

IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DHTDA-A-D
 REPLACES: MAINDEC-8E-D3AB-D

PRODUCT NAME: TD8E DECTAPE DIAGNOSTIC

DATE CREATED: NOVEMBER 1, 1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: E. STEINBERGER/B. HANSEN

1, ABSTRACT

TD8E DECTAPE DIAGNOSTIC IS A PROGRAM WHICH HAS BEEN WRITTEN TO CHECKOUT AND TEST TD8E DECTAPE CONTROLS WITH TU56 DECTAPE TRANSPORTS; THE PROGRAM TESTS THE BASIC FUNCTIONS OF THE CONTROL (IOT SKIPS, DATA TRANSFERS, ETC) AS WELL AS CHECKING THE ABILITY TO READ AND WRITE ON DECTAPE,

2, REQUIREMENTS

2,1 EQUIPMENT

PDP-8E
TD8E DECTAPE CONTROL
TU56 DECTAPE TRANSPORT (AT LEAST ONE)
ALL NECESSARY CABLES AND MODULES

2,2 STORAGE

THE PROGRAM OCCUPIES MEMORY FROM LOCATION 20 TO LOCATION 9177 AND USES LOCATIONS 7200 TO 7577 AS DATA BUFFER AREA,

2,3 PRELIMINARY PROGRAMS

NONE

#, LOADING PROCEDURE

3,1 METHOD

THE PROGRAM IS LOADED USING THE STANDARD BINARY LOADER TECHNIQUE;

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

THE FOLLOWING IS A TABLE OF CONTROL SWITCH SETTINGS AND THEIR ACTION UPON THE PROGRAM:

SR	STATE	ACTION
0	1	LOOP ON CURRENT SUBTEST
	0	DON'T LOOP
1	1	LOOP ON CURRENT TEST
	0	DON'T LOOP
2	1	LOOP ON CONTROL TESTS
	0	DON'T LOOP
3	1	DON'T PRINT ERRORS
	0	PRINT ERRORS
4	1	DON'T HALT ON ERRORS
	0	HALT ON ERROR
5	1	
	0	
6	1	
	0	
7	1	
	0	
8	1	
	0	
9	1	
	0	
10	1	
	0	
11	1	SINGLE UNIT TRANSPORT
	0	DUAL UNIT TRANSPORT

4.2 STARTING ADDRESSES

0200	OPERATOR INTERVENTION TESTS
0201	CONTROL AND DATA TRANSFER TESTS
2100	SEARCH AND FIND ALL BLOCK NUMBERS
2200	DISPLAY BLOCK NUMBERS IN AC
2237	ROUTINE TO ROCK DECTAPE 0 (TIME DEPENDENT ON SWITCH REGISTER)
2400	READ AND CHECK THE MARK TRACK FROM ENDOZONE TO ENDOZONE
7200	IOT MODIFICATION PROGRAM

4,3 PROGRAM AND/OR OPERATOR ACTION

4,3,1 TO TEST "SELECT ERROR" AND "WRITE LOCK OUT"

4,3,1,1 DUAL TRANSPORTS

- A) SET SWITCH REGISTER TO 0200
- B) ON THE TRANSPORTS, SET ONE TRANSPORT TO UNIT 0, ON=LINE, WRITE LOCK; SET THE OTHER TRANSPORT TO UNIT 1, OFF=LINE;
- C) DEPRESS "LOAD ADDRESS", THEN "CLEAR", THEN "CONTINUE", THE PROGRAM SHOULD TYPE "OK"
- D) REVERSE THE ROLES OF THE TWO TRANSPORTS AND REPEAT STEP C;
- E) SET BOTH TRANSPORTS TO UNIT 1, ON=LINE; DEPRESS "LOAD ADDRESS", THEN "CLEAR", THEN "CONTINUE"; THE PROGRAM SHOULD INDICATE NO UNIT 0 SELECTED
- F) PROCEED TO 4,3,2

4,3,1,2 SINGLE TRANSPORT

- A) SET SWITCH REGISTER TO 0200
- B) ON THE TRANSPORT, SET TO UNIT 0, ON=LINE, WRITE LOCK
- C) DEPRESS "LOAD", THEN "CLEAR", THEN "CONTINUE", THE PROGRAM SHOULD TYPE "OK"
- D) PROCEED TO 4,3,2

4,3,2 TO TEST CONTROL AND ABILITY TO PERFORM DATA TRANSFERS

- A) SET SWITCH REGISTER TO 0201, DEPRESS "LOAD ADDRESS"

- B) SET SWITCH REGISTER PER 4;1, SET SR11 IF ONLY ONE TRANSPORT EXISTS OR ONLY ONE TRANSPORT IS TO BE TESTED,
- C) MOUNT A STANDARD PDP-8 DECTAPE (2702 BLOCKS, 201 WORDS PER BLOCK) ON EACH TRANSPORT TO BE TESTED WITH THE TAPES WRAPPED AT LEAST 2 TURNS ON EACH TAKE UP REEL, RESPECTIVELY,
- D) SET A TRANSPORT TO UNIT 0, ON-LINE, WRITE ENABLE; SET THE OTHER TRANSPORT (IF IT EXISTS OR IS TO BE TESTED) TO UNIT 1, ON-LINE, WRITE ENABLE,
- E) DEPRESS "CLEAR", THEN "CONTINUE", THE PROGRAM WILL PERFORM THE BASIC CONTROL TESTS ON THE TD8E, AND, IF SR2 IS A 0, PROCEED TO MOVE TAPE AND PERFORM DATA TRANSFERS TO AND FROM TAPE, CHECKING THE RESULTS;

4,3,3 TO MODIFY THE TD8E IOT SET TO HANDLE A CONTROL FOR UNITS OTHER THAN 0 AND 1,

- A) SET SWITCH REGISTER TO 7200, DEPRESS "LOAD ADDRESS"
- B) SET SWITCH REGISTER BITS 6, 7 AND 8 TO DEVICE SELECTOR BITS 6, 7, AND 8 OF THE CONTROL TO BE TESTED (4, 5, 6 OR 7)
- C) DEPRESS "CLEAR", THEN "CONTINUE", THE PROGRAM WILL MODIFY ALL TD8E IOT'S TO HANDLE THE SELECTED CONTROL;
- D) PERFORM ALL TESTS INDICATED IN 4;3;1 AND 4;3;2 ABOVE FOR THE SELECTED CONTROL SUBSTITUTING UNIT 2, 4 OR 6 FOR UNIT 0 AND UNIT 3, 5 OR 7 FOR UNIT 1 ABOVE,
- E) CAUTION- THE CODE TO CHANGE THE IOT'S IS IN THE DATA BUFFER AREA FOR THE DATA TRANSFER TESTS AND WILL BE DESTROYED WHEN THAT PORTION OF THE PROGRAM IS RUN; AN OVERLAY TAPE IS PROVIDED TO ALLOW THIS CODE TO BE READ BACK INTO MEMORY FOR RE-EXECUTION,
MAINDEC=00=DHTOA=A=PB2

5, OPERATING PROCEDURE

5,1 OPERATIONAL SWITCH SETTINGS

SEE 4,1

5,2 SUBROUTINE ABSTRACTS

NONE

5,3 PROGRAM AND/OR OPERATOR ACTION

SEE 4,3

5,3,1 IF PROBLEMS ARE SUSPECTED IN THE CONTROL WHEN READING THE TIMING TRACK OFF OF DECTAPE INCLUDING SINGLE LINE FLAG AND QUAD LINE FLAG, A ROCK TAPE ROUTINE HAS BEEN PROVIDED AT LOCATION 2237 TO ALLOW 'SCOPING OF SINGLE LINE FLAG, QUAD LINE FLAG, UP=TO=SPEED, ETC,

- A) SET SWITCH REGISTER TO 2237, DEPRESS "LOAD ADDRESS"
- B) SET SWITCH REGISTER TO 0070, DEPRESS "CLEAR" THEN "CONTINUE", THE DECTAPE ON UNIT 0 SHOULD START ROCKING,
- C) MODIFY SWITCH REGISTER SETTING TO INCREASE OR DECREASE "ROCK" PERIOD,
- D) CAUTION-IF THE NUMBER IN THE SWITCH REGISTER IS TOO SMALL, THE DECTAPE TRANSPORT WILL NOT GET UP TO SPEED BEFORE IT TURNS AROUND,

5,3,2 A ROUTINE HAS BEEN PROVIDED AT LOCATION 2100 TO ALLOW A DECTAPE TO BE RUN FROM ENDEONE TO ENDEONE COMPARING ALL BLOCK NUMBERS; TO RUN THIS ROUTINE, START THE COMPUTER AT LOCATION 2100, THE SR HAS NO AFFECT UPON THE ROUTINE, TO RUN UNIT 1 CHANGE THE CONTENTS OF UNIT (LOCATION 2234) TO 4000, A HALT WILL OCCUR AT LOCATION 2150 IF AN ERROR OCCURS, THE CONTENTS OF THE AC EQUALS THE BLOCK THAT WAS BEING SEARCHED FOR, PRESS "CONT" AND THE PROGRAM WILL HALT AT LOCATION 2153 WITH THE AC EQUAL TO THE BLOCK THAT WAS FOUND, PRESS "CONT" AGAIN, THE PROGRAM WILL RECYCLE UNTIL ANOTHER ERROR IS FOUND,

5,3,3 A ROUTINE HAS BEEN PROVIDED AT LOCATION 2200 TO ALLOW A DECTAPE TO BE RUN FROM ENDEONE TO ENDEONE WITH THE BLOCK NUMBERS DISPLAYED IN THE AC, TO RUN THIS ROUTINE, START THE COMPUTER AT LOCATION 2200, THE SR HAS NO AFFECT UPON THE ROUTINE, TO RUN UNIT 1, CHANGE THE CONTENTS OF UNIT (LOCATION 2234) TO 4000, NO ERRORS ARE DETECTED,

5,3,4 A ROUTINE HAS BEEN PROVIDED AT LOCATION 2400 TO ALLOW A DECTAPE TO BE RUN FROM ENDEONE TO ENDEONE WITH THE MARK TRACK FORMAT BEING CHECKED, AN ERROR HALT WILL OCCUR IF AN ERROR IS DETECTED, TO RUN THIS ROUTINE (ONLY ON UNIT 0), START THE COMPUTER AT LOCATION 2400, THE SR HAS NO AFFECT UPON THE ROUTINE.

6, ERRORS

6,1 ERROR HALTS AND DESCRIPTION

MOST ERROR HALTS ARE PRECEDED BY AN ERROR TYPEOUT (UNLESS SR3 IS A 1); IF NO ERROR TYPEOUT OCCURS, CONSULT THE LISTING FOR THE CAUSE OF THE ERROR;

6,2 ERROR RECOVERY

MOST ERRORS (EXCEPT DATA ERRORS) CAN BE 'SCOPED BY SETTING SR0 TO 1 AND DEPRESSING "CONTINUE"

DATA ERRORS CANNOT BE 'SCOPED, BUT DATA TRANSFERS CAN BE CONTINUED BY DEPRESSING "CONTINUE".

6,3 IF TAPE RUNS OFF THE END

NORMALLY, TAPE WILL NOT RUN OFF THE END OF THE REEL UNLESS THE PROGRAM IS IN A 'SCOPE LOOP OR A SELECT ERROR OCCURS WHEN A TAPE IS MOVING (THE OPERATOR SETTING BOTH TAPE UNITS TO THE SAME NUMBER DURING THE DATA TRANSFER TESTS).

IF TAPE RUNS OFF THE END AND THE PROGRAM HANGS AROUND LOCATION 0621, CHECK THE ABILITY TO READ THE TIMING TRACK INTO THE TDBE CONTROL AND THE CIRCUITS RELATING TO THE TIMING PULSE GENERATOR.

IF TAPE RUNS OFF THE END AND THE PROGRAM HANGS AROUND LOCATION 1466, CHECK THE ABILITY TO READ THE MARK TRACK INTO THE TDBE CONTROL AND THE CIRCUITS RELATING TO THE MARK TRACK REGISTER.

