PASSES
021210  013737  021042  021222        MOV PASCNTR,7$
021216  004537  014446                JSR R5,SGLDEC
021222  000000              7$:       OPEN
021224  104400  015052                TYPE ,MCRLF
021230  005737  022514      5$:       TST SHORTRPT        ;IS THIS A SHORT REPORT
```



```
021234  001030                        BNE AERRS           ;YES,DON'T PRINT WRITE/READ TOTALS

                                    ;PRINTS TOTAL NUMBER OF SECTORS WRITTEN AND/OR
                                    ;READ FOR THE DURATION OF RUN TIME.
021236  104400  016154                TYPE ,MWRTEN        ;PRINT WRT/RD MESSAGE
021242  104400  016165                TYPE ,MSLASH
021246  104400  015674                TYPE ,MREAD
021252  104400  015047                TYPE,DBLSP
021256  012746  021030                MOV #WTCNTR,-(SP)    ;ADDRESS OF TOTAL WRITES ON STACK
021262  004737  014202                JSR PC,@#$DB2D       ;TYPE TOTAL WRITES
021266  004737  014376                JSR PC,RTJUST
021272  104400  016165                TYPE ,MSLASH
021276  012746  021034                MOV #RDCNTR,-(SP)    ;ADDRESS OF READS ON STACK
021302  004737  014202                JSR PC,@#$DB2D       ;TYPE TOTAL READS
021306  004737  014376                JSR PC,RTJUST
021312  104400  015240                TYPE ,CRLLF

                                    ;TYPE OUT MICRO CPU RELATED ERRORS
021316  104400  015052      AERRS:    TYPE ,MCRLF
021322  005737  017024                TST PARLOG          ;WERE THERE ANY PARITY ERRORS
021326  001435                        BEQ 1$
021330  013737  017024  021342        MOV PARLOG,20$      ;YES,TYPE OUT THE NUMBER
021336  004537  014446                JSR R5,SGLDEC
021342  000000              20$:      OPEN
021344  104400  015047                TYPE ,DBLSP
021350  104400  016017                TYPE ,MPAR
021354  104400  015052                TYPE,MCRLF
021360  005737  017026                TST MPARLOG         ;WERE THERE ANY HARD PARITY ERRORS
021364  001416                        BEQ 1$
021366  013737  017026  021400        MOV MPARLOG,21$     ;YES,TYPE OUT THE COUNT
021374  004537  014446                JSR R5,SGLDEC
021400  000000              21$:      OPEN
021402  104400  015047                TYPE ,DBLSP
021406  104400  015755                TYPE ,MUNREC
021412  104400  016017                TYPE ,MPAR
021416  104400  015052                TYPE ,MCRLF
021422  005737  017030      1$:       TST NDERLOG         ;ANY STATUS ERROR FLAG ERRORS
021426  001414                        BEQ 3$              ;IF NONE CHECK NEXT ERROR
021430  013737  017030  021442        MOV NDERLOG,24$     ;YES,TYPE OUT COUNT
021436  004537  014446                JSR R5,SGLDEC
021442  000000              24$:      OPEN
021444  104400  015047                TYPE ,DBLSP
021450  104400  016034                TYPE ,MNOFLAG
021454  104400  015052                TYPE ,MCRLF
021460  005737  017034      3$:       TST INTER           ;ANY NO INTERUPT ON DONE ERRORS
021464  001414                        BEQ 4$              ;IF NONE CHECK NEXT ERROR GROUP
021466  013737  017034  021500        MOV INTER,25$       ;YES,PRINT OUT NUMBER
021474  004537  014446                JSR R5,SGLDEC
021500  000000              25$:      OPEN
021502  104400  015047                TYPE ,DBLSP
021506  104400  015260                TYPE ,MINTER
021512  104400  015052                TYPE ,MCRLF
                                    ;TYPES DRIVE RELATED ERRORS
021516  104400  015240      4$:       TYPE ,CRLLF
021522  104400  015106                TYPE ,MUNIT0        ;TYPE OUT COLUMN HEADINGS
021526  104400  015116                TYPE ,MUNIT1
```

```
3157  021532  104400  016170                  TYPE ,SPACE
3158  021536  104400  016067                  TYPE ,MERS
3159  021542  104400  015052                  TYPE ,MCRLF
3160  021546  005002                          CLR R2                  ;R2 USED AS AN ERROR PRINTED FLAG
3161  021550  013746  022516                  MOV ZERO,-(SP)          ;PUT ADDR OF ZERO MARK ON STACK
3162  021554  012746  014462                  MOV #MUNXDU,-(SP)       ;PUT ADDRESSES OF ALL RECOVERABLE
3163  021560  012746  014506                  MOV #MDDMIS,-(SP)       ;ERROR MESSAGES ON THE STACK
3164  021564  012746  014527                  MOV #MDERHDR,-(SP)
3165  021570  012746  016010                  MOV #MWRITE,-(SP)
3166  021574  012746  015674                  MOV #MREAD,-(SP)
3167  021600  012746  015644                  MOV #MBADCHC,-(SP)
3168  021604  012746  015736                  MOV #MCRC,-(SP)
3169  021610  012746  015775                  MOV #MSEEK,-(SP)
3170  021614  012701  017036                  MOV #ZSEKLOG,R1         ;PUT FIRST ERROR COUNTER IN REGISTER
3171  021620  005716          7$:             TST (SP)                ;GET ADDRESS OF MESSAGE
3172  021622  001436                          BEQ 5$                  ;IF 0 FINISHED SOFT ERRS DO HARD ERRS
3173  021624  005721                          TST (R1)+               ;IS DRIVE 0 COUNTER CLEAR
3174  021626  001005                          BNE 6$                  ;NO,GO PRINT ERROR COUNTER
3175  021630  005711                          TST (R1)                ;IS DRIVE 1 COUNTER CLEAR
3176  021632  001003                          BNE 6$                  ;NO,GO PRINT ERROR COUNTER
3177  021634  005721                          TST (R1)+               ;ADJUST R1 FOR NEXT UNIT 0 COUNTER
3178  021636  005726                          TST (SP)+               ;ADJUST STACK FOR NEXT ADDRESS
3179  021640  000767                          BR 7$                   ;GO TEST NEXT PAIR OF COUNTERS
3180  021642  005202          6$:             INC R2                  ;MAKE ERROR PRINTED FLAG NON ZERO
3181  021644  014137  021654                  MOV -(R1),30$           ;TYPE CONTENTS OF UNIT 0 COUNTER
3182  021650  004537  014446                  JSR R5,SGLDEC
3183  021654  000000          30$:            OPEN
3184  021656  104400  015047                  TYPE ,DBLSP
3185  021662  005721                          TST (R1)+               ;ADJUST R1 FOR UNIT 1 COUNTER
3186  021664  012137  021674                  MOV (R1)+,31$           ;TYPE COUNTER,GET ADDR OF NEXT IN R1
3187  021670  004537  014446                  JSR R5,SGLDEC
3188  021674  000000          31$:            OPEN
3189  021676  104400  015047                  TYPE ,DBLSP
3190  021702  012637  021710                  MOV (SP)+,13$
3191  021706  104400                          TYPE                    ;TYPE ERROR MESSAGE FROM STACK
3192  021710  000000          13$:            OPEN
3193  021712  104400  015052                  TYPE ,MCRLF
3194  021716  000740                          BR 7$                   ;GET NEXT COUNTER
3195  021720  104400  015052          5$:     TYPE ,MCRLF
3196  021724  013746  022516                  MOV ZERO,-(SP)          ;PUT ADDR OF ZERO MARK ON THE STACK
3197  021730  012746  015615                  MOV #MDDERH,-(SP)       ;PUT ADDRESS OF ALL HARD ERROR
3198  021734  012746  014527                  MOV #MDERHDR,-(SP)      ;MESSAGES ON THE STACK
3199  021740  012746  016010                  MOV #MWRITE,-(SP)
3200  021744  012746  015674                  MOV #MREAD,-(SP)
3201  021750  012746  015644                  MOV #MBADCHC,-(SP)
3202  021754  012746  015736                  MOV #MCRC,-(SP)
3203  021760  012746  015775                  MOV #MSEEK,-(SP)
3204  021764  012701  017076                  MOV #ZHSEKLOG,R1        ;PUT FIRST HARD ERROR COUNTER IN R1
3205  021770  005716          12$:            TST (SP)                ;GET ADDRESS OF MESSAGE FROM STACK
3206  021772  001437                          BEQ BERRS               ;IF 0 FINISHED ERROR LOGS
3207  021774  005721                          TST (R1)+               ;IS DRIVE 0 COUNTER CLEAR
3208  021776  001005                          BNE 11$                 ;NO,PRINT ERROR COUNTER AND MESSAGE
3209  022000  005711                          TST (R1)                ;IS DRIVE 1 COUNTER CLEAR
3210  022002  001003                          BNE 11$                 ;NO,GO PRINT THE ERROR
3211  022004  005721                          TST (R1)+               ;ADJUST R1 FOR NEXT UNIT 0 LOG
3212  022006  005726                          TST (SP)+               ;ADJUST STACK FOR NEXT ADDRESS
```