7, RESTRICTIONS

7,1 STARTING RESTRICTIONS

NONE IF PARAGRAPH 4,3 IS PROPERLY FOLLOWED;

7,2 OPERATING RESTRICTIONS

NONE IF PARAGRAPH 4,3 IS PROPERLY FOLLOWED;

8, MISCELLANEOUS
-----8,1 EXECUTION TIME

THE EXECUTION TIME OF THE CONTROL TESTS IS NORMALLY LESS THAN 1 MINUTE, DEPENDING UPON THE POSITION OF TAPE ON UNIT 0;

THE EXECUTION TIME OF THE DATA TESTS DEPENDS ON WHETHER ONE OR TWO TRANSPORTS IS BEING EXERCISED, PASS "N" COMPLETE WILL BE PRINTED ON THE TELEPRINTER AFTER ALL DATA PATTERNS HAVE BEEN EXERCISED ONCE, (NORMALLY LESS THAN 1 HOUR PER PASS);

9, PROGRAM DESCRIPTION
-----9,1 DATA REGISTER TEST (SA#0201)

IN THIS TEST THE DATA REGISTER IS CHECKED FOR ITS ABILITY TO BE LOADED AND READ, FIRST THE COMPLEMENT OF THE DATA TO BE CHECKED IS LOADED INTO THE DATA REGISTER, THEN THE DATA ITSELF IS LOADED INTO THE REGISTER, THIS IS DONE TO CHECK THAT ALL BITS CAN BE LOADED TO A 1 FROM A 0 AND TO A 0 FROM A 1, THE DATA IS THEN READ INTO THE AC AND CHECKED FOR ERRORS, AN INCREMENT PATTERN IS USED,

9,2 COMMAND REGISTER TEST (SA#0236)

IN THIS TEST THE COMMAND REGISTER IS CHECKED FOR ITS ABILITY TO BE LOADED AND READ, DATA IS LOADED INTO THE COMMAND REGISTER THEN READ IN THE AC AND CHECKED FOR ERRORS, A 400 INCREMENT PATTERN IS USED, THE STOP/GO BIT IS MASKED OUT,

9,3 INITIALIZE TEST (SA#0305)

THIS TESTS CHECKS THAT "CAF" CLEARS THE COMMAND REGISTER, THE C,R, IS LOADED WITH 6400, THEN "CAF" IS ISSUED, THE C,R, IS THEN READ AND CHECKED TO CONTAIN 0,

9,4 CHECK SDLC, SOLD, SORC, AND SORD AND AC CLEAR (SA#0400)

THIS TEST CHECKS THE AC CLEAR FUNCTION OF THE SDLC, SOLD, SORC AND SORD INSTRUCTION, THIS IS DONE BY SETTING THE AC TO 7777, THEN ISSUING THE APPROPRIATE IOT (ONE AT A TIME) AND CHECK TO SEE IF THE AC DID OR DID NOT CLEAR (SOLD DOES NOT CLEAR THE AC, THE OTHERS DO),

9,5 CHECK SINGLE LINE SKIP INSTRUCTION AND LOGIC TEST (SA=0600)

THIS TEST CHECKS THE SINGLE LINE FLAG LOGIC AND SKIP INSTRUCTION, IN PARTICULAR IT TESTS: SINGLE LINE FLAG CLEAR AFTER A "CAF"; SINGLE LINE FLAG SETS; SDSS DOES NOT CLEAR SINGLE LINE FLAG; CAF CLEARS SINGLE LINE FLAG; SOLD CLEARS SINGLE LINE FLAG; SDRG CLEARS SINGLE LINE FLAG; SDRD CLEARS SINGLE LINE FLAG; SDST, SDSQ, AND SOLC DOES NOT CLEAR SINGLE LINE FLAG.

9,6 CHECK QUAD LINE SKIP INSTRUCTION AND LOGIC TEST (SA=1024)

THIS TEST CHECKS THE QUAD LINE FLAG LOGIC AND SKIP INSTRUCTION; IN PARTICULAR IT TESTS: QUAD LINE FLAG CLEAR AFTER A "CAF"; QUAD LINE FLAG SETS AT PROPER TIME; SDSQ DOES NOT CLEAR QUAD LINE FLAG; CAF CLEARS QUAD LINE FLAG; SOLD CLEARS QUAD LINE FLAG; SDRG CLEARS QUAD LINE FLAG; SDRD CLEARS QUAD LINE FLAG; SDST, SDSQ, AND SOLC DOES NOT CLEAR QUAD LINE FLAG; ALL QUAD LINE FLAG COUNTER FLIP/FLOPS GET CLEARED (BY SOLD);

9,7 CHECK TIMING ERROR SKIP INSTRUCTION AND LOGIC TEST (SA=1315)

THIS TEST CHECKS THE TIMING ERROR LOGIC AND SKIP INSTRUCTION; IN PARTICULAR IT TESTS: TIMING ERROR CLEAR AFTER A "CAF"; TIMING ERROR SETS IN READ MODE (SDSQ SKIPS); SDST DOES NOT CLEAR TIMING ERROR; CAF CLEARS TIMING ERROR; TIMING ERROR STATUS BIT CAN BE READ INTO AC BY SDRG; SOLC CLEARS TIMING ERROR; TIMING ERROR SETS IN WRITE MODE (PERFORMED AT REVERSE ENDZONE AT BEGINNING OF TAPE); TIMING ERROR STATUS CLEARS "WRITE"; SDRG SDRD SOLD ISSUED AT THE WRONG TIME SETS TIMING ERROR.

9,8 CHECK UP TO SPEED CIRCUITRY TEST (SA=1600)

THIS TEST CHECKS THE UP-TO-SPEED CIRCUITRY TO FUNCTION PROPERLY WHEN CERTAIN COMMANDS ARE GIVEN TO THE DECTAPE CONTROL, THE CHECK IS PERFORMED VIA THE FEATURE OF THE UP-TO-SPEED CIRCUITRY CLEARING THE MARK TRACK REGISTER WHEN THE UP-TO-SPEED DELAY STARTS TIMING OUT, THE COMMANDS ISSUED ARE: STOP TO GO; GO TO STOP; REVERSE TO FORWARD; FORWARD TO REVERSE; UNIT 0 TO UNIT 1; UNIT 1 TO UNIT 0 (ONLY IF UNIT 1 EXISTS - SR11 SET TO 1)

9,9 ROUTINE TO SEARCH AND FIND ALL BLOCK NUMBERS (SA=2100)

THIS ROUTINE RUNS TAPE FROM ENDZONE TO ENDZONE COMPARING ALL
 BLOCK NUMBERS,

9,10 DISPLAY BLOCK NUMBER ROUTINE (SA=2200)

THIS ROUTINE RUNS TAPE FROM ENDZONE TO ENDZONE DISPLAYING
 THE CURRENT BLOCK NUMBER IN THE AC,

9,11 ROUTINE TO ROCK DECTAPE 0 (SA=2237)

THIS ROUTINE ROCKS DECTAPE 0 FOR A DISTANCE DETERMINED
 BY THE CONTENTS OF THE SWITCH REGISTER; THIS ROUTINE
 CAN BE USED TO CHECK "UP TO SPEED", SINGLE LINE FLAG,
 AND QUAD LINE FLAG LOGIC,

9,12 ROUTINE TO RUN DECTAPE FROM ENDZONE TO ENDZONE AND CHECK

THE MARK TRACK IN BLOCKS (SA=2400)

THIS ROUTINE RUNS DECTAPE 0 FROM ENDZONE TO ENDZONE AND
 CHECKS THE CONTENTS OF THE MARK TRACK ON TAPE IN THE
 BLOCKS ON TAPE,

9,13 CHECK SELECT ERROR CIRCUITRY TEST (SA=2500, 2600)

THIS TEST CHECKS THE "SELECT ERROR" CIRCUITRY OF THE
 TUBE CONTROL UNIT 0 IS "ON-LINE", UNIT 1 IS "OFF-LINE"
 OR NO-EXISTANT; FUNCTIONS CHECKED ARE: "SELECT ERROR"
 STATUS FROM UNIT 1; "SELECT ERROR" PREVENTING "WRITE"
 FROM SETTING; NO "SELECT ERROR" FROM UNIT 0,

9,14 CHECK WRITE LOCK OUT CIRCUITRY TEST (SA=2673)

THIS TEST CHECKS THE "WRITE LOCK OUT" CIRCUITRY OF THE
 TUBE CONTROL, UNIT 0 IS "ON-LINE" AND "WRITE LOCKED",
 FUNCTIONS CHECKED ARE: "WRITE-LOCK" STATUS FROM UNIT 0;
 WRITE LOCK STATUS PREVENTING "WRITE FROM SETTING,

"OK" IS PRINTED ON THE TELEPRINTER AFTER THE TWO TESTS
 DESCRIBED IN 9,14 AND 9,13 ABOVE ARE COMPLETED,

9.19

DATA TRANSFER TEST (SA#3000)

DATA TRANSFER TESTS IS A SERIES OF ROUTINES WHICH CHECK THE READ • WRITE • SEARCH CAPABILITIES OF THE TDC CONTROL; EIGHT BASIC DATA PATTERNS ARE USED FOR DATA TRANSFER, THESE ARE: A BUFFER FULL OF 0'S; A BUFFER FULL OF =1'S; A BUFFER FULL OF 2525'S; A BUFFER FULL OF THE DATA PATTERN 2225, 5522, 2555, REPEATED; A BUFFER FULL OF INCREMENT BY 1 DATA PATTERN; A BUFFER FULL OF DECREMENT BY 1 DATA PATTERN; A BUFFER FULL OF 6161'S; A BUFFER FULL OF 3434'S;

DATA TRANSFERS ARE PERFORMED IN BOTH THE FORWARD AND REVERSE DIRECTION, DATA IS WRITTEN IN THE FORWARD DIRECTION, FIRST INTO BLOCK 0, THE SEQUENCE OF OPERATIONS IS: WRITE DATA IN THE FORWARD DIRECTION; READ DATA IN THE FORWARD DIRECTION, CHECK CHECKSUM AND DATA; READ DATA IN THE REVERSE DIRECTION, CHECK CHECKSUM ONLY, THIS SEQUENCE IS REPEATED EVERY 100 BLOCKS (BLOCK 0, 100, 200, 300, ETC) UP TO AND INCLUDING BLOCK 2700; IF IT IS DESIRED TO GO A DIFFERENT NUMBER OF BLOCKS FORWARD CHANGE LOCATION 3154 TO THE DESIRED NUMBER OF BLOCKS;

DATA IS THEN WRITTEN IN THE REVERSE DIRECTION, FIRST INTO BLOCK 2701, THE SEQUENCE OF OPERATIONS IS: WRITE DATA IN THE REVERSE DIRECTION; READ DATA IN REVERSE DIRECTION, CHECK CHECKSUM AND DATA; READ DATA IN THE FORWARD DIRECTION, CHECK CHECKSUM ONLY; THIS SEQUENCE IS REPEATED EVERY 100 BLOCKS (BLOCK 2701, 2801, 2901, 2401, ETC) DOWN TO AND INCLUDING BLOCK 1, IF IT IS DESIRED TO GO A DIFFERENT NUMBER OF BLOCKS REVERSE CHANGE LOCATION 3146 TO THE 2'S COMPLEMENT OF THE DESIRED NUMBER OF BLOCKS;

AFTER UNIT 0 HAS BEEN COMPLETELY TRAVERSED ONCE (FORWARD AND BACKWARD), UNIT 1 WILL BE RUN, IF AVAILABLE, THE PROGRAM WILL THEN PROCEED TO THE NEXT DATA PATTERN AND UNIT 0 AGAIN, AFTER ALL 8 DATA PATTERNS HAVE BEEN EXERCISED ON BOTH UNITS, THE PROGRAM WILL PRINT "PASS 'N' COMPLETE" ON THE TELEPRINTER, THEN PROCEED BACK TO THE FIRST DATA PATTERN,

/TD8E DIAGNOSTIC

/
/COPYRIGHT 1971
/DIGITAL EQUIPMENT CORP.
/MAYNARD, MASS;

/DECTAPE COMMANDS

6771	SDSS=6771	/SKIP ON SINGLE LINE FLAG
6772	SDST=6772	/SKIP ON TIMING ERROR
6773	SDSQ=6773	/SKIP ON QUADRUPLE LINE FLAG
6774	SDLC=6774	/LOAD COMMAND REGISTER
6775	SDLD=6775	/LOAD DATA REGISTER, CLEAR FLAGS
6776	SDRC=6776	/READ COMMAND REGISTER AND MARK TRACK, CLEAR FLAGS
6777	SDRD=6777	/READ DATA REGISTER, CLEAR FLAGS

6807	/NEW PDP-8E INSTRUCTIONS CAF=6807	/CLEAR ALL FLAGS (GENERATE INITIALIZE)
------	--------------------------------------	--

0017 *17

0017	0000	AUTO, 0
0020	0000	OUT, 0
0021	0000	IN, 0
0022	0000	CNTR1, 0
0023	0000	CNTR2, 0
0024	0000	GOOD, 0
0025	0000	HEAD1, 0
0026	0000	HEAD2, 0
0027	0000	BLK, 0
0030	0000	FILPNT, 0
		/SWITCH OPTIONS:
		/SR0(1): LOOP ON CURRENT SUBTEST
		/SR1(1): LOOP ON CURRENT TEST
		/SR2(1): LOOP ON CONTROL TESTS
		/SR3(1): DON'T PRINT ERRORS
		/SR4(1): DON'T HALT ON ERRORS
		/SR11(1): SINGLE UNIT TRANSPORT (S-ONLY)

0031	0000	TYPE, 0
0032	0046	TLS
0033	0041	TSP
0034	5033	JMP ,=I
0035	0042	TCF
0036	7200	CLA
0037	5431	JMP I TYPE
0040	0000	CRLF, 0
0041	1177	TAD (215
0042	4031	JMS TYPE
0043	1176	TAD (212
0044	4031	JMS TYPE
0045	5440	JMP I CRLF
0046	0000	LOOP1, 0
0047	7604	LAS

0050 7004
0051 7700
0052 2046
0053 5446

RAL
SMA CLA
ISZ LOOP1
JMP I LOOP1

0054 7770
0055 0000
0056 0000
0057 0000
0060 0000

M10, =10
BLKTRY, 0
DISBL, 0
DISDA, 0
BLKCN, 0

0061 0000
0062 1175
0063 1774
0064 6774
0065 4773
0066 4773
0067 5461

BLKREV, 0
TAD (3000
TAD UNIT
IOT172, SDLC
JMS RDQUAD
JMS RDQUAD
JMP I BLKREV

0070 0000
0071 6771
0072 5071
0073 6776
0074 0172
0075 1171
0076 7640
0077 5071
0100 5470

BLKEND, 0
IOT173, SDSS
JMP =I
IOT174, SDRC
AND (77
TAD (=22
SZA CLA
JMP =0
JMP I BLKEND

0101 0000
0102 6771
0103 5102
0104 6777
0105 3057
0106 6776
0107 0172
0110 1170
0111 7640
0112 5102
0113 5501

BLKSER, 0
IOT175, SDSS
JMP =I
IOT176, SDRD
DCA DISDA
IOT177, SDRD
AND (77
TAD (=26
SZA CLA
JMP =I0
JMP I BLKSER

0200 0200
0200 5777

PAGE
JMP SELECT /GO TO OPERATOR INTERVENTION TESTS FIRST
/ROUTINE TO CHECK THE LOADING AND READING OF THE DATA REGISTER

0201 7300
0202 3020
0203 1376
0204 3025
0205 1020
0206 7040
0207 6775
0210 7200
0211 1020

DATREG, CLA CLL
DCA OUF /START WITH 0
TAD (MESS1
DCA HEAD1
TAD OUF
CMA
IOT1, SDRD /LOAD DATA REGISTER WITH
CLA /COMPLEMENT OF DATA
TAD OUF

0212	6775	10T2,	SOLO		/LOAD DATA REGISTER WITH DATA
0213	7200		CLA		
0214	6777	10T3,	SORD		/READ DATA REGISTER
0215	3021		DCA	IN	
0216	7604		LAS		
0217	7710		SPA	CLA	/LOOP?
0220	5210		JMP	DATREG+7	/YES
0221	1021		TAD	IN	/COMPARE DATA IN
0222	7041		CIA		
0223	1020		TAD	OUT	/WITH DATA SENT OUT
0224	7650		SNA	CLA	/EQUAL?
0225	5232		JMP	DATLUP	/YES
0226	4775		JMS	ERROR1	
0227	7604		LAS		
0230	7710		SPA	CLA	/LOOP?
0231	5210		JMP	DATREG+7	/YES
0232	2020	DATLUP,	ISE	OUT	/INCREMENT NUMBER TO BE SENT
0233	5205		JMP	DATREG+4	/GO BACK TO DO NEXT NUMBER
0234	4046		JMS	LOOP1	
0235	5201		JMP	DATREG	

/ROUTINE TO CHECK THE LOADING AND READING OF THE COMMAND REGISTER

0236	7300	COMREG,	CLA	CLL	
0237	3020		DCA	OUT	/START WITH 0
0240	1374		TAD	(MESS2	
0241	3025		DCA	HEAD1	
0242	1020		TAD	OUT	
0243	0373		AND	(6400	
0244	6774	10T4,	SOLO		/LOAD COMMAND REGISTER WITH DATA
0245	7200		CLA		
0246	6776	10T5,	SORD		/READ COMMAND REGISTER
0247	0372		AND	(7400	/MASK TO C'R, BITS
0250	3021		DCA	IN	/AND STORE
0251	7604		LAS		
0252	7710		SPA	CLA	/LOOP
0253	5242		JMP	COMREG+4	/YES
0254	1020		TAD	OUT	/GET GOOD WORD
0255	0373		AND	(6400	/MASK OUT 80 BIT
0256	7041		CIA		
0257	1021		TAD	IN	/COMPARE IT WITH WORD IN
0260	7650		SNA	CLA	/BITS OK?
0261	5206		JMP	CLOOP	/YES
0262	4775		JMS	ERROR1	
0263	7604		LAS		
0264	7710		SPA	CLA	
0265	5242		JMP	COMREG+4	
0266	1020	CLOOP,	TAD	OUT	
0267	1371		TAD	(400	
0270	7450		SNA		
0271	5303		JMP	INITST=2	
0272	3020		DCA	OUT	
0273	7604		LAS		
0274	7010		RAR		/MOVE SINGLE UNIT BIT INTO LINK

0275	7620	SNL	CLA		/SINGLE UNIT
0276	5242	JMP		COMREG+4	/NO
0277	7010	RAR			
0300	1020	TAD		OUT	/YES, WORKING
0301	7640	SEA	CLA		/ON 2ND UNIT?
0302	5242	JMP		COMREG+4	/NO
0303	4046	JMS		LOOP1	
0304	5236	JMP		COMREG	
0305	7300	INITST,	CLA	CLL	/TEST INIT TO CLEAR CR
0306	1370	TAD		(MESS3	
0307	3025	DCA		HEAD1	
0310	1367	TAD		(MESS4	
0311	3026	DCA		HEAD2	
0312	1373	TAD		(6400	
0313	6774	IOT6,	SDLC		/LOAD CR WITH 74
0314	6007	CAF			/CLEAR CR
0315	7604	LAS			
0316	7710	SPA	CLA		/LOOP?
0317	5305	JMP		INITST	/YES
0320	6776	IOT7,	SORC		/READ CR
0321	0372	AND		(7400	
0322	7650	SNA	CLA		/CR BITS 0?
0323	5330	JMP		,+5	/YES, OK
0324	4766	JMS		ERROR2	/NO, ERROR, INIT (CAF) DID NOT CLEAR CR
0325	7604	LAS			
0326	7710	SPA	CLA		/LOOP?
0327	5305	JMP		INITST	/YES
0330	4046	JMS		LOOP1	
0331	5305	JMP		INITST	
0332	5771	JMP		CHKCLA	
0366	0537				
0367	5054				
0370	5044				
0371	0400				
0372	7400				
0373	6400				
0374	5021				
0375	0476				
0376	5000				
0377	2600				
	0400				

PAGE

/CHECK SDLC, SLDL, SORC, SDRD TO CLEAR AC AT PROPER TIME (OR NOT AT ALL)

0400	7300	CHKCLA,	CLA	CLL	
0401	1377	TAD		(MESS5	
0402	3025	DCA		HEAD1	
0403	1376	TAD		(MESS6	
0404	3026	DCA		HEAD2	
0405	1167	CSOCL,	TAD	(6777	/SET AC TO 6777
0406	6774	IOT8,	SDLC		
0407	7650	SNA	CLA		/DID SDLC CLEAR AC (AC SHOULD CLEAR)?
0410	5215	JMP		,+5	/YF

0411	7604		LAS		/NO,ERROR
0412	7710		SPA	CLA	/LOOP?
0413	5205		JMP	CSDLC	/NO
0414	4337		JMS	ERROR2	/ERROR
0415	7604		LAS		
0416	7710		SPA	CLA	/LOOP?
0417	5205		JMP	CSDLC	/YES
0420	1375		TAD	(MESS7	
0421	3026		DCA	HEAD2	
0422	7300	CSDRC,	CLA	CLL	
0423	6774	IOT9,	SDLC		/LOAD COMMAND REGISTER WITH 0
0424	7240		CLA	CMA	/SET AC TO ALL 1'S
0425	6776	IOT10,	SORC		/READ COMMAND REGISTER
0426	7650		SNA	CLA	/ALL ZERO'S (AC SHOULD CLEAR BEFORE READING)?
0427	5234		JMP	,+3	/YES
0430	7604		LAS		/NO,ERROR
0431	7710		SPA	CLA	/LOOP?
0432	5222		JMP	CSDRC	/YES
0433	4337		JMS	ERROR2	/ERROR
0434	7604		LAS		
0435	7710		SPA	CLA	/LOOP?
0436	5222		JMP	CSDRC	/YES
0437	1374		TAD	(MESS8	
0440	3026		DCA	HEAD2	
0441	7240	CSDLD,	CLA	CMA	/SET AC TO ALL 1'S
0442	6775	IOT11,	SDLD		/LOAD DATA REGISTER
0443	7640		SZA	CLA	/DID SOLD CLEAR AC (AC SHOULD NOT CLEAR)?
0444	5251		JMP	,+5	/NO, ALL OK
0445	7604		LAS		/YES,ERROR
0446	7710		SPA	CLA	/LOOP?
0447	5241		JMP	CSDLD	/YES
0450	4337		JMS	ERROR2	/ERROR
0451	7604		LAS		
0452	7710		SPA	CLA	/LOOP?
0453	5241		JMP	CSDLD	/YES
0454	1373		TAD	(MESS9	
0455	3026		DCA	HEAD2	
0456	7300	CSDRD,	CLA	CLL	
0457	6775	IOT12,	SDLD		/LOAD REGISTER WITH 0
0460	7240		CLA	CMA	/SET AC TO ALL 1'S
0461	6777	IOT13,	SDRD		/READ DATA REGISTER
0462	7650		SNA	CLA	/ALL ZEROS (AC SHOULD CLEAR BEFORE READING)?
0463	5270		JMP	,+3	/YES
0464	7604		LAS		/NO,ERROR
0465	7710		SPA	CLA	/LOOP?
0466	5256		JMP	CSDRD	/YES
0467	4337		JMS	ERROR2	/ERROR
0470	7604		LAS		
0471	7710		SPA	CLA	/LOOP?
0472	5256		JMP	CSDRD	/YES
0473	4046		JMS	LOOP1	
0474	5200		JMP	CHKCLA	
0475	5772		JMP	SINGLE	

/ERROR HANDLER ROUTINE=DATA WORD TYPEOUTS

```

0476 0000 ERROR1, 0
0477 7604 LAS /GET SR
0500 0371 AND (400 /MASK TO TYPEOUT BIT
0501 7640 SZA CLA /TYPE OUT ERROR?
0502 5325 JMP ERR1HT /NO
0503 4040 JMS CRLF /YES
0504 1025 TAD HEAD1
0505 7450 SNA /TYPE OUT HEADER?
0506 5315 JMP ,+7 /NO
0507 4770' JMS MESSAGE /YES, PRINT HEADER
0510 3025 DCA HEAD1
0511 4040 JMS CRLF /CR=LF
0512 1367 TAD (FORMT1 /PRINT REST OF FORMAT
0513 4770' JMS MESSAGE
0514 4040 JMS CRLF /CR=LF
0515 1020 TAD OUT /PRINT GOOD DATA
0516 0366 AND (6400
0517 4765' JMS OPRINT
0520 1364 TAD (240 /SPACE
0521 4031 JMS TYPE
0522 1021 TAD IN /PRINT BAD DATA
0523 4765' JMS OPRINT
0524 4040 JMS CRLF /CRLF
0525 7604 ERR1HT, LAS /GET SR
0526 0363 AND (200 /MASK TO HALT BIT
0527 7650 SNA CLA /STOP?
0530 7402 E1HLT, HLT /NO
0531 5676 JMP ; ERROR1 /EXIT

0532 0717 FORMT1, TEXT "GOOD BAD"
0533 1704
0534 4002
0535 0104
0536 0000
    
```

/ERROR HANDLER = NO DATA WORD TYPEOUTS

```

0537 0000 ERROR2, 0
0540 7604 LAS /GET SR
0541 0371 AND (400 /MASK TO TYPEOUT BIT
0542 7640 SZA CLA /TYPE OUT ERROR?
0543 5356 JMP ERR2HT /NO
0544 4040 JMS CRLF /YES
0545 1025 TAD HEAD1
0546 7450 SNA /TYPE OUT HEADER
0547 5353 JMP ,+4 /YES
0550 4770' JMS MESSAGE
0551 3025 DCA HEAD1
0552 4040 JMS CRLF
0553 1026 TAD HEAD2 /TYPE OUT ERROR MESSAGE
0554 4770' JMS MESSAGE
0555 4040 JMS CRLF
0556 7604 ERR2HT, LAS /GET SR
    
```

0557 0363
 0560 7650
 0561 7402
 0562 5737

 0563 0200
 0564 0240
 0565 2316
 0566 6400
 0567 0532
 0570 2264
 0571 0400
 0572 0600
 0573 5161
 0574 5151
 0575 5136
 0576 5123
 0577 5076
 0600

AND (200
 SNA CLA
 HLT
 JMP I ERROR2

/MASK TO HALT BIT
 /STOP?
 /YES

PAGE

/CHECK SINGLE LINE SKIP INSTRUCTION AND LOGIC

0600 7300
 0601 1377
 0602 3025
 0603 1376
 0604 3026
 0605 6774
 0606 6771
 0607 7410
 0610 4775
 0611 1374
 0612 6774
 0613 1373
 0614 3026

 0615 7200
 0616 1372
 0617 3022
 0620 3023
 0621 6771
 0622 7410
 0623 5234
 0624 2023
 0625 5221
 0626 2022
 0627 5221
 0630 4775
 0631 7604
 0632 7710
 0633 5200
 0634 1371
 0635 3026

SINGLE, CLA CLL
 TAD (MESS10
 DCA HEAD1
 TAD (MESS11
 DCA HEAD2

 IOT14, SDSS
 SKP
 JMS ERROR2
 TAD (1000

 IOT15, SDLC
 TAD (MESS12
 DCA HEAD2

 SINGLE, CLA
 TAD (=1000
 DCA CNTR1
 DCA CNTR2
 IOT16, SDSS
 SKP
 JMP SINGLE
 ISE CNTR2
 JMP ,=4
 ISE CNTR1
 JMP ,=0
 JMS ERROR2
 LAS
 SPA CLA
 JMP SINGLE
 TAD (MESS13
 DCA HEAD2