```
3213  022010  000767                          BR 12$
3214  022012  014137  022022  11$:            MOV -(R1),32$           ;TYPE CONTENTS OF UNIT 0 COUNTER
3215  022016  004537  014446                  JSR R5,SGLDEC
3216  022022  000000          32$:            OPEN
3217  022024  104400  015047                  TYPE ,DBLSP
3218  022030  005721                          TST (R1)+               ;ADJUST R1 FOR UNIT 1 COUNTER
3219  022032  012137  022042                  MOV (R1)+,33$           ;TYPE UNIT 1 LOG
3220  022036  004537  014446                  JSR R5,SGLDEC
3221  022042  000000          33$:            OPEN
3222  022044  104400  015047                  TYPE ,DBLSP
3223  022050  104400  015755                  TYPE ,MUNREC            ;TYPE UNRECOVERABLE
3224  022054  012637  022062                  MOV (SP)+,14$
3225  022060  104400                          TYPE                    ;ERROR MESSAGE FROM STACK
3226  022062  000000          14$:            OPEN
3227  022064  104400  015052                  TYPE ,MCRLF
3228  022070  000737                          BR 12$                  ;GET NEXT LOG
3229                        ;PRINTS ERRORS PER ERROR CODES
3230  022072  005702          BERRS:          TST R2                  ;WAS AN ERROR PRINTED IN THE LAST GROUP
3231  022074  001004                          BNE 5$                  ;IF R2 IS NONZERO BRANCH
3232  022076  104400  015040                  TYPE ,TAB
3233  022102  104400  016311                  TYPE ,MNONE             ;TYPE NONE
3234  022106  005002          5$:             CLR R2                  ;CLEAR ERROR PRINTED FLAG AGAIN
3235  022110  104400  015240                  TYPE ,DBLLF
3236  022114  104400  016172                  TYPE ,MCODES            ;TYPE ERROR PER ERROR CODE
3237  022120  012700  000001                  MOV #1,R0               ;SET UP CODE COUNTER
3238  022124  012701  017132                  MOV #ERCODE,R1          ;GET ADDR OF FIRST ERROR CODE
3239  022130  020027  000022          2$:     CMP R0,#22              ;IS THE LAST CODE PRINTED
3240  022134  001001                          BNE 3$                  ;NO,TEST THE NEXT ONE
3241  022136  000427                          BR TYPTRK               ;YES,GO PRINT TRACK INFOMATION
3242  022140  005721          3$:             TST (R1)+               ;TEST FOR 0 WORD,GET NEXT ADDRESS
3243  022142  001002                          BNE 1$                  ;IF NOT 0 TYPE COUNT AND CODE NUMBER
3244  022144  005200                          INC R0                  ;INC CODE COUNTER
3245  022146  000770                          BR 2$
3246  022150  014137  022160          1$:     MOV -(R1),4$            ;PUT CONTENTS OF COUNTER IN OPEN
3247  022154  004537  014446                  JSR R5,SGLDEC           ;FOR TYPE OUT
3248  022160  000000          4$:             OPEN
3249  022162  104400  015040                  TYPE ,TAB
3250
3251  022166  010002                          MOV R0,R2               ;PUT CODE NUMBER IN REGISTER TO
3252  022170  006302                          ASL R2                  ;TYPE IT OUT IN THE FROM OF ERROR CODE
3253  022172  006302                          ASL R2
3254  022174  006302                          ASL R2
3255  022176  010246                  MOV     R2,-(SP)        ;;SAVE R2 FOR TYPEOUT
3256  022200  104402                  TYPOS                   ;;GO TYPE--OCTAL ASCII
3257  022202     004                  .BYTE   4               ;;TYPE 4 DIGIT(S)
3258  022203     001                  .BYTE   1               ;;TYPE LEADING ZEROS
3259  022204  104400  015052                  TYPE ,MCRLF
3260  022210  005200                          INC R0                  ;SET UP FOR NEXT COUNTER
3261  022212  005721                          TST (R1)+               ;GET NEXT ERROR COUNTER
3262  022214  000745                          BR 2$
3263                        ;PRINTS TRACK ACCESS,HEAD MOVEMENT,AND ERRORS PER TRACK
3264  022216  005702          TYPTRK:         TST R2                  ;WAS AN ERROR PRINTED IN LAST GROUP
3265  022220  001004                          BNE 5$                  ;IF PRINTED ERROR FLAG NONZERO BRANCH
3266  022222  104400  015040                  TYPE ,TAB
3267  022226  104400  016311                  TYPE ,MNONE             ;TYPE NONE
3268  022232  104400  015240          5$:     TYPE ,DBLLF
```