/CLEAR ALL FLAGS INITIALLY
 /SKIP ON SINGLE LINE

 /ERROR, SDSS SHOULD NOT HAVE SKIPPED

 /LOAD COMMAND REGISTER WITH U0,FWD,GO,READ

 /SET UP FOR
 /A DELAY
 /OF ABOUT 1 SECOND
 /SINGLE LINE FLAG UP YET?
 /NO
 /YES
 /NO, COUNT

 /DELAY OVER?
 /NO
 /YES, NO SINGLE LINE FLAG, OR SDSS DOES NOT SKIP

 /LOOP?
 /YES

0636	6771	SING1,	SDSS		/FLAG STILL UP?
0637	4775'		JMS	ERROR2	/SINGLE LINE FLAG CLEARED BY SDSS
0640	7604		LAS		
0641	7710		SPA CLA		/LOOP?
0642	5236		JMP	SING1	/YES
0643	1370		TAD	(MESS14	
0644	3026		DCA	HEAD2	
0645	6771	SING2,	SDSS		/WAIT FOR SINGLE LINE FLAG
0646	5245		JMP	=I	
0647	6007		CAF		/CLEAR FLAG WITH CAF
0650	7604		LAS		
0651	7710		SPA CLA		/LOOP?
0652	5245		JMP	SING2	/YES
0653	6771	IOT17,	SDSS		/DID FLAG CLEAR?
0654	5261		JMP	SING3=4	/YES
0655	4775'		JMS	ERROR2	/NO, SINGLE LINE FLAG NOT CLEARED BY CAF
0656	7604		LAS		
0657	7710		SPA CLA		/LOOP?
0660	5245		JMP	SING2	/YES
0661	1374		TAD	(1000	
0662	6774	IOT18,	SDLC		/LOAD COMMAND REGISTER AGAIN
0663	1367		TAD	(MESS15	
0664	3026		DCA	HEAD2	
0665	6771	SING3,	SDSS		/WAIT FOR SINGLE LINE FLAG
0666	5265		JMP	=I	
0667	7200		CLA		
0670	6775	IOT19,	SDLD		/ISSUE SDLD TO CLEAR SINGLE LINE FLAG
0671	7604		LAS		
0672	7710		SPA CLA		/LOOP?
0673	5265		JMP	SING3	/YES
0674	6771	IOT20,	SDSS		/FLAG STILL UP?
0675	5302		JMP	SING4=2	/NO
0676	4775'		JMS	ERROR2	/YES, ERROR, SINGLE LINE FLAG NOT CLEARED BY SDLD;
0677	7604		LAS		
0700	7710		SPA CLA		
0701	5265		JMP	SING3	
0702	1366		TAD	(MESS16	
0703	3026		DCA	HEAD2	
0704	6771	SING4,	SDSS		/WAIT FOR SINGLE LINE FLAG
0705	5304		JMP	=I	
0706	7200		CLA		
0707	6776	IOT21,	SDRC		/ISSUE SDRC TO CLEAR SINGLE LINE FLAG
0710	7604		LAS		
0711	7710		SPA CLA		/LOOP?
0712	5304		JMP	SING4	/YES
0713	6771	IOT22,	SDSS		/FLAG CLEARED?
0714	5321		JMP	SING5=2	/YES
0715	4775'		JMS	ERROR2	/NO, ERROR, SINGLE LINE FLAG NOT CLEARED BY SDRC
0716	7604		LAS		
0717	7710		SPA CLA		/LOOP?
0720	5304		JMP	SING4	/YES
0721	1365		TAD	(MESS17	
0722	3026		DCA	HEAD2	

/T08E DIAGNOSTIC

PAL10 V141 19-007-72

11109 PAGE 1-8

0723	6771	SING5,	SDSS		/WAIT FOR SINGLE LINE FLAG
0724	5323		JMP	,=1	
0725	7200		CLA		
0726	6777	IOT23,	SDRD		/ISSUE SDRD TO CLEAR SINGLE LINE FLAG
0727	7604		LAS		
0730	7710		SPA CLA		/LOOP?
0731	5323		JMP	SING5	/YES
0732	6771	IOT24,	SDSS		/FLAG CLEARED?
0733	5774		JMP	SING6=2	/YES
0734	4775		JMS	ERROR2	/NO, ERROR, SINGLE LINE FLAG NOT CLEARED BY SDRD
0735	7604		LAS		
0736	7710		SPA CLA		/LOOP?
0737	5323		JMP	SING5	/YES
0740	5774		JMP	SING6=2	
0765	5416				
0766	5373				
0767	5350				
0770	5326				
0771	5305				
0772	7000				
0773	5253				
0774	1000				
0775	0537				
0776	5222				
0777	5174				
	1000				

PAGE

1000	1377		TAD	(MESS18	
1001	3026		DCA	HEAD2	
1002	6771	SING6,	SDSS		/WAIT FOR SINGLE LINE FLAG
1003	5202		JMP	,=1	
1004	6772	IOT25,	SDST		/ISSUE SDST
1005	7000		NOF		
1006	6773	IOT26,	SDSQ		/SDSQ
1007	7200		CLA		
1010	1376		TAD	(1000	
1011	6774	IOT27,	SDLC		/AND SDLC
1012	6771	IOT28,	SDSS		/DID SDST, SDSQ, OR SDLC CLEAR FLAG?
1013	7410		SKP		/YES
1014	5221		JMP	SING7	/NO
1015	4775		JMS	ERROR2	/ERROR, SDST, SDSQ, OR SDLC CLEARED SINGLE LINE FLAG
1016	7604		LAS		
1017	7710		SPA CLA		/LOOP?
1020	5202		JMP	SING6	/YES
1021	6774	SING7,	SDLC		
1022	4046		JMS	LOOP1	
1023	5774		JMP	SINGLE	

/CHECK QUAD LINE SKIP INSTRUCTION AND LOGIC

1024	7300	QUAD,	CLA CLL	
1025	1373		TAD	(MESS19

1026	3025		DCA	HEAD1	
1027	1372		TAD	(MESS20	
1030	3026		DCA	HEAD2	
1031	6774		SDLC		/CLEAR ALL FLAGS INITIALLY
1032	6773	IOT29,	SDSQ		/SKIP ON QUAD LINE
1033	7410		SKP		
1034	4775'		JMS	ERROR2	/ERROR, SDSQ SHOULD NOT HAVE SKIPPED
1035	1376		TAD	(1000	
1036	6774	IOT30,	SDLC		/LOAD COMMAND REGISTER WITH U0, FWD, GO, READ
1037	1371		TAD	(MESS21	
1040	3026		DCA	HEAD2	
1041	6777	QUAD0,	SDRD		/CLEAR DANGLING FLAGS
1042	6771	IOT31,	SDSS		/WAIT FOR 1ST
1043	5242		JMP	,=1	/SINGLE LINE FLAG
1044	6773	IOT32,	SDSQ		/QUAD FLAG UP?
1045	5250		JMP	,+3	/NO
1046	4775'		JMS	ERROR2	/YES ERROR
1047	5274		JMP	Q0LUP	
1050	6771	IOT33,	SDSS		/WAIT FOR SINGLE
1051	7410		SKP		/LINE FLAG
1052	5250		JMP	,=2	/TO GO AWAY
1053	6773	IOT34,	SDSQ		/QUAD FLAG UP?
1054	5257		JMP	,+3	/NO
1055	4775'		JMS	ERROR2	/YES, ERROR
1056	5274		JMP	Q0LUP	
1057	6771	IOT35,	SDSS		/WAIT FOR NEXT
1060	5257		JMP	,=1	/SINGLE LINE FLAG
1061	6773	IOT36,	SDSQ		/QUAD FLAG UP?
1062	5265		JMP	,+3	/NO
1063	4775'		JMS	ERROR2	/YES, ERROR
1064	5274		JMP	Q0LUP	
1065	1370		TAD	(MESS22	
1066	3026		DCA	HEAD2	
1067	6771	IOT37,	SDSS		/WAIT FOR SINGLE
1070	7410		SKP		/LINE FLAG
1071	5267		JMP	,=2	/TO GO AWAY
1072	6773	IOT38,	SDSQ		/QUAD FLAG UP?
1073	4775'		JMS	ERROR2	/NO, ERROR
1074	7604	Q0LUP,	LAS		
1075	7710		SPA	CLA	
1076	5235		JMP	QUAD0=4	
1077	1367		TAD	(MESS23	
1100	3026		DCA	HEAD2	
1101	6773	QUAD1,	SDSQ		/FLAG STILL UP?
1102	4775'		JMS	ERROR2	/QUAD LINE FLAG CLEARED BY SDSQ
1103	7604		LAS		
1104	7710		SPA	CLA	/LOOP?
1105	5301		JMP	QUAD1	/YES
1106	1366		TAD	(MESS24	
1107	3026		DCA	HEAD2	
1110	6773	QUAD2,	SDSQ		/WAIT FOR QUAD FLAG
1111	5310		JMP	,=1	
1112	6007		CAF		/CLEAR THE FLAG WITH CAF
1113	7604		LAS		

1114	7710		SPA CLA	/LOOP?
1115	5310		JMP QUAD2	/YES
1116	6773	IOT39,	SQSQ	/DID FLAG CLEAR?
1117	5324		JMP QUAD3=4	/YES
1120	4775'		JMS ERROR2	/NO, QUAD LINE FLAG NOT CLEARED BY CAF
1121	7604		LAS	
1122	7710		SPA CLA	/LOOP?
1123	5310		JMP QUAD2	/YES
1124	1376		TAD (1000	
1125	6774	IOT40,	SQLC	/LOAD COMMAND REGISTER AGAIN
1126	1365		TAD (MESS25	
1127	3026		DCA HEAD2	
1130	6773	QUAD3,	SQSQ	/WAIT FOR QUAD FLAG
1131	5330		JMP ,=1	
1132	7200		CLA	
1133	6775	IOT41,	SQLD	/ISSUE SQLD TO CLEAR QUAD FLAG
1134	7604		LAS	
1135	7710		SPA CLA	/LOOP?
1136	5330		JMP QUAD3	/YES
1137	6773	IOT42,	SQSQ	/FLAG STILL UP?
1140	5764'		JMP QUAD4=2	/NO
1141	4775'		JMS ERROR2	/YES, ERROR, QUAD FLAG NOT CLEARED BY SQLD
1142	7604		LAS	
1143	7710		SPA CLA	/LOOP?
1144	5330		JMP QUAD3	/YES
1145	5764'		JMP QUAD4=2	
1164	1200			
1165	5656			
1166	5635			
1167	5615			
1170	5564			
1171	5546			
1172	5516			
1173	5471			
1174	0600			
1175	0537			
1176	1000			
1177	5441			
	1200		PAGE	
1200	1377		TAD (MESS26	
1201	3026		DCA HEAD2	
1202	6773	QUAD4,	SQSQ	/WAIT FOR QUAD FLAG
1203	5202		JMP ,=1	
1204	7200		CLA	
1205	6776	IOT43,	SQRC	/ISSUE SQRC TO CLEAR QUAD FLAG
1206	7604		LAS	
1207	7710		SPA CLA	/LOOP?
1210	5202		JMP QUAD4	/YES
1211	6773	IOT44,	SQSQ	/FLAG CLEARED?
1212	5217		JMP QUAD5=2	/YES
1213	4776'		JMS ERROR2	/NO, ERROR, QUAD FLAG NOT CLEARED BY SQRC
1214	7604		LAS	
1215	7710		SPA CLA	/LOOP?

/T08E DIAGNOSTIC

	PAL10	V141	19 OCT 72	11109	PAGE 1=11
1216	5202	JMP	QUAD4	/YES	
1217	1375	TAD	(MESS27		
1220	3026	DCA	HEAD2		
1221	6773	QUAD5,	SQSQ	/WAIT FOR QUAD FLAG	
1222	5221	JMP	=1		
1223	7200	CLA			
1224	6777	IOT45,	SDRD	/ISSUE SDRD TO CLEAR QUAD LINE FLAG	
1225	7604	LAS			
1226	7710	SPA CLA		/LOOP?	
1227	5221	JMP	QUAD5	/YES	
1230	6773	IOT46,	SDSQ	/FLAG CLEARED?	
1231	5236	JMP	QUAD6=2	/YES	
1232	4776	JMS	ERROR2	/NO, ERROR, QUAD FLAG NOT CLEARED BY SDRD	
1233	7604	LAS			
1234	7710	SPA CLA		/LOOP?	
1235	5221	JMP	QUAD5	/YES	
1236	1374				
1237	3026	TAD	(MESS28		
1240	6773	QUAD6,	DCA		
1241	5240	SDSQ	HEAD2	/WAIT FOR QUAD FLAG	
1242	6772	JMP	=1		
1243	7000	IOT47,	SDST	/ISSUE SDST	
1244	6771	IOT48,	SDSS	/SDSS	
1245	7000	NOF			
1246	7200	CLA			
1247	1373	TAD	(1000		
1250	6774	IOT49,	SDLC	/AND SDLC	
1251	7604	LAS			
1252	7710	SPA CLA		/LOOP?	
1253	5240	JMP	QUAD6	/YES	
1254	6773	IOT50,	SDSQ	/DID STST, SDSS, OR SDLC CLEAR FLAG?	
1255	7410	SKP		/YES	
1256	5263	JMP	QUAD7=2	/NO	
1257	4776	JMS	ERROR2	/ERROR, SDST, SDSS, OR SDLC CLEARED QUAD FLAG	
1260	7604	LAS			
1261	7710	SPA CLA		/LOOP?	
1262	5240	JMP	QUAD6	/YES	
1263	1372				
1264	3026	TAD	(MESS29		
1265	7300	QUAD7,	DCA		
1266	1371	CLA CLL			
1267	3022	TAD	(=2	/SET LOOP COUNT TO=2	
1270	6775	DCA	CNTR1		
1271	6771	IOT51,	SDLD	/CLEAR QUAD FLAG FLIP/FLOPS	
1272	5271	IOT52,	SDSS	/WAIT FOR SINGLE LINE	
1273	6771	JMP	=1	/TO COME	
1274	7410	IOT53,	SDSS	/GO AWAY	
1275	5274	SKP			
1276	6771	JMP	=1		
1277	5276	IOT54,	SDSS	/AND COME AGAIN	
1300	2022	JMP	=1	/	
1301	5270	ISE	CNTR1	/TWICE THRU?	
1302	7604	JMP	QUAD7=3	/NO	
		LAS		/YES	

1303	7710		SPA CLA	/LOOP?
1304	5265		JMP QUAD7	/YES
1305	6773	10T55,	SDSQ	/NO, IS QUAD FLAG UP?
1306	7410		SKP	/NO
1307	4776		JMS ERROR2	/YES, ERROR QUAD FLAG COUNTER FLIP/FLOPS NOT CLEARED
1310	7604		LAS	/BY SDLO
1311	7710		SPA CLA	/LOOP?
1312	5265		JMP QUAD7	/YES
1313	4046		JMS LOOP1	
1314	5770		JMP QUAD	

/CHECK TIMING ERROR SKIP INSTRUCTION AND LOGIC

1315	7300		TIMING, CLA CLL	
1316	1367		TAD (MESS30	
1317	3025		DCA HEAD1	
1320	1306		TAD (MESS31	
1321	3026		DCA HEAD2	
1322	6774		SDLC	/CLEAR ALL FLAGS INITIALLY
1323	1373		TAD (1000	
1324	6774	10T56,	SDLC	/LOAD COMMAND REGISTER WITH UP,FWD,GO,READ
1325	6771	10T57,	SOSS	/WAIT FOR SINGLE
1326	5325		JMP =I	/LINE FLAG
1327	6772	10T58,	SDST	/SKIP ON TIMING ERROR
1330	7410		SKP	
1331	4776		JMS ERROR2	/ERROR, SDST SHOULD NOT HAVE SKIPPED
1332	1369		TAD (MESS32	
1333	3026		DCA HEAD2	
1334	6773	TIMES,	SDSQ	/WAIT FOR QUAD FLAG
1335	5334		JMP =I	
1336	7200		CLA	
1337	3022		DCA CN9R1	
1340	2022		ISE CN9R1	/WAIT A WHILE SO THAT TIMING ERROR
1341	5340		JMP =I	/CAN SET
1342	6772	10T59,	SDST	/TIMING ERROR SET?
1343	7410		SKP	/NO
1344	5331		JMP TIME1=0	/YES
1345	4776		JMS ERROR2	/ERROR, TIMING ERROR NOT SET IN READ MODE
1346	7604		LAS	
1347	7710		SPA CLA	/LOOP?
1350	5334		JMP TIMES	/YES
1351	1364		TAD (MESS33	
1352	3026		DCA HEAD2	
1353	6772	TIME1,	SDST	/TIMING ERROR STILL SET?
1354	4776		JMS ERROR2	/TIMING ERROR CLEARED BY SDST
1355	7604		LAS	
1356	7710		SPA CLA	/LOOP?
1357	5333		JMP TIME1	/YES
1360	5763		JMP TIME2=2	
1363	1400			
1364	6140			
1365	6101			
1366	6092			
1367	6026			

1370 1024
 1371 7776
 1372 5773
 1373 1000
 1374 5744
 1375 5722
 1376 0537
 1377 5700
 1400

PAGE

1400	1377	TAD	(MESS34	
1401	3026	DCA	HEAD2	
1402	6772	SDST		/WAIT FOR TIMING ERROR
1403	5202	JMP	=I	
1404	6007	CAF		/CLEAR FLAG WITH CAF
1405	7604	LAS		
1406	7710	SPA CLA		/LOOP?
1407	5202	JMP	TIME2	/YES
1410	6772	SDST		/DID FLAG CLEAR?
1411	5216	JMP	TIME3=4	
1412	4776	JMS	ERROR2	/NO, TIMING ERROR NOT CLEARED BY CAF
1413	7604	LAS		
1414	7710	SPA CLA		/LOOP?
1415	5202	JMP	TIME2	/YES
1416	1375	TAD	(1000	
1417	6774	SDLC		/LOAD COMMAND REGISTER AGAIN
1420	1374	TAD	(MESS35	
1421	3026	DCA	HEAD2	
1422	6772	SDST		/WAIT FOR TIMING ERROR
1423	5222	JMP	=I	
1424	6776	SDRC		/READ DECTAPE COMMAND REGISTER FOR STATUS
1425	3021	DCA	IN	/SAVE
1426	7604	LAS		
1427	7710	SPA CLA		/LOOP?
1430	5222	JMP	TIME3	/YES
1431	1021	TAD	IN	/GET STATUS BACK AGAIN
1432	0373	AND	(100	/MASK TO BIT 5
1433	7440	SEA		/TIMING ERROR STATUS SET?
1434	5241	JMP	TIME4=4	/YES, OK
1435	4776	JMS	ERROR2	/NO, ERROR, TIMING ERROR STATUS NOT SET
1436	7604	LAS		
1437	7710	SPA CLA		/LOOP?
1440	5222	JMP	TIME3	/YES
1441	1375	TAD	(1000	
1442	6774	SDLC		/LOAD COMMAND REGISTER AGAIN
1443	1372	TAD	(MESS36	
1444	3026	DCA	HEAD2	
1445	6772	SDST		/WAIT FOR TIMING ERROR
1446	5245	JMP	=I	
1447	6774	SDLC		/CLEAR FLAG WITH SDLC
1400	7604	LAS		
1401	7710	SPA CLA		/LOOP?
1402	5245	JMP	TIME4	/YES

1453	6772	IOT65,	SDST		/DID FLAG CLEAR?
1454	5261		JMP	TIME5=2	/YES
1455	4776'		JMS	ERROR2	/NO, TIMING ERROR NOT CLEARED BY SOLC
1456	7604		LAS		
1457	7710		SPA CLA		/LOOP?
1460	5245		JMP	TIME4	
1461	1371		TAD	(MESS37	
1462	3026		DCA	HEAD2	
1463	7300	TIME5,	CLA CLL		
1464	1370		TAD	(3000	
1465	6774	IOT66,	SOLC		/GET TAPE MOVING BACKWARD
1466	6771	IOT67,	SDSS		/WAIT FOR END ZONE
1467	5266		JMP	=1	
1470	6776	IOT68,	SDRC		
1471	0367		AND	(77	
1472	1366		TAD	(=22	
1473	7640		SZA CLA		
1474	5266		JMP	=6	
1475	6776	IOT69,	SDRC		/SET "WRITE"
1476	0365		AND	(7000	
1477	1364		TAD	(400	
1500	6774	IOT70,	SOLC		
1501	3022		DCA	CNTR1	
1502	2022		ISE	CNTR1	/WAIT A WHILE
1503	5302		JMP	=1	
1504	6772	IOT71,	SDST		/TIMING ERROR?
1505	4776'		JMS	ERROR2	/NO, ERROR
1506	1363		TAD	(MESS38	
1507	3026		DCA	HEAD2	
1510	6776	IOT72,	SDRC		/YES, READ STATUS
1511	0364		AND	(400	
1512	7640		SZA CLA		/"WRITE" CLEARED
1513	4776'		JMS	ERROR2	/NO, ERROR
1514	7604		LAS		
1515	7710		SPA CLA		/LOOP?
1516	5263		JMP	TIME5	/YES
1517	1362		TAD	(MESS39	
1520	3026		DCA	HEAD2	
1521	7300	TIME6,	CLA CLL		
1522	1375		TAD	(1000	/SET UNIT 0 RUNNING FORWARD
1523	6774	IOT73,	SOLC		
1524	6771	IOT74,	SDSS		/WAIT FOR "UP TO SPEED"
1525	5324		JMP	=1	
1526	1361		TAD	(=0	
1527	3023		DCA	CNTR2	
1530	3022		DCA	CNTR1	
1531	6776	IOT75,	SDRC		/ISSUE MANY SDRC, SDRD, SOLC'S
1532	6777	IOT76,	SDRD		
1533	6775	IOT77,	SOLD		
1534	2022		ISE	CNTR1	
1535	5331		JMP	=4	

1536 2023
 1537 5331
 1540 7604
 1541 7710
 1542 5327
 1543 6772
 1544 4776'
 1545 7604
 1546 7710
 1547 5321
 1550 4046
 1551 5760'
 1552 5757'
 1557 1600
 1560 1315
 1561 7773
 1562 6324
 1563 6276
 1564 0400
 1565 7000
 1566 7756
 1567 0077
 1570 3000
 1571 6252
 1572 6231
 1573 0100
 1574 6177
 1575 1000
 1576 0537
 1577 6157
 1600

IOT78,

ISE CNTR2
 JMP ,=6
 LAS
 SPA CLA
 JMP TIME6+6
 SDST
 JMS ERROR2
 LAS
 SPA CLA
 JMP TIME6
 JMS LOOP1
 JMP TIMING
 JMP UTSMRK

/LOOP?
 /YES
 /TIMING ERROR?
 /NO, ERROR
 /LOOP?
 /YES

PAGE

/CHECK UP TO SPEED CIRCUITRY TO CLEAR MARK TRACK WINDOW

1600 7300
 1601 1377
 1602 3025
 1603 1376
 1604 3026
 1605 6774
 1606 1375
 1607 6774
 1610 7604
 1611 7710
 1612 5200
 1613 6776
 1614 0374
 1615 7440
 1616 4773'
 1617 7604
 1620 7710
 1621 5200

UTSMRK, CLA CLL

IOT82,

IOT83,

IOT84,

TAD (MESS43
 DCA HEAD1
 TAD (MESS44
 DCA HEAD2
 SOLC
 TAD (1000
 SOLC
 LAS
 SPA CLA
 JMP UTSMRK
 SORC
 AND (??
 SZA
 JMS ERROR2
 LAS
 SPA CLA
 JMP UTSMRK

/CLEAR STOP/GO BIT
 /SET STOP/GO BIT
 /LOOP?
 /YES
 /READ MARK TRACK
 /ZERO?
 /NO, ERROR
 /LOOP?
 /YES

1622	1372		TAD	(MESS45	
1623	3026		DCA	HEAD2	
1624	7300	UTSMK1,	CLA	CLL	
1625	1375		TAD	(1000	/SET STOP/GO BIT,
1626	6774	IOT85,	SDLC		
1627	6771	IOT86,	SDSS		/SINGLE LINE FLAG?
1630	5227		JMP	,=1	/NO
1631	6776	IOT87,	SDRC		/YES, READ MARK TRACK
1632	0374		AND	(77	
1633	7650		SNA	CLA	/ZERO?
1634	5227		JMP	,=5	/YES, TRY AGAIN
1635	6774	IOT88,	SDLC		/CLEAR STOP/GO BIT
1636	7604		LAS		
1637	7710		SPA	CLA	/LOOP?
1640	5224		JMP	UTSMK1	/YES
1641	6776	IOT89,	SDRC		/READ MARK TRACK
1642	0374		AND	(77	
1643	7440		SZA		/ZERO?
1644	4773		JMS	ERROR2	/NO, ERROR
1645	7604		LAS		
1646	7710		SPA	CLA	/LOOP?
1647	5224		JMP	UTSMK1	/YES
1650	1371		TAD	(MESS46	
1651	3026		DCA	HEAD2	
1652	7300	UTSMK2,	CLA	CLL	
1653	1370		TAD	(3000	/SET STOP/GO AND FWD/REV
1654	6774	IOT90,	SDLC		
1655	6771	IOT91,	SDSS		
1656	5255		JMP	,=1	
1657	6776	IOT92,	SDRC		
1660	0374		AND	(77	
1661	7650		SNA	CLA	
1662	5255		JMP	,=5	
1663	1375		TAD	(1000	/CLEAR FWD/REV (BIT1)
1664	6774	IOT93,	SDLC		
1665	7604		LAS		
1666	7710		SPA	CLA	
1667	5252		JMP	UTSMK2	
1670	6776	IOT94,	SDRC		
1671	0374		AND	(77	
1672	7440		SZA		/MARK TRACK ZERO?
1673	4773		JMS	ERROR2	/NO, ERROR
1674	7604		LAS		
1675	7710		SPA	CLA	
1676	5252		JMP	UTSMK2	
1677	1367		TAD	(MESS47	
1700	3026		DCA	HEAD2	
1701	7300	UTSMK3,	CLA	CLL	
1702	1375		TAD	(1000	/SET STOP/GO, CLEAR FWD/REV (BIT1)
1703	6774	IOT95,	SDLC		
1704	6771	IOT96,	SDSS		
1705	5304		JMP	,=1	
1706	6776	IOT97,	SDRC		

1707 0374
 1710 7650
 1711 5304
 1712 1370
 1713 6774
 1714 7604
 1715 7710
 1716 5301
 1717 6776
 1720 0374
 1721 7440
 1722 4773'
 1723 7604
 1724 7710
 1725 5301
 1726 5766'
 1766 2000
 1767 6517
 1770 3000
 1771 6466
 1772 6441
 1773 0537
 1774 0077
 1775 1000
 1776 6415
 1777 6352
 2000

IOT98,

IOT99,

AND (77
 SNA CLA
 JMP ,=5
 TAD (3000
 SDLC
 LAS
 SPA CLA
 JMP UTSMK3
 SDRC
 AND (77
 SZA
 JMS ERROR2
 LAS
 SPA CLA
 JMP UTSMK3
 JMP UTSMK4#2

/SET FWD/REV (BIT 1)

/MARK TRACK ZERO?
/NO, ERROR

PAGE

2000 1377
 2001 3026
 2002 7300
 2003 1376
 2004 6774
 2005 6771
 2006 5205
 2007 6776
 2010 0375
 2011 7650
 2012 5205
 2013 1374
 2014 6774
 2015 7604
 2016 7710
 2017 5202
 2020 6776
 2021 0375
 2022 7440
 2023 4773'
 2024 7604
 2025 7710
 2026 5202
 2027 7604
 2030 7010
 2031 7630

UTSMK4,

IOT100,

IOT101,

IOT102,

IOT103,

IOT104,

TAD (MESS40
 DCA HEAD2
 CLA CLL
 TAD (1000
 SDLC
 SDSS
 JMP ,=I
 SDRC
 AND (77
 SNA CLA
 JMP ,=5
 TAD (5000
 SDLC
 LAS
 SPA CLA
 JMP UTSMK4
 SDRC
 AND (77
 SZA
 JMS ERROR2
 LAS
 SPA CLA
 JMP UTSMK4
 LAS
 RAR
 SEL CLA

/SET STOP/GO, CLEAR UNIT (BIT0)

/SET UNIT (BIT0)

/MARK TRACK 0?
/NO

/IS THERE A UNIT1?