```
3249  022236  005737  022514                        TST SHORTRPT        ;IF SHORT REPORT DON'T TYPE TRACK ACCESS OR ERRO
3250  022242  001121                                BNE 6$
3251  022244  104400  016223                        TYPE ,MTKLUG        ;TYPE FORMAT OF ERROR PRINT OUT
3252  022250  005037  022330                        CLR 2$              ;TRACK ADDRESS COUNTER
3253  022254  012704  017170                        MOV #TKACC-4,R4     ;SET UP ADDRESS OF TRACK ACCESSED
3254  022260  012703  017654                        MOV #HDMOVE-4,R3    ;ADDRESS OF HEAD MOVEMENT COUNTER
3255  022264  012702  020344                        MOV #U0TRK,R2       ;UNIT 0 ERROR PER TRACK COUNTER
3256  022270  012701  020576                        MOV #U1TRK,R1       ;UNIT 1 ERROR PER TRACK COUNTER
3257  022274  062704  000004          1$:           ADD #4,R4           ;ADJUST FOR ADDRESS OF NEXT COUNTER
3258  022300  005714                                TST (R4)            ;HAS THIS TRACK BEEN ACCESSED
3259  022302  001010                                BNE 10$             ;YES,PRINT OUT THE COUNTERS
3260  022304  005764  000002                        TST 2(R4)           ;TEST UPPER WORD OF ACCESS COUNTER
3261  022310  001005                                BNE 10$
3262  022312  062703  000004                        ADD #4,R3           ;THIS TRACK HAS NOT BEEN USED, ADJUST
3263  022316  005722                                TST(R2)+            ;REGISTERS TO POINT TO NEXT TRACK COUNTERS
3264  022320  005721                                TST (R1)+
3265  022322  000463                                BR 12$              ;TEST FOR LAST TRACK
3266  022324  004537  014446          10$:          JSR R5,SGLDEC       ;TYPE TRACK NUMBER
3267  022330  000000                  2$:           OPEN                ;THIS LOCATION USED AS TRACK ADDR COUNTER
3268  022332  104400  016170                        TYPE ,SPACE
3269  022336  010446                                MOV R4,-(SP)        ;TYPE TRACK ACCESSED COUNT
3270  022340  004737  014202                        JSR PC,@#$DB2D
3271  022344  004737  014376                        JSR PC,RTJUST
3272  022350  104400  016170                        TYPE ,SPACE
3273  022354  062703  000004                        ADD #4,R3
3274  022360  010346                                MOV R3,-(SP)        ;TYPE HEAD MOVED TO COUNT
3275  022362  004737  014202                        JSR PC,@#$DB2D
3276  022366  004737  014376                        JSR PC,RTJUST
3277  022372  104400  015047                        TYPE ,DBLSP
3278  022376  104400  016170                        TYPE ,SPACE
3279  022402  005712                                TST (R2)            ;ARE THERE ANY UNIT 0 ERRORS
3280  022404  001421                                BEQ 20$             ;NO,CHECK UNIT 1
3281  022406  012237  022416                        MOV (R2)+,3$        ;TYPE UNIT 0 ERRORS
3282  022412  004537  014446                        JSR R5,SGLDEC
3283  022416  000000                  3$:           OPEN
3284  022420  005711                                TST (R1)            ;ARE THERE ANY UNIT 1 ERRORS
3285  022422  001420                                BEQ 7$              ;NO,GO ADJUST POINTERS
3286  022424  104400  016170                        TYPE ,SPACE
3287  022430  104400  015047          22$:          TYPE ,DBLSP
3288  022434  012137  022444                        MOV (R1)+,4$        ;TYPE UNIT 1 ERRORS
3289  022440  004537  014446                        JSR R5,SGLDEC
3290  022444  000000                  4$:           OPEN
3291  022446  000407                                BR 11$              ;GO TO NEXT TRACK
3292
3293  022450  005722                  20$:          TST (R2)+           ;NO ERRORS,ADJUST REG.FOR NEXT COUNTER
3294  022452  005711                                TST (R1)            ;ARE THERE ANY UNIT 1 ERRORS
3295  022454  001403                                BEQ 7$              ;NO,GO TO NEXT TRACK
3296  022456  104400  015040                        TYPE ,TAB           ;YES,TYPE TAB FOR PLACEMENT
3297  022462  000762                                BR 22$              ;GO TYPE THE NUMBER
3298  022464  005721                  7$:           TST (R1)+           ;NO ERRORS,ADJUST REG. FOR NEXT COUNTER
3299  022466  104400  015052          11$:          TYPE ,MCRLF
3300  022472  005237  022330          12$:          INC 2$
3301  022476  023727  022330  000115                CMP 2$,#115         ;HAVE ALL TRACK ERRORS BEEN TYPED
3302  022504  001273                                BNE 1$              ;NO,TYPE THE NEXT ONE
3303  022506  005037  022514          6$:           CLR SHORTRPT        ;SET UP FOR LONG REPORT
3304  022512  000000                  HLT17:        HALT                ;YES DONE
```



```
3305
3306  022514  000000                  SHORTRPT:     0
3307  022516  000000                  ZERO:         0
3308
3309          000001                          .END
```

```
ADJSUM  005656          750    1160    1177#
AERRS   021316         3081    3099#
ASTAT   010070          822     979     986    1081    1113    1459*   1462    1465    1579#   1601*   1611
BADBYT  006436         1170*   1171    1172    1223    1264#
BERRS   022072         3186    3210#
BIT0  = 000001          138#    986
BIT00 = 000001          128#    138
BIT01 = 000002          127#    137
BIT02 = 000004          126#    136
BIT03 = 000010          125#    135
BIT04 = 000020          124#    134
BIT05 = 000040          123#    133
BIT06 = 000100          122#    132
BIT07 = 000200          121#    131
BIT08 = 000400          120#    130
BIT09 = 001000          119#    129
BIT1  = 000002          137*    822     979
BIT10 = 002000          118#
BIT11 = 004000          117#
BIT12 = 010000          116#
BIT13 = 020000          115#
BIT14 = 040000          114#   1788    1802
BIT15 = 100000          113#    520    1152    1594
BIT2  = 000004          136#    488     515    1462
BIT3  = 000010          135#
BIT4  = 000020          134#    880    1521    1555    1562    3041
BIT5  = 000040          133#
BIT6  = 000100          132#   1081    1113    1779    1784    1790
BIT7  = 000200          131#    498    1159    1218    1587
BIT8  = 000400          130#   1091    1118    1197    1485    1816
BIT9  = 001000          129#    804    1079    1380    1382    1384
BPTVEC= 000014          145#
BRONPA  010360         1651*   1659*   1661#
BRONTE  003014          711*    718*    719#
BRONTR  011320         1893*   1900*   1903#
BSTAT   010072          668    1493*   1494    1496    1500#   1608*   1617
BUFADR  016604         1177*   1178*   1182*   1185*   1186    1660    1756    2955#
BYTECN  006434          755*    760*   1157*   1174*   1219    1236*   1283#   1399    1409*
CHARCT  011142          706*   1811    1814*   1815*   1832*   1848#
CKBYTE  005626         1174#
CKSUM   005746         1161*   1171*   1179*   1180*   1181*   1182    1183*   1184*   1185    1189*   1252
CKSWR = 104405          654     771     777     844     849    1005    1010    1031    1036    1060    1067    1094    1120
                       1199    1232    1244    1262    1444    1449    1470    1479    1503    1822    2489#
COMMWD  004022          809     886#    944
CONT1   004366          951     960#
CONT10  005402         1080    1113#
CONT11  005276         1069    1094#   1122
CONT12  002542          647     651#
CONT13  003310          779     761#
CONT14  003650          851     853#
CONT15  002654         1012    1014#
CONT16  005000         1038    1040#
CONT17  005310         1096    1098#
CONT2   004436          961     970#
CONT20  006202         1234    1236#
CONT21  007244         1451    1453#
```



```
CONT22  011070         1824    1833#
CONT3   006412         1243    1276#
CONT4   003276          773     777#
CONT5   003336          782     785     787#
CONT6   005112         1050    1060#
CONT7   005076         1057#   1075
CR    = 000015           53#   2174    2184
CRCBAD  017050         2980#
CRCER   004560          987     995#
CRCLUG  017044         2978#
CRETRY  017020          937*   1016*   2964#
CRLF  = 000200           54#   2145    2184
DATABY  010416         1671#   1680*   1690*   1699*   1709*   1717*   1726*   1728*   1741*   1750*   1754    1755
DATACK  004126          906#    938*    998*   1206    1239
DATACR  004732          997    1001    1028#
DATAER  006000         1173    1197#
DATALO  017064         2986#
DATARE  017012          934*   1265*   2961#
DATA0   010422         1662    1680#
DATA1   010434         1663    1690#
DBLLF   015240          615     788     862     956     966    1022    1054    1072    1106    1272    1586    1593    2750#
                       3053    3069    3073    3096    3134    3215    3248
DBLSP   015047          587     605     610    1222    1226    1554    1565    1571    2709#   3088    3105    3113    3122
                       3150    3164    3169    3197    3202    3277    3287
DDCHK   006204          970    1079#
DDERR   005260         1091#   1124
DDISP = 177570           60#    346
DDMIS   017070         2988#
DDRETR  017010          658*    671*    933*   1100*   2960#
DELUNI  010074          793    1536    1582#
DISPLA  001216          346#    448*
DISPRE  000174          190#    448
DONE    010706         1800#   1808
DONEBI= 000040          161#   1405    1426    1467
DONECK  007122          487     510     660     810     945    1422#   1488    1524    1603
DRVSWP  006544          725    1330#
DRY1    001614          503     509#    514     523
DSWR  = 177570           59#    345
DTESTP  001212          344#    499     501     504     540     542     547     804    1079    1380    1382*   1384*
D0LOAD  014643          535    2682#
EBIE  = 000103          163*   1167
EMPBUF  005536          999    1154#   1194    1268    1279
EMPDUN  006212         1162    1239#
EMPER   005750         1163    1191#
EMPFLA  005622         1168#   1175    1237
EMTVEC= 000030          148#
ERCNTR  006442         1000    1158*   1198*   1204    1242    1248    1286#
ERCODE  017132         1499    3007#   3218
ERDUMP  021062          215    1598    3052#
ERRVEC= 000004          141#
ERWRT   004124          709*    813     905#   1065*
EXIT    006432         1156    1277    1281#
FBIE  = 000101          162#    757
FILLBU  003136          750#    766     786
FILLDO  003372          751     800#
FILLER  003214          752     763#
```