2032 5262

JMP UTSMK6 /NO

2033 1372

TAD (MESS49

2034 3026

DCA HEAD2

2035 7300

UTSMK5, CLA CLL

2036 1374

TAD (5000

/SET STOP/60, UNIT (BIT0)

2037 6774

IOT105, SOLC

2040 6771

IOT106, SDSS

2041 5240

JMP ,=1

2042 6776

IOT107, SDRG

2043 0375

AND (77

2044 7650

SNA CLA

2045 5240

JMP ,=5

2046 1376

TAD (1000

/CLEAR UNIT (BIT0)

2047 6774

IOT108, SOLC

2050 7604

LAS

2051 7710

SPA CLA

2052 5235

JMP UTSMK5

2053 6776

IOT109, SDRG

2054 0375

AND (77

2055 7440

SEA

/MARK TRACK ZERO?

2056 4773

JMS ERROR2

/NO, ERROR

2057 7604

LAS

2060 7710

SPA CLA

2061 5235

JMP UTSMK5

2062 4046

UTSMK6, JMS

LOOP1

2063 5771

JMP

UTSMRK

2064 1370

TAD

(4000

2065 6774

IOT110, SOLC

/STOP UNIT 1 IF MOVING

2066 7604

LAS

2067 7006

RTL

2070 7710

SPA CLA

2071 5767

JMP DAYREG

2072 5766

JMP XFER

/ROUTINE TO SEARCH AND FIND ALL BLOCK NUMBERS
/THE RIGHT HAND REEL MUST HAVE AT LEAST FOUR TURNS OF TAPE ON IT

2100

*2100

2100 7300

BLKCH, CLA

CLL

2101 3056

DCA

DISBL

2102 1160

TAD

[=2702

2103 3060

DCA

BLKCN

2104 4081

JMS

BLKREV

2105 4070

JMS

BLKEND

2106 1165

TAD

(1000

2107 1765

TAD

UNIT

2110 6774

IOT171, SOLC

2111 4764

JMS

ROGUAD

2112 4764

JMS

ROGUAD

2113 4101

FBLKCT, JMS

BLKSER

2114 1057

TAD

DISDA

2115 7041

CIA

2116 1056

TAD

DISBL

2117 7640

SEA

CLA

2120 5346

JMP

BLKERR

/BLOCKS DIDN'T COMPARE

2121	2056		ISE	DISBL
2122	2060		ISE	BLKCN
2123	5313		JMP	FBLKCT
2124	4070		JMS	BLKEND
2125	1164		TAD	[2701
2126	3056		DCA	DISBL
2127	1166		TAD	[=2702
2130	3060		DCA	BLKCN
2131	4061		JMS	BLKREV
2132	4101	RBLKCT,	JMS	BLKSER
2133	1057		TAD	DISDA
2134	7041		CIA	
2135	1056		TAD	DISBL
2136	7640		SEA	CLA
2137	5346		JMP	BLKERR
2140	7040		CMA	
2141	1056		TAD	DISBL
2142	3056		DCA	DISBL
2143	2060		ISE	BLKCN
2144	5332		JMP	RBLKCT
2145	5300		JMP	BLKCH

2146	7300	BLKERR,	CLA	CLL
2147	1056		TAD	DISBL
2150	7402		HLT	
2151	7200		CLA	
2152	1057		TAD	DISDA
2153	7402		HLT	
2154	5300		JMP	BLKCH

/AC=THE BLOCK NUMBER THAT WAS BEING SEARCHED FOR

/AC=THE BLOCK NUMBER THAT WAS FOUND
/RETURN TO START OF ROUTINE

2164	4707
2165	2234
2166	3000
2167	0201
2170	4000
2171	1600
2172	6604
2173	0537
2174	9000
2175	0077
2176	1000
2177	6550
	2200

PAGE

/TAPE 2
/ROUTINE TO RUN FROM END ZONE TO END ZONE
/AND DISPLAY THE CURRENT BLOCK NUMBER IN THE AC

2200	7300	DBLOCK,	CLA	CLL
2201	3233		DCA	DISBLK
2202	1377		TAD	(3000
2203	1234		TAD	UNIT
2204	6774	IOY111,	SDLC	

/ZERO DISBLK

/LOAD CONTROL WITH UNIT REV GO READ

2205	7300	CLA CLL	
2206	6771	DISLUP, SDSS	/WAIT FOR SINGLE LINE FLAG
2207	5206	JMP	,=1
2210	7300	CLA CLL	
2211	6777	IOT112, SDRD	/READ DATA BUFFER
2212	3236	DCA DISDAT	/AND SAVE
2213	6776	IOT113, SDRD	/READ MARK TRACK
2214	0376	AND	(77
2215	1375	TAD	(=26
2216	7440	SZA	
2217	5224	JMP DISEND	/BLOCK NUMBER?
2220	1236	TAD DISDAT	/NO, CHECK FOR END ZONE
2221	2233	ISE DISBLK	/YES, DISPLAY BLOCK NUMBER
2222	5221	JMP	,=1
2223	5206	JMP DISLUP	
2224	1374	DISEND, TAD	(4
2225	7640	SZA CLA	/END ZONE?
2226	5206	JMP DISLUP	/NO, LOOP
2227	6776	IOT114, SDRD	/YES, EXTRACT DIRECTION BIT
2230	7006	RTL	/AND COMPLEMENT
2231	7032	CML RTR	
2232	5204	JMP DISLUP=2	/GO LOAD INTO CONTROL
2233	0000	DISBLK, 0	
2234	0000	UNIT, 0	
2235	0000	DISTRK, 0	
2236	0000	DISDAT, 0	
		/ROUTINE TO ROCK DECTAPE UNIT 0	
		/FOR A DISTANCE DETERMINED BY ACS	
2237	7300	ROCK: CLA CLL	
2240	1373	TAD	(1000
2241	6774	IOT115, SDLC	
2242	7604	LAS	
2243	7040	CMA	
2244	3001	DCA	1
2245	2000	ISE	0
2246	5245	JMP	,=1
2247	2001	ISE	1
2250	5245	JMP	,=3
2251	7000	NOP	
2252	1377	TAD	(3000
2253	6774	IOT116, SDLC	
2254	7604	LAS	
2255	7040	CMA	
2256	3001	DCA	1
2257	2000	ISE	0
2260	5257	JMP	,=1
2261	2001	ISE	1
2262	5257	JMP	,=3
2263	5240	JMP	ROCK+1
2264	0000	MESSAGE, 0	
2265	3310	DCA MPNTR	
2266	1715	TAD I MPNTR	
2267	7012	RTR	
2270	7012	RTR	

2271	7012	RTR	
2272	0376	AND	(77
2273	7450	SNA	
2274	5664	JMP I	MESSAGE
2275	1372	TAD	(=40
2276	7510	SPA	
2277	1371	TAD	(100
2300	1370	TAD	(240
2301	4031	JMS	TYPE
2302	1715	TAD I	MPNTR
2303	0376	AND	(77
2304	7450	SNA	
2305	5664	JMP I	MESSAGE
2306	1372	TAD	(=40
2307	7510	SPA	
2310	1371	TAD	(100
2311	1370	TAD	(240
2312	4031	JMS	TYPE
2313	2313	ISE	MPNTR
2314	5266	JMP	MESSAGE+2
2315	0000	MPNTR,	0
2316	0000	OPRINT,	0
2317	3340	DCA	ONUMB
2320	1367	TAD	(=4
2321	3341	DCA	OCNT
2322	1340	TAD	ONUMB
2323	7004	RAL	
2324	7004	OPLOOP,	RAL
2325	7006	RTL	
2326	3340	DCA	ONUMB
2327	1340	TAD	ONUMB
2330	0366	AND	(7
2331	1365	TAD	(260
2332	4031	JMS	TYPE
2333	1340	TAD	ONUMB
2334	2341	ISE	OCNT
2335	5324	JMP	OPLOOP
2336	7200	CLA	
2337	5716	JMP I	OPRINT
2340	0000	ONUMB,	0
2341	0000	OCNT,	0
2365	0260		
2366	0007		
2367	7774		
2370	0240		
2371	0100		
2372	7740		
2373	1000		
2374	0004		
2375	7752		
2376	0077		
2377	3000		
	2400		

PAGE

/ROUTINE TO RUN FROM ENDOZONE TO ENDOZONE CHECKING THE MARK TRACK IN BLOCKS

2400	7300	BLKCHK, CLA	CLL		
2401	1377		TAD	(3000	/START TAPE MOVING BACKWARD
2402	6774	IOT117, SDLC			
2403	4315		JMS	RD6MRK	/WAIT FOR WINDOW TO OPEN
2404	4307	ENDZ,	JMS	RD1MRK	/READ BACK MARK TRACK
2405	1376		TAD	(=22	
2406	7640		SEA	CLA	/ENDZONE?
2407	5204		JMP	,=3	/NO
2410	6776	IOT118, SDRQ			/TURN AROUND
2411	7006		RTL		
2412	7032		CML	RTR	
2413	6774	IOT119, SDLC			
2414	4315		JMS	RD6MRK	/WAIT FOR WINDOW TO OPEN
2415	4307		JMS	RD1MRK	/READ MARK TRACK
2416	1375		TAD	(=26	
2417	7650		SNA	CLA	/BLOCK NUMBER?
2420	5236		JMP	RVGARD	/YES, GO CHECK REVERSE GUARD
2421	5215		JMP	,=4	/NO, LOOK AGAIN
2422	4315	FWDEXP, JMS		RD6MRK	/READ MARK TRACK
2423	1374		TAD	(=25	
2424	7440		SEA		/EXPAND CODE?
2425	7402		HLT		/NO, ERROR
2426	4315	BLKMRK, JMS		RD6MRK	/READ MARK TRACK
2427	1375		TAD	(=24	
2430	7450		SNA		/BLOCK NUMBER?
2431	5236		JMP	RVGARD	/YES, GO CHECK REVERSE GUARD
2432	7001		IAC		/NO
2433	7440		SEA		/EXPAND CODE?
2434	7402		HLT		/NO, UNKNOWN
2435	5204		JMP	ENDZ	/YES, EXPAND CODE, GO LOOK FOR ENDZONE
2436	4315	RVGARD, JMS		RD6MRK	/GET MARK TRACK
2437	1373		TAD	(=02	
2440	7440		SEA		/REVERSE GUARD?
2441	7402		HLT		/NO, ERROR
2442	1372		TAD	(=4	/SET UP
2443	3000		DCA	0	/FOR 4 MARKS
2444	4315		JMS	RD6MRK	/GET MARK TRACK
2445	1371	LOCK, TAD		(=10	
2446	7440		SEA		/LOCK, REV CHKSM, REV FINAL, REV PRE=FINAL?
2447	7402		HLT		/NO, ERROR
2450	2000		ISE	0	
2451	5244		JMP	,=0	
2452	1370	DATA, TAD		(=122	/SET UP
2453	3000		DCA	0	/FOR 02 MARKS
2454	4315		JMS	RD6MRK	/GET MARK TRACK
2455	1367		TAD	(=70	
2456	7440		SEA		/DATA MARK?
2457	7402		HLT		/NO, ERROR
2460	2000		ISE	0	
2461	5254		JMP	,=0	
2462	1372	PREFIN, TAD		(=4	/SET UP
2463	3000		DCA	0	/FOR 4 MARKS
2464	4315		JMS	RD6MRK	/GET MARK TRACK
2465	1366		TAD	(=93	
2466	7440		SEA		/PREFINAL, FINAL, CHKSM, REVLOCK?