```
FILLFL  003176            758#    761
FIRST   001202            246#    247     595     600     602     607     617     625*    2086    2088*   2091    2094    2108
FLOATØ  010444           1664    1699#
FLOATI  010506           1665    1717#
FUNCTI  004120            803*    806*    892*    893     903#    941*    1514    1521    1582
GETPAT  010312            736     915     1135    1295    1312    1331    1358    1651#   1836
GETSEC  012316            747     932     2105#
GETTRA  011254            739     918     1138    1298    1315    1334    1364    1372    1892#
GETUNI  010614            738     917     1137    1297    1314    1333    1337    1363    1779#
GNS   = ****** U         2482    2483    2484    2485    2487    2489    2490    2491    2492    2493
GOODBY  006440           1227*   1229    1285#
GTSWR = 104404            454    2487#
HAROSK  003676            854     856     859#
HCNTR1  007200           1392*   1401*   1415*   1424*   1430*   1438#   1809*   1837*
HCNTR2  007202           1425*   1431*   1432    1439#
HCRCBA  017110           2996#
HCRCLO  017104           2994#
HDATAL  017124           3002#
HDDLOG  017130           3004#
HDMOVE  017660           1984    3023#   3254
HERE    011140           1840    1846#
HLTØ    004652           1013#
HLT1    004730           1026#
HLT12   004776           1039#
HLT13   005306           1097#
HLT14   006200           1235#
HLT15   007242           1452#
HLT16   011062           1831#
HLT17   022512           3304#
HLT20   007176           1435#
HLT21   007120           1418#
HLT3    002662            674#
HLT4    002132            570#
HLT5    002534            649#
HLT6    003306            780#
HLT7    003646            852#
HOME    002550            493     654#    857
HPARLO  017026            565*    787*   2970#   3108    3110
HRDLOG  017114           2998#
HSEKLO  017100           2992#
HT    = 000011             51#   2143    2184
HWRTLO  017120           3000#
ICOND   002452            635#    705    3062
ID    = 001201            245#    577     580     589    1858*   1862    1938    1973    1995    1998*
IDCOMP  012042           2047#   2050
IDNEXT  011552           1970    1973#
INITSE  012216            746     931    2086#
INITTR  011144            737     916    1136    1296    1313    1332    1362    1371    1855#
INTER   017034           1443*   2973#   3125    3127
INTLEA  012312           2097*   2102#   2109    2110*
INTSER  007250            460    1453    1459#
INTVEC  001204            268#    459     483
IOTVEC= 000020            146#
LAST  = 001203            247#    597     602     612    2089*   2090    2106
LASTTR  011722           2010    2013#
LF    = 000012             52#   2178    2184
```



```
LOAD    010564           1681    1704    1727    1751    1754#
LOGICA  011130            184    1842#
MABELL  015061           1821    2717#
MASTAT  015210           1610    2744#
MBADCR  015644           1008    1023    2822#   3147    3181
MBSTAT  015224           1616    2747#
MCODES  016172           2888#   3216
MCOLMU  014616           1217    2677#
MCRC    015736           1034    2838#   3148    3182
MCRLF   015052            563     568     631     635     645     776     792     874     959     969    1009    1024    1035
                         1057    1093    1109    1111    1213    1215    1231    1257    1275    1448    1474    1483    1506
                         1512    1519    1535    1609    1621    1813    1825    1830    2711#   3079    3099    3107    3116
                         3124    3132    3139    3173    3175    3207    3239    3299
MDDER   015615           1108    2815#   3177
MDDMIS  014506           1090    2656#   3143
MDELET  015126           1585    1592    2730#
MDERMD  014527           1214    1274    2661#   3144    3178
MDUMP   016055           2861#   3060
MEMPTY  015402           1192    1193    2777#
MEOP    015055           1820    2713#
MERCT   015333           1247    2768#
MEREUP  015057           1818    2715#
MERMEA  015063           1507    2719#
MERS    016067           1254    2864#   3138
MFIL    015366            764     765    2774#
MFIRST  015520            606    2798#
MGOOD   016322           1256    2908#
MHALT1  015724           1025    2835#
MHALT3  015713            673    2852#
MHALT4  015702            569    2829#
MHUNG   015432           1417    1434    2784#
MICON   015417            636    2781#
MID     015514            588    2796#
MINIT1  015535            490    2804#
MINIT2  015567            517    2810#
MINTER  015260           1447    2758#   3131
MINTVE  015150            482    2734#
MLAST   015527            611    2801#
MNODRY  015016            648    1534    2702#
MNOFLA  016034           1482    2857#   3123
MNONE   016311           2905#   3213    3247
MNONST  015446            630    2787#
MNOPAS  016125           2874#   3072
MOD     015510            583    2794#
MPAR    016017            562     567     775     791    2853#   3106    3115
MPASS   016142           1508    1826    2878#   3075
MPAT    015244           1560    2752#
MPRES   014567            670    2670#
MREAD   015674            958     982     983     990     991    1056    2827#   3087    3146    3160
MREC    015627            957     967    2818#
MRESTA  016111           2870#   3065
MREV    016554            441    2943#   3061
MRXCS   015167            472     555    1513    2738#
MRXDB   015177            477    2741#
MRX11   016344           1540    2914#
MSECT   014604           1549    2674#
```

```
MSEEK   015775          869    2847#   3149    3183
MSEQ    015254         1572    2756#
MSHORT  016076         2867#   3057
MSLASH  016165         2884#   3086    3092
MSUM    016327         1251    2910#
MTEST   015250         1566    2754#
MTKLOG  016223         2894#   3251
MTRK    014555         1545    2666#
MUKNIN  015312         1473    2764#
MUNIT0  015106          639    1557    1584    2724#   3135
MUNIT1  015116          642    1559    1591    2727#   3136
MUNREC  015755          566     789     863    1055    1075    1107    1273    2843#   3114    3203
MUNXDD  014462         1123    2651#   3142
MWRITE  016010          825     826     836     837     847     864     968    1074    2850#   3145    3179
MWRTEN  016154         2881#   3085
NEWTRK  011562         1928    1931    1942    1944    1972    1974    1981#   2000    2005    2016    2032    2034    2058
NEXTRD  004450          972     974#    993    1059    1112
NOERLO  017030         1478*   2971#   3117    3119
NOERTY  006170         1201    1205    1232#
NOINTE  007204          811     946    1443#
NOPRNT  007502         1495    1503#
NOSEL   001676          500     523#
OD      001200          244#    245     575     580     584     593     622*   1855*   1856    1859*   1860    1864    1868*
                       1924*   1927    1940*   1951    1953*   1968*   1969    1971    2001    2003*   2015    2023    2025*
                       2028    2033    2044*
ODCOMP  012062         2048    2051#   2054
ODNEXT  012012         2029    2033#
OD2BIG  016363          621    2918#
ONEWRT  003130          749#    831     858    1067
OPCOND  001702          502     505     507     524#
OPEN    = 000000        170#    586     591     609     614     774     790     873    1221    1250    1511    1548    1553
                       1629    2638    3068    3078    3104    3112    3121    3129    3163    3168    3172    3196    3201
                       3206    3228    3267    3283    3290
PARLOG  017024          464     554*    770*   2969#   3100    3102
PARTES  003244          767     770#    828     985    1195
PASCNT  021042          406    1509    1810*   1827    3039#   3070    3076
PAT     010420          547*    548*    549*    550*    551*   1561    1653    1655*   1656    1672#   1834
PATGEN  010426         1681#   1682    1691
PAT125  010514         1666    1726#
PAT314  010534         1667    1741#
PC      =%000007         72#    487*    493*    503*    510*    513*    521*    523*    564*    582*    604*    619*    633*
                        644*    651*    652*    660*    665*    667*    677*    705*    736*    737*    738*    739*    740*
                        743*    746*    747*    750*    758*    809*    810*    817*    836*    841*    843*    848*    857*
                        860*    865*    867*    875*    884*    894*    896*    901*    915*    916*    917*    918*    919*
                        920*    923*    931*    932*    944*    945*    953*    963*    970*    973*    976*    992*    995*
                        999*   1003*   1020*   1029*   1046*   1052*   1069*   1084*   1086*   1092*   1104*   1110*   1116*
                       1119*   1125*   1135*   1136*   1137*   1138*   1139*   1140*   1143*   1153*   1160*   1168*   1187*
                       1216*   1241*   1259*   1261*   1267*   1270*   1278*   1281*   1295*   1296*   1297*   1298*   1299*
                       1302*   1312*   1313*   1314*   1315*   1316*   1319*   1331*   1332*   1333*   1334*   1335*   1336*
                       1337*   1338*   1339*   1342*   1358*   1362*   1363*   1364*   1365*   1366*   1371*   1372*   1373*
                       1376*   1398*   1429*   1453*   1477*   1488*   1491*   1518*   1520*   1524*   1530*   1536*   1538*
                       1577*   1603*   1606*   1622*   1681*   1704*   1727*   1749*   1751*   1758*   1760*   1785*   1792*
                       1805*   1808*   1836*   1842*   1846*   1869*   1987*   2045*   2076*   2098*   2111*   2152*   2159*
                       2166*   2180*   2182*   2371*   2550    2552*   2555*   2592*   2597*   2641*   2646*   2647*   3045*
                       3062*   3090*   3091*   3094*   3095*   3270*   3271*   3275*   3276*
PCONT   003332          766*    786#    827*    984*   1194*
```



```
PIRQ    = 177772         58#
PIRQVE= 000240          152#
PRESCH  012102         2052    2055#
PRESTR  011362          676*    871    1892*   1914#   1925    1969    2028    2055
PRETRY  017016          749*    783*    784     800*    935*   2963#
PR0     = 000000         75#
PR1     = 000040         76#
PR2     = 000100         77#
PR3     = 000140         78#
PR4     = 000200         79#    168#    754
PR5     = 000240         80#
PR6     = 000300         81#
PR7     = 000340         82#    169#    438     701    1395
PS      = 177776         55#     56
PSW     = 177776         56#
PTYP1   003262          764*    774#    825*    982*   1192*
PTYP2   003352          765*    790#    826*    983*   1193*
PWRVEC= 000024          147#   2498*   2499*   2508*   2514*   2526*   2527*
P2RETR  017014          494*    560*    657*   1486*   2962#
RANDAT  010544         1668    1749#   1752
RANGEN  012122         1749    2045    2060#
RANUM   012214         1750    2046*   2047    2049*   2051    2053*   2055    2057    2075*   2080#
RAN1    012210          462*   2061    2067*   2072    2078#
RAN2    012212          463*   2062    2069    2074*   2079#
RDAFTW  004122          710*    904#    914*   1049    1134*   1530*   1361*   1369*
RDCHK   006444          723    1295#
RDCHR = 104406         2430    2490#
RDCNTR  021034          942*    943*   3035#   3093
RDCODE  010200          667    1601#
RDDONE  004316          939     950#
RDER    = 000017         160#   1487    1602
RDERR   004460          940     978#
RDIE    = 000107         165#    941
RDLIN = 104407         2491#
RDLOG   017054         2982#
RDONLY  006504          724    1312#
RDRETR  017004          468     707*    950     955*   1042*   1048*   2958#
RDOSTA= 000013          158#   1523
RDISTA= 000033          159#    509    1526
READ    004200          920     931#   1153   1316
READCH  005526         1140    1152#   1299   1336   1339   1366   1373
RECAL = 040001          167#    659
REREAD  004246          818     938#    984    989   1018   1044   1102   1267
RESREG= 104411         2493#   2596
RESTAR  002700          214     700#   1542   2531
RESTCN  021040          704*   3038#   3063   3066
RESVEC= 000010          142#
RETURN  005454         1082    1114   1125#
REWRIT  003400          801#    827
RTJUST  014376         2627#   2647   3091   3095   3271   3276
RTN     002676          656     677#
RXCS    001206          290#    473    491    509*   556    659*   665    757*   893*  1167*  1396   1405   1426
                       1460    1467   1487*  1523*  1526*  1602*
RXDB    001210          291#    478    488    496    511    515    518    661    759*   895*   897*  1170   1459
                       1489    1493   1528   1532   1601   1604   1608
RXERRO  007404         1461    1481   1485#
```

```
RXPWR   007702   1464    1540#
R0      =%000000   63#    759     1172    1186*   1227    1228*   1956*   1957*   1958*   2060*   2061*   2062*   2063*
                2065*   2066*   2067    2068*   2069*   2070*   2071*   2072*   2073*   2074    2075    2135    2136*
                2137    2140*   2288    2313*   2466    2467*   2468    2469*   2470*   2471*   2472*   2500    2525*
                2572*   2573    2591*   2595*   2627    2628*   2629    2630    2632    2634*   2636*   2639*   3217*
                3219    3224*   3231    3240*
R1      =%000001   64#    971     996     1152*   1159*   1208    1211    1218*   1280*   2289    2312*   2501    2524*
                2574*   2580*   2586*   3150*   3153    3155    3157    3161    3165    3166    3184*   3187    3189
                3191    3194    3198    3199    3218*   3222    3226    3241    3256*   3264    3284    3288    3294
                3298
R2      =%000002   65#    574*    628     632*    1701*   1703*   1706    2290    2311*   2502    2523*   2571*   2574
                2575*   2581*   2582*   2587*   2588*   3140*   3160*   3210    3214*   3231*   3232*   3233*   3234*
                3235    3244    3255*   3263    3279    3281    3293
R3      =%000003   66#    459*    460*    461*    840*    842*    859*    861*    952*    954*    962*    964*    1002*
                1004*   1019*   1021*   1028*   1030*   1045*   1047*   1051*   1053*   1068*   1070*   1083*   1085*
                1103*   1105*   1115*   1117*   1258*   1260*   1269*   1271*   1422*   1428    2226    2235*   2241*
                2242*   2245*   2250*   2251*   2252    2261*   2291    2310*   2426    2427*   2428    2431*   2432
                2436    2438    2440*   2442*   2503    2522*   2579*   2584*   2590*   2591    3043*   3254*   3262*
                3273*   3274
R4      =%000004   67#    715*    716*    717*    718     1656*   1657*   1658*   1659    1660*   1755*   1756    1897*
                1898*   1899*   1900    2227    2229*   2230*   2231*   2232    2233*   2247    2249*   2257*   2260*
                2292    2309*   2504    2521*   2577*   2580    2586    2588    5253*   3257*   3258    3260    3269
R5      =%000005   68#    464*    465*    466     468*    469*    470     585*    590*    608*    613*    872*    877*
                878*    879*    882*    883*    886*    887*    888*    889*    890*    891*    1220*   1249*   1496*
                1497*   1498*   1499*   1502*   1510*   1547*   1552*   1828*   1839*   1842    1981*   1982*   1983*
                1984*   1985*   1986*   2228    2254*   2236*   2238*   2239*   2240*   2241    2259*   2293    2308*
                2505    2520*   2578*   2582    2589    2645    2648*   3067*   3077*   3103*   3111*   3120*   3128*
                3162*   3167*   3195*   3200*   3227*   3266*   3282*   3289*
R6      =%000006   69#    71
R7      =%000007   70#    72
SAVREG= 104410   2492#   2570
SA200   001220   213     435#    571     650     675
SECCNT  012306   815*    974*    1154    1276*   2090*   2091*   2092*   2093*   2100#
SEEKER  003576   838     840#    992
SEKLOG  017040   2976#
SEKRTY  003672   851*    858#    989*
SEKTYP  003736   848     865     869#
SEQSAV  007012   1359*   1370    1387#
SEQUEN  011370   540*    541*    1359    1360*   1370*   1573    1894    1896*   1897    1917#
SEQ1    011372   1904    1924#   1960
SEQ2    011426   1905    1938#   1959
SEQ3    011460   1906    1951#
SEQ4    011524   1907    1968#
SEQ5    011610   1908    1995#
SEQ6    011742   1909    2023#
SEQ7    012022   1910    2044#   2056
SGLOEC  014446   585     590     608     613     872     1220    1249    1510    1547    1552    1828    2645#   3067
                3077    3103    3111    3120    3128    5162    3167    3195    3200    3227    3266    3282    3289
SHORTR  022514   1597*   3054    3060    3249    3303*   3306#
SP      =%000006   71#    437*    438*    439*    446     473*    478*    483*    556*    700*    701*    702*    751*
                752*    754*    755*    763     801*    802*    821     866*    898*    899*    939*    940*    978
                1058*   1066*   1162*   1163*   1164*   1165*   1191    1223*   1229*   1379*   1393*   1394*   1411
                1412*   1413*   1422    1428*   1476*   1514*   1537*   1561*   1567*   1573*   1611*   1617*   1759
                1804*   2135*   2136    2137*   2159    2140    2141*   2143    2145    2147    2153    2155*   2157*
                2165*   2169    2173    2174    2178    2218*   2219    2220    2221*   2226*   2227*   2228*   2234
                2259    2260    2261    2262*   2265*   2288*   2289*   2290*   2291*   2292*   2293*   2294*   2295*
```