2467	7402	HLT		/NO, ERROR
2470	2000	ISE	0	
2471	5264	JMP	=5	
2472	4315	GUARD,	RD6MRK	/GET MARK TRACK
2473	1365	TAD	(=51	
2474	7440	SEA		/GUARD?
2475	7402	HLT		/NO, ERROR
2476	4315	REVBLK,	RD6MRK	/GET MARK TRACK
2477	1364	TAD	(=45	
2500	7440	SEA		/REVERSE BLOCK NUMBER?
2501	7402	HLT		/NO, ERROR
2502	4315	REVEXP,	RD6MRK	/GET MARK TRACK
2503	1374	TAD	(=25	
2504	7440	SEA		/REVERSE EXPAND CODE?
2505	7402	HLT		/NO, ERROR
2506	5222	JMP	FWDEXP	
		/READ 1 SHIFT OF MARK TRACK SUBROUTINE		
2507	0000	RD1MRK,	0	
2510	6771	IOT120,	SDSS	
2511	5310	JMP	=1	
2512	6776	IOT121,	SDRC	
2513	0363	AND	(77	
2514	5707	JMP I	RD1MRK	
		/READ 6 SHIFTS OF MARK TRACK SUBROUTINE		
2515	0000	RD6MRK,	0	
2516	1362	TAD	(=6	
2517	3307	DCA	RD1MRK	
2520	6771	IOT122,	SDSS	
2521	5320	JMP	=1	
2522	6776	IOT123,	SDRC	
2523	2307	ISE	RD1MRK	
2524	5320	JMP	=4	
2525	0363	AND	(77	
2526	5715	JMP I	RD6MRK	
2562	7772			
2563	0097			
2564	7733			
2565	7727			
2566	7705			
2567	7710			
2570	7656			
2571	7770			
2572	7774			
2573	7746			
2574	7753			
2575	7732			
2576	7756			
2577	3000			
	2600			

PAGE

/CHECK SELECT ERROR STATUS BIT AND ABILITY TO CLEAR "WRITE"
 /UNIT 1 IS "OFF-LINE" OR NON-EXISTANT
 /UNIT 0 IS "ON LINE" AND "WRITE LOCKED"
 SELECT, CLA CLL

2600 7300

2601	1377	TAD	(MESS50	
2602	3025	DCA	HEAD1	
2603	1376	TAD	(MESS51	
2604	3026	DCA	HEAD2	
2605	6774	10T124, SDLC		
2606	6772	10T125, SDST		/IS TIMING ERROR SET?
2607	7410	SKP		
2610	4775'	JMS	ERROR2	/YES, ERROR
2611	1374	TAD	(MESS52	
2612	3026	DCA	HEAD2	
2613	1373	TAD	(4000	
2614	6774	10T126, SDLC		/SET UNIT BIT TO 1
2615	7200	CLA		
2616	6776	10T127, SORC		/READ STATUS
2617	3021	DCA	IN	/SAVE
2620	7604	LAS		
2621	7710	SPA CLA		/LOOP?
2622	5200	JMP	SELECT	/YES
2623	1021	TAD	IN	
2624	0372	AND	(100	
2625	7650	SNA CLA		/SELECT ERROR?
2626	4775'	JMS	ERROR2	/NO, ERROR
2627	7604	LAS		
2630	7710	SPA CLA		/LOOP?
2631	5200	JMP	SELECT	/YES
2632	1371	TAD	(MESS53	
2633	3026	DCA	HEAD2	
2634	1370	10T128, TAD	(4000	
2635	6774	10T128, SDLC		/TRY TO SET "WRITE"
2636	7604	LAS		
2637	7710	SPA CLA		/LOOP?
2640	5234	JMP	SELECT1	/YES
2641	6776	10T129, SORC		/READ STATUS
2642	3021	DCA	IN	/SAVE
2643	1021	TAD	IN	
2644	0367	AND	(400	
2645	7640	SEA CLA		/WRITE SET?
2646	4775'	JMS	ERROR2	/YES, ERROR
2647	7604	LAS		
2650	7710	SPA CLA		/LOOP?
2651	5234	JMP	SELECT1	/YES
2652	1366	TAD	(MESS63	
2653	3026	DCA	HEAD2	
2654	6774	10T130, SDLC		/SELECT UNIT 0
2655	7604	LAS		
2656	7710	SPA CLA		/LOOP?
2657	5234	JMP	SELECT2	/YES
2660	6776	10T130, SORC		/READ STATUS
2661	3021	DCA	IN	/SAVE
2662	1021	TAD	IN	
2663	0372	AND	(100	
2664	7640	SEA CLA		/SELECT ERROR?
2665	4775'	JMS	ERROR2	/YES
2666	7604	LAS		
2667	7710	SPA CLA		/LOOP?

```

2670 5254      JMP      SELECT2      /YES
2671 4046      JMS      LOOP1
2672 5200      JMP      SELECT

/CHECK WRITE LOCK OUT STATUS BIT AND ABILITY TO CLEAR "WRITE"
/UNIT 0 IS "WRITE=LOCKED"
WL0,
2673 7300      CLA      CLL
2674 1365      TAD      (MESS54
2675 3025      DCA      HEAD1
2676 1364      TAD      (MESS55
2677 3026      DCA      HEAD2
2700 6774      SDLC
2701 6776      IOT131, SORC          /READ STATUS
2702 3021      DCA      IN              /SAVE
2703 7604      LAS
2704 7710      SPA     CLA          /LOOP?
2705 5273      JMP      WL0            /YES
2706 1021      TAD      IN
2707 0363      AND      (200
2710 7600      SNA     CLA          /WRITE LOCK OUT BIT SET?
2711 4775      JMS      ERROR2       /NO, ERROR
2712 7604      LAS
2713 7710      SPA     CLA          /LOOP?
2714 5273      JMP      WL0            /YES
2715 1362      TAD      (MESS56
2716 3026      DCA      HEAD2
2717 1367      TAD      (400
2720 6774      IOT132, SDLC          /TRY TO SET "WRITE"
2721 7604      LAS
2722 7710      SPA     CLA          /LOOP?
2723 5317      JMP      WLE           /YES
2724 6776      IOT133, SORC          /READ STATUS
2725 3021      DCA      IN              /SAVE
2726 1021      TAD      IN
2727 0367      AND      (400
2730 7640      SEA     CLA          /WRITE SET?
2731 4775      JMS      ERROR2       /YES, ERROR
2732 7604      LAS
2733 7710      SPA     CLA          /LOOP?
2734 5317      JMP      WLE           /YES
2735 4046      JMS      LOOP1
2736 5273      JMP      WL0
2737 1361      TAD      (OR
2740 4760      JMS      MESSAGE
2741 4040      JMS      CRUF
2742 7402      HLT
2743 5342      JMP
2744 1713      OK,   TEXT      "=I
2745 0000      "OK"

2760 2264
2761 2744
2762 6767
2763 0200
2764 6744
2765 6730

```

2766 7107
 2767 0400
 2770 4400
 2771 6706
 2772 0100
 2773 4000
 2774 6664
 2775 0537
 2776 6653
 2777 6640
 3000

PAGE

/TD8EA READ-WRITE AND SEARCH TEST PROGRAM
 /COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS, 01754

7200
 7400

BUFF1=7200
 BUFF2=7400

/ROUTINE TO CHECK DATA TRANSFERS ON TAPE

3000	7300	XFER,	CLA	CLL	
3001	3001		DCA	IN	
3002	1377		TAD	(FILL0	
3003	4224		JMS	WREAD	/FILL A BUFFER, THEN WRITE AND READ 0'S
3004	1376		TAD	(FILL1	
3005	4224		JMS	WREAD	/FILL A BUFFER, THEN WRITE AND READ =1'S
3006	1375		TAD	(FILL25	
3007	4224		JMS	WREAD	/FILL A BUFFER, THEN WRITE AND READ 2525
3010	1374		TAD	(FILPAT	
3011	4224		JMS	WREAD	/FILL A BUFFER, THEN WRITE AND READ 2225, /522,2555
					/INCREMENT PATTERN
3012	1373		TAD	(FILINC	
3013	4224		JMS	WREAD	
3014	1372		TAD	(FILDEC	/DECREMENT PATTERN
3015	4224		JMS	WREAD	
3016	1371		TAD	(SPEC1	/6141
3017	4224		JMS	WREAD	
3020	1370		TAD	(SPEC2	/3434
3021	4224		JMS	WREAD	
3022	4767		JMS	PASCNT	
3023	5202		JMP	XFER+2	

/ROUTINE TO WRITE AND READ BACK AND COMPARE EVERY 100TH BLOCK ON TAPE

3024	0000	WREAD,	D		
3025	3030		DCA	FILPNT	
3026	3766		DCA	SUNIT	
3027	1365		TAD	(=000	
3030	4430		JMS	FILPNT	/FILL BUFF1 WITH DATA
3031	7200		BUFF1		
3032	3027		DCA	BLK	/ZERO BLOCK NUMBER
3033	1364		TAD	(MESS37	
3034	3025		DCA	HEAD1	
3035	1363	WRZL1,	TAD	(MESS58	
3036	3026		DCA	HEAD2	
3037	1027		TAD	BLK	

/TD8E DIAGNOSTIC

PAL10

V141

19-OCT-72

11109 PAGE 1-27

3040	4762'	JMS	WRITE	/WRITE ONTO TAPE
3041	7200	BUFF1		
3042	7600	=200		
3043	1361	TAD	(MESS59	
3044	3026	DCA	HEAD2	
3045	1027	TAD	BLK	
3046	4760'	JMS	READ	/READ BACK INTO MEMORY
3047	7400	BUFF2		
3050	7577	=201		
3051	1365	TAD	(=200	
3052	4757'	JMS	COMPAR	/COMPARE DATA
3053	7200	BUFF1		
3054	7401	BUFF2+1		
3055	1336	TAD	(MESS60	
3056	3026	DCA	HEAD2	
3057	1027	TAD	BLK	
3060	4755'	JMS	READR	/READ BACK BACKWARDS
3061	7400	BUFF2		
3062	7577	=201		
3063	1027	TAD	BLK	/BUMP BLOCK NUMBER
3064	1334	TAD	(100	
3065	3027	DCA	BLK	
3066	1027	TAD	BLK	
3067	1353	TAD	(=2701	
3070	7710	SPA	CLA	
3071	5235	JMP	WRRL1	
3072	1352	TAD	(2701	
3073	3027	DCA	BLK	/SET BLOCK NUMBER TO 2701
3074	1381	TAD	(MESS61	
3075	3025	DCA	HEAD1	
3076	1350	TAD	(MESS62	
3077	3026	DCA	HEAD2	
3100	1027	TAD	BLK	
3101	4747'	JMS	WRITER	/WRITE ONTO TAPE BACKWARDS
3102	7200	BUFF1		
3103	7600	=200		
3104	1356	TAD	(MESS60	
3105	3026	DCA	HEAD2	
3106	1027	TAD	BLK	
3107	4755'	JMS	READR	/READ BACK INTO MEMORY BACKWARDS
3110	7400	BUFF2		
3111	7577	=201		
3112	1365	TAD	(=200	
3113	4757'	JMS	COMPAR	/COMPARE DATA
3114	7200	BUFF1		
3115	7401	BUFF2+1		
3116	1361	TAD	(MESS59	
3117	3026	DCA	HEAD2	
3120	1027	TAD	BLK	
3121	4760'	JMS	READ	/READ BACK FORWARD
3122	7400	BUFF2		
3123	7577	=201		
3124	1027	TAD	BLK	/BUMP BLOCK NUMBER
3125	1346	TAD	(=100	

WRRL2,

3126 3027
 3127 1027
 3130 7700
 3131 5296
 3132 7604
 3133 7010
 3134 7630
 3135 5624
 3136 1766
 3137 1343
 3140 7450
 3141 5624
 3142 3766
 3143 5232
 3145 4000
 3146 7700
 3147 3671
 3150 7075
 3151 7062
 3152 2701
 3153 5057
 3154 0100
 3155 3714
 3156 7030
 3157 3200
 3160 4000
 3161 7037
 3162 4472
 3163 7035
 3164 7012
 3165 7600
 3166 4471
 3167 3477
 3170 3461
 3171 3443
 3172 3421
 3173 3400
 3174 3276
 3175 3200
 3176 3203
 3177 3227
 3200

DCA BLK
 TAD BLK
 SMA CLA
 JMP WREL2
 LAS
 RAR
 SZL CLA
 JMP I WREAD
 TAD SUNIT
 TAD (4000)
 SNA
 JMP I WREAD
 DCA SUNIT
 JMP WREL1-3

/TWO UNITS?
 /NO
 /YES, INCREMENT
 /UNIT
 /EXIT IF BACK TO ZERO
 /STORE BACK IF NO=ZERO
 /LOOP

PAGE

/SUBROUTINE TO COMPARE TWO DATA BUFFERS, INDICATE AN ERROR
 /CALLING SEQUENCE:
 / TAD (N) /MINUS (2'S) NUMBER OF WORDS TO COMPARE
 / JMS COMPAR /CALL SUBROUTINE
 / GADD /1ST ADDRESS OF GOOD DATA
 / TADD /1ST ADDRESS OF TEST DATA
 / /RETURN HERE WHEN DONE

3200 0000
 3201 3224
 3202 1000

COMPAR, 0
 DCA CCNTR
 TAD I COMPAR

3203	3225	DCA	GPNTR
3204	2200	ISZ	COMPAR
3205	1600	TAD I	COMPAR
3206	3226	DCA	TPNTR
3207	2200	ISZ	COMPAR
3210	1377	TAD	(DATMES
3211	3776'	DCA	DATHD
3212	1625	CONLUP, TAD I	GPNTR
3213	7041	CIA	
3214	1626	TAD I	TPNTR
3215	7640	SZA CLA	
3216	4775'	JMS	DAYERR
3217	2225	ISZ	GPNTR
3220	2226	ISZ	TPNTR
3221	2224	ISZ	CCNTR
3222	5212	JMP	CONLUP
3223	5600	JMP I	COMPAR
3224	0000	CCNTR, 0	
3225	0000	GPNTR, 0	
3226	0000	TPNTR, 0	