```
                2296*   2297*   2304*   2305*   2306*   2307*   2308    2309    2310    2311    2312    2313    2331*
                2332*   2333    2340*   2345*   2344*   2348*   2349*   2353    2356*   2360    2362    2364    2365*
                2372    2374    2376*   2377    2379*   2380*   2381*   2382*   2383*   2398*   2399*   2402*   2403*
                2404    2408*   2409*   2410    2415    2415    2417*   2426*   2431    2442    2443*   2444*   2445*
                2466*   2467    2500*   2501*   2502*   2503*   2504*   2505*   2506*   2507    2515*   2519    2520
                2521    2522    2523    2524    2525    2530*   2550    2551*   2553*   2554*   2571    2573*   2627*
                2628    2639    2640*   2645*   3052*   3089*   3093*   3141*   3142*   3143*   3144*   3145*   3146*
                3147*   3148*   3149*   3151    3158    3170    3176*   3177*   3178*   3179*   3180*   3181*   3182*
                3183*   3185    3192    3204    3235*   3269*   3274*
SPACE   016170   2886#   3137    3268    3272    3278    3286
SRETRY  017022   470     748*    855*    936*    2965#
STACK = 001200   46#    437     700     3052
START1  001412   468#
STATER  002046   513     554#    663     1491    1530    1606
STKLMT= 177774   57#
STOP    010732   743     923     1143    1302    1319    1542    1376    1808#
STYP1   003626   836*    847#    990*
STYP2   003720   837*    864#    991*
SUM     010612   1179    1652*   1754*   1762#
SWHLT1  003636   846     849#
SWR     001214   345#    436*    443     447*    452     655     772     778     781     845     850     853     1006
                1011    1014    1032    1037    1040    1061    1088    1095    1098    1121    1200    1202    1233
                1245    1263    1445    1450    1471    1480    1504    1823    2327    2364*   2506    2519*
SWREG   000176   191#    447     452     2327    2340
SW0   = 000001   110#
SW00  = 000001   100#    110
SW01  = 000002   99#     109
SW02  = 000004   98#     108
SW03  = 000010   97#     107
SW04  = 000020   96#     106
SW05  = 000040   95#     105
SW06  = 000100   94#     104
SW07  = 000200   93#     103
SW08  = 000400   92#     102
SW09  = 001000   91#     101
SW1   = 000002   109#
SW10  = 002000   90#
SW11  = 004000   89#     781     853     1014    1040    1061    1098    1263
SW12  = 010000   88#     1202
SW13  = 020000   87#     772     845     1006    1032    1088    1121    1200    1245    1445    1471    1480    1504
SW14  = 040000   86#     1823
SW15  = 100000   85#
SW2   = 000004   108#
SW3   = 000010   107#
SW4   = 000020   106#
SW5   = 000040   105#
SW6   = 000100   104#
SW7   = 000200   103#
SW8   = 000400   102#     655
SW9   = 001000   101#
S2BIG   016455   624     2930#
TAB     015040   643     1615    2706#   3056    3059    5212    3229    3246    3296
TARGET  011360   877     886     897     1177    1180    1546    1860*   1892    1913#   1927*   1930*   1941*   1943*
                1971*   1973*   1981    1999*   2004*   2009*   2012*   2015*   2030*   2033*   2057*
TBITVE= 000014   143#
TEST    002140   542*    543*    544*    545*    546*    572#    712     714*    715     1567
```

```
TESTDR  001526   492    494#
TESTSE  002726   706*  1590
TEST7   006632   726   1358#
TEST7X  006642  1360*  1378   1383   1385
THREE   012314  2095   2103*  2105
TKACC   017174   889   3020*  3253
TKVEC = 000060   150#
TPVEC = 000064   151#
TRAPVE= 000034   149#
TRCK    007014   758    894    896   1168   1392#
TRKCNT  011356   741*   921*  1141*  1300*  1317*  1340*  1367*  1374*  1865*  1866*  1867*  1912#  1954*
                1955*  1958   1997*  2006   2013   2026*  2027*
TRKERR  003770   843    877#   995   1086   1119   1261
TRTVEC= 000014   144#
TSECTO  012310   895   1178   1181   1550   2094*  2095*  2101#  2105*  2106   2108*  2109*
TSTLIM  002142   574#   651
TSTMSG  002432   582    604    628#
TYPADR  007716  1092   1110   1216   1518   1545#
TYPCOD  010250  1520   1610#
TYPE  = 104400   441    472    477    482    490    517    535    555    562    563    566    567    568
                569    583    587    588    605    606    610    611    615    621    624    630    631
                635    636    639    642    643    645    648    673    774    775    776    788    789
                790    791    792    847    862    863    864    869    870    874    956    957    958
                959    966    967    968    969   1008   1009   1022   1023   1024   1025   1034   1035
               1054   1055   1056   1057   1072   1073   1074   1090   1093   1106   1107   1108   1109
               1111   1123   1213   1214   1215   1217   1222   1226   1231   1247   1251   1254   1256
               1257   1272   1273   1274   1275   1417   1434   1447   1448   1473   1474   1482   1483
               1506   1507   1508   1512   1513   1519   1534   1535   1540   1541   1545   1549   1554
               1557   1559   1560   1565   1566   1571   1572   1584   1585   1586   1591   1592   1593
               1609   1610   1615   1616   1621   1813   1818   1820   1821   1825   1826   1830   2148
               2253   2338   2339   2342   2355   2366   2385   2434   2437   2441   2482#  2528   2637
               3053   3056   3057   3059   3060   3061   3065   3069   3072   3073   3075   3079   3085
               3086   3087   3088   3092   3096   3099   3105   3106   3107   3113   3114   3115   3116
               3122   3123   3124   3130   3131   3132   3134   3135   3136   3137   3138   3139   3164
               3169   3171   3173   3175   3197   3202   3203   3205   3207   3212   3213   3215   3216
               3229   3239   3246   3247   3248   3251   3268   3272   3277   3278   3286   3287   3296
               3299
TYPERR  006114   1209   1212   1219#
TYPOC = 104401   2341   2483#
TYPON = 104403   1230   2485#
TYPOS = 104402   474    479    484    557   1224   1515   1562   1568   1574   1612   1618   2484#  3236
TYPSEL  010012   644   1558   1560#
TYPTMK  022216  3221   3244#
UKNINT  017032  1469*  2972#
UNITSE  010704   495*   498*   506*   520*   524   533*   637    640    646    880    892   1091*  1118*
                1197*  1377   1485*  1555   1587*  1588   1594*  1595   1779   1781   1783*  1784*  1786
                1788   1790*  1791*  1796#  1800   1802   1816   1833*  3041
UNXDD   017074  2990#
U0TMK   020344   879   3028#  3255
U1LOG   021044   841    860    953    963   1003   1020   1029   1046   1052   1069   1084   1104   1116
                1259   1270   3041#
U1TMK   020576   882   3031#  3256
WRITE   003112   740    746#   919   1139   1335   1338   1365
WRTDON  003462   801    813#
WRTER   003510   802    821#
WRTIE = 000105   164#   803
```



```
WRTLOG  017060  2984#
WRTONL  003052   720    736#
WRTRD   004130   721    914#
WRTSEK  003550   823    831#
WTCNTP  021030   807*   808*  3034*  3089
WTDDIE= 000115   166#   806
WTRDCK  005456   722   1134#
WTRETR  017006   708*   960    965*  1063*  1071*  2959#
XFRBYT  003202   759#
XHOME   002670   666    670    676#
XID     011366  1861*  1862*  1865   1916#  1925   1941   1956   1999   2011*  2012   2030   2031*  2047
                2049
XOD     011364  1863*  1864*  1866   1915#  1957   2004   2008*  2009   2051   2053
XPATGE  010452  1700*  1718
XRDCMK  006450  1296*  1303
XRDONL  006510  1313*  1320
XREAD   004204   932#   975   1278
XRETUR  011740  2014   2016#
XSTART  002722   552    705#
XWRITE  003116   747#   819
XWRTDN  003056   737#   744
XWRTRD  004140   916#   924
XWTRDC  005466  1136#  1144
XXPATG  010522  1727*  1729   1742
ZCRCDA  017046  1002   2979#
ZCRCLO  017042  1028   2977#
ZDATAL  017062  1258   2985#
ZDDMIS  017066  1083   2987#
ZERO    022516  3141   3176   3307#
ZHCRCB  017106  1019   2995#
ZHCRCL  017102  1045   2993#
ZHDATA  017122  1269   3001#
ZHDDLO  017126  1103   3003#
ZHRDLO  017112  1051   2997#
ZHSEKL  017076   859   2991#  3184
ZHWRTL  017116  1068   2999#
ZRDLOG  017052   952   2981#
ZSEKLO  017036   840   2975#  3150
ZUNXDD  017072  1115   2989#
ZWRTLO  017056   962   2983#
$AUTOB  013714   456*  2335   2456#
$CHARC  012572  2150*  2160*  2167   2176*  2181#
$CKSWR  013140  2327#  2489
$CNTLG  013665  2338   2451#
$CNTLU  013660  2355   2450#
$CRLF   012607  2149   2191#  2366   2450
$DB2U   014202  2552   2570*  3090   3094   3270   3275
$DECVL  014362  2572   2618#
$FILLC  012604  2153   2188#
$FILLS  012603  2187#
$GTSWR  013210  2339#  2487
$HD   = 000003    20     21
$ILLUP  014130  2498   2514   2533#
$INTAG  013715  2367   2457#
$LF     012610  2192#  2441   2450
$MAIL = ****** U  452   2137
```

```
$MNEW   013703            2342    2454#
$MSWR   013672            2339    2452#
$NULL   012602            2155    2186#
$OCNT   013034            2225*   2254*   2267#
$OMODE  013036            2220*   2224*   2229    2232*   2243*   2269#
$POWER  014136            1541    2529    2536#
$PWRAD  014124            2531#
$PWRDN  013764             176    2498#   2526
$PWRMG  014120            2529#
$PWRUP  014036            2508    2514#
$QUES   012606            2190*   2385    2434    2450
$RDCHR  013422            2398#   2490
$RDDEC= ****** U          2492
$RDLIN  013542            2426#   2491
$RDOCT= ****** U          2492
$RDSZ = 000010            2419#
$RESRE  013076            2303#   2493
$R2A  = ****** U          2494
$SAVRE  013040            2287#   2492
$SAVR6  014134            2507*   2515    2516*   2517*   2535#
$SB2D   014146            2550#   2646
$SETUP= 000114             208#    449    2322    2456
$STUP = 177777             208#
$SWR  = 160000              20      21#   2532
$TKB    013136            2319#   2331    2348    2402    2408
$TKS    013134            2318#   2329    2345    2369*   2400    2406
$TN   = 000001              20#
$TNPWR  014312            2577    2578    2598#
$TPB    012600            2173*   2185#
$TPFLG  012605            2131    2189#
$TPS    012576            2171    2184#
$TRAP   013716             180    2466#
$TRP  = 000012            2474#   2483#   2484#   2485#   2466*   2487    2488#   2489    2490#   2491#   2492#   2493#   2494#
$TRPAD  013740            2471    2481#
$TTYIN  013650            2427    2428    2445    2449#
$TYPBN= ****** U          2486
$TYPDS= ****** U          2486
$TYPE   012356            2131#   2474    2482
$TYPEC  012526            2152    2159    2166    2171#   2172    2371
$TYPEX  012574            2177    2179    2182#
$TYPOC  012636            2223#   2463
$TYPON  012652            2222    2225#   2485
$TYPOS  012612            2218#   2484
$0FILL  013035            2219*   2223*   2233    2268#
.     = 022520             172#    174     175#    179#    183#    186#    189#    209#    243#    719    1661    1903    2184
                          2185    2186    2187    2188    2189    2190    2191    2192    2193    2318    2319    2449#   2450
                          2456    2457    2458    2510    2554    2618#   2948#   2955#   3007#   3020#   3023#   3028#   3031#
```



```
COMMEN      1#    153#
ENDCOM      1#    153#
ERROR      47#
ESCAPE      1#    153#
GETPRI      1#    153#
GETSWR      1#      6#    153#    449
MULT        1#    153#
NEWTST      1#    153#
POP         1#    153#   2308    2519    2520
PUSH        1#    153#   2288    2500    2506
REPORT      1#    153#
SCOPE      48#
SETPRI      1#    153#
SETTRA   2474#   2483    2484    2485    2487    2489    2490    2491    2492    2493
SETUP       1#    153#
SKIP        1#    153#
SLASH       1#    153#
SPACE     153#
STARS       1#      6#    153#    194     218     250     270     288     348     413     431     679     698     730     798
          908     926    1129    1147    1288    1306    1324    1346    1388    1455    1625    1647    1675    1684    1693
         1712    1720    1735    1744    1764    1777    1850    1873    1890    1919    1933    1946    1963    1976    1990
         2018    2038    2082    2116    2195    2272    2317    2522    2390    2419    2460    2496    2512    2541    2559
         2620    2950    3050
SWRSU       1#    153#
TRMTRP   2474#
TYPBIN      1#    153#
TYPDEC      1#    153#
TYPNAM      1#    153#
TYPNUM      1#    153#
TYPOCS      1#    153#    473     478     483     556    1514    1561    1567    1573    1611    1617    3235
TYPOCT      1#    153#   2340
TYPTXT      1#    153#
$ESCA       1#    153#
$NEWT       1#    153#
$SET     2474#   2483    2484    2485    2487    2489    2490    2491    2492    2493
$SKIP       1#    153#
.EQUAT      1#      5#     43
.HEADE      1#      5#     10
.KT11       1#
.SETUP      1#      5#    208
.SWRMI      1#
.SACT1      1#
.SAPTB      1#
.SAPTH      1#
.SAPTY      1#
.SASTA      1#
.SCATC      1#
.SCMTA      1#
.SDB2D      1#      6#   2557
.SDB2O      1#
.SDIV       1#
.SEOP       1#
.SERRO      1#
.SERRT      1#
.SMULT      1#
.SPOWE      1#      6#   2494
```

```
.$RAND      1#
.$RDUE      1#
.$RDUC      1#
.$READ      1#      5#   2315
.$R2AZ      1#
.$SAVE      1#      6#   2270
.$SR2D      1#      6#   2539
.$SB2D      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#      6#   2458
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#      5#   2114
.$TYPO      1#      5#   2193
.$40CA      1#
.1170       1#
```



```
ADC      808.   891   943  1431  1986  2587
ADD      807    866   879   882   889   890   942  1058  1066  1171  1180  1181  1430  1476  1499
        1537   1754  1804  1964  1985  2053  2061  2062  2072  2105  2109  2141  2221  2231  2356
        2365   2553  2586  2588  3043  3257  3262  3273
ASL      717    878   867   888  1658  1899  1955  1982  1983  2027  2379  2380  2381  2470  3232
        3233   3234
ASLB    1183
BCC     1702
BEQ      444    503   512   516   532   561   576   579   596   599   601   662   785   805   823
         856    881   951   961   980  1017  1043  1064  1080  1101  1114  1203  1207  1240  1243
        1253   1266  1277  1381  1400  1408  1427  1465  1490  1495  1529  1605  1707  1757  1817
        1824   1840  1970  2056  2144  2179  2248  2336  2363  2378  2631  3042  3055  3064  3101
        3109   3118  3126  3152  3186  3280  3285  3295
BGE      670   1959  2052  2107
BGT     1205   2255  2375  2416
BHI      578    581   598   603
BIC      533    541   543   551  1218  1382  1551  1587  1594  1783  1790  1833  1855  1924  1940
        1953   1968  2025  2044  2046  2063  2073  2245  2332  2349  2376  2403  2409  2417
BICB     711   1651  1893  1998  2003
BIS      498    520  1091  1118  1152  1159  1197  1384  1485  1784  1791  1868  2250  2251  2383
        2590
BISB     718    892  1659  1900
BIT      488    499   504   515   655   661   772   781   804   822   845   853   880   979   986
        1006   1014  1032  1040  1061  1079  1088  1098  1113  1121  1200  1202  1245  1263  1377
        1380   1426  1445  1462  1465  1471  1480  1489  1504  1521  1528  1555  1604  1779  1788
        1802   1816  1823  3041
BITB     511   1081  1405  1467  1582  2006  2167
BLE     2048
BLOS    2429
BLT     2158   2256  2373  2414  2583
BMI     1461   1533  1596
BNE      451    453   467   471   489   505   530   594   618   629   647   656   672   713   742
         773    782   814   816   846   854   922   975   987  1001  1007  1015  1033  1041  1050
        1062   1082  1089  1099  1122  1142  1155  1173  1201  1246  1264  1301  1318  1341  1368
        1375   1378  1404  1416  1433  1446  1466  1468  1472  1481  1505  1522  1556  1583  1654
        1780   1789  1803  1812  1835  1838  1857  1895  1926  2007  2014  2029  2087  2138  2146
        2154   2168  2175  2246  2328  2334  2354  2361  2368  2405  2411  2433  2439  2518  2594
        2633   3071  3081  3154  3156  3188  3190  3211  3220  3223  3245  3250  3259  3261  3302
BPL      492    497   502   519   525   638   641   666   779   851   972   997  1012  1038  1096
        1209   1212  1234  1397  1451  1589  1782  1787  1801  1939  1952  1996  2002  2024  2132
        2172   2244  2330  2346  2401  2407
BR       210    211   212   445   455   507   514   534   592   616   623   626   664   719   744
         761    924   993  1075  1124  1144  1175  1210  1255  1303  1320  1343  1383  1385  1492
        1525   1527  1531  1558  1607  1661  1682  1691  1710  1718  1729  1742  1752  1819  1903
        1928   1931  1942  1944  1960  1972  1974  2000  2005  2010  2016  2032  2034  2050  2054
        2058   2134  2151  2161  2170  2177  2222  2237  2258  2357  2384  2386  2412  2435  2510
        2534   2585  2635  3058  3074  3159  3174  3193  3208  3221  3225  3242  3265  3291  3297
CLC      539   1705  2064
CLR      465    495   574   622   625   706   709   710   753   898   938  1157  1158  1161  1164
        1280   1369  1392  1409  1412  1424  1652  1680  1717  1809  1832  1861  1863  2068  2235
        2343   2344  2516  2579  3140  3214  3252  3303
CLRB     506   1859  2093  2150  2176  2440  2595
CMP      443    446   452   466   470   950   960  1204  1399  1756  1834  2327  2333  2353  2360
        2372   2374  2404  2410  2413  2415  2428  2589  3219  3301
CMPB     529    577   580   597   602   617   668  1172  1925  1969  2013  2028  2047  2051  2055
        2106   2143  2145  2153  2174  2178  2335  2367  2432  2438  2632
```

```
        2456    2460    2466    2470    2474    2483    2484    2485    2486    2487    2489    2490    2491    2492    2493
        2494    2496    2506    2507    2512    2519    2520    2528    2530    2532    2536    2541    2559    2620    2950
        3050    3237    3238
.IFF      47     195     219     251     271     289     349     414     432     475     476     480     481     486     559
         680     699     731     749     909     927    1130    1148    1289    1307    1325    1347    1389    1456    1517
        1564    1570    1576    1614    1620    1626    1648    1676    1685    1694    1715    1721    1736    1745    1765
        1778    1851    1874    1891    1920    1934    1947    1964    1977    1991    2019    2039    2083    2117    2196
        2273    2318    2323    2391    2393    2398    2419    2420    2429    2433    2450    2461    2467    2497    2513
        2530    2542    2560    2621    2951    3051    3238    3239
.IFT    2393    2398
.IFTF   2338    2391    2394
.IIF      10      15      20      21     474     479     484     557    1515    1562    1568    1574    1612    1618    2184
        2185    2186    2187    2188    2189    2190    2191    2192    2193    2318    2319    2341    2442    2450    2456
        2457    2458    2482    2483    2484    2485    2487    2489    2490    2491    2492    2493    3236
.IRP     208    2288    2308    2500    2506    2519    2520
.LIST      1       3     153     174     208    2419    2474    2482    2483    2484    2485    2486    2487    2488    2489
        2490    2491    2492    2493    2494
.MACRO     1    2474
.MCALL     5       6     153
.NLIST     1       2     153     174     208    2419    2474    2482    2483    2484    2485    2486    2487    2488    2489
        2490    2491    2492    2493    2494
.REPT    174
.SBTTL    43     449    2114    2193    2270    2315    2458    2474    2494    2539    2557
.TITLE    10
.WORD    173     187     344     345     346    1225    2181    2184    2185    2269    2318    2319    2529    2531    2556
        2593    3034    3035
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZRXAE,SEQ/SOL/CRF=SYSMAC.SML,DZRXAE.P11
RUN-TIME:  42 44 6 SECONDS
RUN-TIME RATIO: 375/93=4.0
CORE USED:  34K  (67 PAGES)
```