/SUBROUTINE TO FILL MEMORY WITH ZEROS

/CALLING SEQUENCE:

/	TAD	(=N	/MINUS (2/3) NUMBER OF WORDS TO FILL
/	JMS	FILL0	/CALL SUBROUTINE
/	ADDR		/1ST ADDRESS TO FILL

3227	0000	FILL0, 0	
3230	3241	DCA	FILL0C
3231	1627	TAD I	FILL0
3232	3242	DCA	FILL0P
3233	2227	ISZ	FILL0
3234	3642	DCA I	FILL0P
3235	2242	ISZ	FILL0P
3236	2241	ISZ	FILL0C
3237	5234	JMP	=0
3240	5627	JMP I	FILL0
3241	0000	FILL0C, 0	
3242	0000	FILL0P, 0	

/SUBROUTINE TO FILL MEMORY WITH -1 (7777)

/CALLING SEQUENCE:

/	TAD	(=N	/MINUS (2/3) NUMBER OF WORDS TO FILL
/	JMS	FILL1	/CALL SUBROUTINE
/	ADDR		/1ST ADDRESS TO FILL

3243	0000	FILL1, 0	
3244	3236	DCA	FILL1C
3245	1643	TAD I	FILL1
3246	3237	DCA	FILL1P
3247	2243	ISZ	FILL1
3250	7240	CLA CMA	
3251	3637	DCA I	FILL1P
3252	2257	ISZ	FILL1P

3253 2256
 3254 5250
 3255 5643
 3256 0000
 3257 0000

ISE FILL1C
 JMP ,=4
 JMP I FILL1
 FILL1C, 0
 FILL1P, 0
 /SUBROUTINE TO FILL MEMORY WITH 2525
 /CALLING SEQUENCE:
 / TAO (=N /MINUS (218) NUMBER OF WORDS TO FILL
 / JMS FILL25 /CALL SUBROUTINE
 / ADDR /1ST ADDRESS TO FILL

3260 0000
 3261 3273
 3262 1060
 3263 3275
 3264 2260
 3265 1274
 3266 3675
 3267 2275
 3270 2273
 3271 5265
 3272 5640
 3273 0000
 3274 2525
 3275 0000

FILL25, 0
 DCA FILL2C
 TAO I FILL25
 DCA FILL2P
 ISE FILL25
 TAO FILL2K
 DCA I FILL2P
 ISE FILL2P
 ISE FILL2C
 JMP ,=4
 JMP I FILL25
 FILL2C, 0
 FILL2K, 2525
 FILL2P, 0

/SUBROUTINE TO FILL MEMORY WITH 2225,5522,2555
 /CALLING SEQUENCE:
 / TAO (=N /MINUS (218) NUMBER OF WORDS TO FILL
 / JMS FILPAT /CALL SUBROUTINE
 / ADDR /1ST ADDRESS TO FILL

3276 0000
 3277 3323
 3300 1076
 3301 3321
 3302 2276
 3303 1325
 3304 3322
 3305 1331
 3306 3324
 3307 1722
 3310 3721
 3311 2321
 3312 2323
 3313 7410
 3314 5676
 3315 2322
 3316 2324
 3317 5307
 3320 5303
 3321 0000
 3322 0000
 3323 0000

FILPAT, 0
 DCA FILLC1
 TAO I FILPAT
 DCA FILLP1
 ISE FILPAT
 TAO FILTP
 DCA FILLP2
 TAO FILTC
 DCA FILLC2
 FILPL1, TAO I FILLP2
 DCA I FILLP1
 ISE FILLP1
 ISE FILLC1
 SKP
 JMP I FILPAT
 ISE FILLP2
 ISE FILLC2
 JMP FILPL2
 JMP FILPL1
 FILLP1, 0
 FILLP2, 0
 FILLC1, 0

3324 0000
 3325 3326
 3326 2225
 3327 5522
 3330 2555
 3331 7775
 3375 4000
 3376 4040
 3377 4042
 3400

FILLC2, 0
 FILTP, +1
 2225
 5522
 2555
 FILTC, FILTP=FILTC+1

PAGE

/SUBROUTINE TO FILL MEMORY WITH AN INCREMENT PATTERN
 /CALLING SEQUENCE:
 / TAD (=N /MINUS (2'S) NUMBER OF WORDS TO FILL
 / JMS FILING /CALL SUBROUTINE
 / ADDR /1ST ADDRESS TO FILL

3400 0000
 3401 3216
 3402 1600
 3403 3217
 3404 2200
 3405 3220
 3406 1220
 3407 3617
 3410 2220
 3411 7000
 3412 2217
 3413 2216
 3414 5200
 3415 5600
 3416 0000
 3417 0000
 3420 0000

FILING, 0
 DCA FILICT
 TAD I FILING
 DCA FILIPT
 ISZ FILING
 DCA FILIDT
 TAD FILIDT
 DCA I FILIPT
 ISZ FILIDT
 NOP
 ISZ FILIPT
 ISZ FILICT
 JMP ,=6
 JMP I FILING

FILICT, 0
 FILIPT, 0
 FILIDT, 0

/SUBROUTINE TO FILL MEMORY WITH A DECREMENT PATTERN
 /CALLING SEQUENCE:
 / TAD (=N /MINUS (2'S) NUMBER OF WORDS TO FILL
 / JMS FILDEC /CALL SUBROUTINE
 / ADDR /1ST ADDRESS TO FILL

3421 0000
 3422 3240
 3423 1621
 3424 3241
 3425 2221
 3426 3242
 3427 1242
 3430 3641
 3431 7040
 3432 1242
 3433 2241
 3434 2240
 3435 5226
 3436 7200

FILDEC, 0
 DCA FILOCT
 TAD I FILDEC
 DCA FILOPT
 ISZ FILDEC
 DCA FILODT
 TAD FILODT
 DCA I FILOPT
 CHA
 TAD FILODT
 ISZ FILOPT
 ISZ FILOCT
 JMP ,=9
 CLA

3437 5621
 3440 0000
 3441 0000
 3442 0000

JMP I FILEDC
 FILEDC, 0
 FILEDT, 0
 FILEDD, 0

/SUBROUTINE TO FILL MEMORY WITH 6161

/CALLING SEQUENCE:

/ TAD (=N /MINUS (219) NUMBER OF WORDS TO FILL
 / JMS SPEC1 /CALL SUBROUTINE
 / ADDR /1ST ADDRESS TO FILL
 /

3443 0000
 3444 3256
 3445 1643
 3446 3257
 3447 2243
 3450 1260
 3451 3657
 3452 2257
 3453 2256
 3454 5200
 3455 5643
 3456 0000
 3457 0000
 3460 6161

SPEC1, 0
 DCA SP1CT
 TAD I SPEC1
 DCA SP1PT
 ISE SPEC1
 TAD SPEC1D
 DCA I SP1PT
 ISE SP1PT
 ISE SP1CT
 JMP ,=4
 JMP I SPEC1

SP1CT, 0
 SP1PT, 0
 SPEC1D, 6161

/SUBROUTINE TO FILL MEMORY WITH 3434

/CALLING SEQUENCE:

/ TAD (=N /MINUS (219) NUMBER OF WORDS TO FILL
 / JMS SPEC2 /CALL SUBROUTINE
 / ADDR /1ST ADDRESS TO FILL
 /

3461 0000
 3462 3274
 3463 1661
 3464 3275
 3465 2261
 3466 1276
 3467 3675
 3470 2275
 3471 2274
 3472 5266
 3473 5661
 3474 0000
 3475 0000
 3476 3434
 3477 0000
 3500 4040
 3501 1377
 3502 4776
 3503 2021
 3504 7000
 3505 1021
 3506 4775
 3507 1374

SPEC2, 0
 DCA SP2CT
 TAD I SPEC2
 DCA SP2PT
 ISE SPEC2
 TAD SPEC2D
 DCA I SP2PT
 ISE SP2PT
 ISE SP2CT
 JMP ,=4
 JMP I SPEC2

SP2CT, 0
 SP2PT, 0
 SPEC2D, 3434
 PASCNT, 0

JMS CRUF
 TAD (PASS
 JMS MESSAGE
 ISE IN
 NOP
 TAD IN
 JMS OPRINT
 TAD (COMP

```

3510 4776'
3511 4040
3512 5677
3513 2001
3514 2323
3515 4000
3516 4003
3517 1715
3520 2014
3521 0524
3522 0500
3574 3516
3575 2316
3576 2264
3577 3513
3600

```

PAGE

/REVERSE SEARCH SUBROUTINE

```

3600 0000 RSRCH, 0
3601 3270 DCA RSLOOK
3602 1054 TAD M10 /SET P A COUNT OF 10 TIMES
3603 3055 DCA BLKTRY /TO SEARCH FOR A BLOCK
3604 1377 TAD (1000
3605 1776' TAD SUNIT
3606 6774 RSRCH0, SDC
3607 6776 IOT134, SDC
3610 0375 AND (100
3611 7640 SZA CLA
3612 5563 JMP I [SELEERR
3613 4774' JMS RDSQUAD
3614 4774' JMS RDSQUAD
3615 6771 RSRCH1, SDSS
3616 7410 SKP
3617 6777 IOT135, SDRD
3620 6771 IOT136, SDSS
3621 5220 JMP =I
3622 6776 IOT138, SDC
3623 0373 AND (77
3624 1372 TAD (=26 /BLOCK MARK
3625 7490 SNA
3626 5240 JMP RSRCH2 /YES
3627 1371 TAD (4 /END ZONE
3630 7640 SZA CLA
3631 5215 JMP RSRCH1 /NO, GO READ AGAIN
3632 6776 IOT13A, SDC /READ THE C.R.
3633 7006 RTL /SET THE DIRECTION BIT IN LINK
3634 7032 CHL RTR /COMPLEMENT IT FOR TURN AROUND
3635 2055 ISE BLKTRY /INCREMENT BLOCK TRY COUNTER
3636 5206 JMP RSRCH0
3637 5261 JMP RSTURN+5 /COULDN'T FIND BLOCK AFTER 8 TRIES
3640 6776 RSRCH2, SDC
3641 7006 RTL

```

```

3642 6777 10T137, SDRD /READ THE BLOCK NUMBER
3643 7041 CIA
3644 1270 TAD RSLOOK
3645 7450 SNA
3646 5265 JMP RLOC80
3647 7041 CIA
3650 7420 SNL
3651 1371 TAD (4
3652 7630 SZL CLA
3653 5215 JMP RSRCH1
3654 6776 RSTURN, SDRC
3655 7006 RTL
3656 7032 CML RTR
3657 2035 ISE BLKTRY
3660 5206 JMP RSRCH0
3661 7200 CLA
3662 1270 TAD RSLOOK
3663 7402 HLT /AC=THE BLOCK THAT IT WAS LOOKING FOR
3664 5263 JMP ,=1 /BUT FAILED TO FIND AFTER 10 TRIES;
3665 7630 RLOC80, SZL CLA
3666 5215 JMP RSRCH1
3667 5600 JMP I RSRCH
3670 0000 RSLOOK, 0

```

/WRITE REVERSE SUBROUTINE

```

3671 0000 WRITER, 0
3672 3770' DCA WCNT
3673 1671 TAD I WRITER
3674 3767' DCA WADDR
3675 2271 ISE WRITER
3676 1671 TAD I WRITER
3677 3766' DCA WCOUNT
3700 1271 TAD WRITER
3701 7001 IAC
3702 3765' DCA WRITE
3703 4764' JMS CSUMRT /CALCULATE THE CHECKSUM
3704 0025 25
3705 7177 BUFF1=1
3706 7600 =200
3707 4763' JMS SBXOR
3710 4331 JMS WRFLCK /CHECK FOR WRITE LOCK OUT
3711 1770' TAD WCNT
3712 4200 JMS RSRCH
3713 5762' JMP WRITE1

```

/READ REVERSE SUBROUTINE

```

3714 0000 READR, 0
3715 3761' DCA RCNT
3716 1714 TAD I READR
3717 3760' DCA RADDR
3720 2314 ISE READR
3721 1714 TAD I READR
3722 3757' DCA RCOUNT
3723 1314 TAD READR
3724 7001 IAC

```

3725	3756'	DCA	READ
3726	1761'	TAD	RCNT
3727	4200	JMS	RSERCH
3730	5755'	JMP	READ1

3731	0000	WRTLCK, 0		/ROUTINE TO CHECK FOR WRITE LOCKOUT
3732	1776'	TAD	SUNIT	
3733	6774	IOT151, SCLC		
3734	6776	IOT15A, SDRC		
3735	0354	AND	(200	
3736	7640	SEA	CLA	
3737	5562	JMP I	CWROERR	
3740	5731	JMP I	WRTLCK	

3754 0200
 3755 4612
 3756 4600
 3757 4660
 3760 4657
 3761 4656
 3762 4512
 3763 4714
 3764 4303
 3765 4472
 3766 4945
 3767 4564
 3770 4470
 3771 0004
 3772 7752
 3773 0077
 3774 4707
 3775 0100
 3776 4471
 3777 1000
 4000

PAGE

/DATA ERROR HANDLER

4000	0000	DAYERR, 0	
4001	7604	LAS	
4002	0377	AND	(400
4003	7640	SEA	CLA
4004	5233	JMP	DAYHLT=3
4005	1240	TAD	DAYHD
4006	7650	SNA	CLA
4007	5220	JMP	DAYNUM
4010	4250	JMS	HEADTP
4011	1240	TAD	DAYHD
4012	4776'	JMS	MESSAGE
4013	3240	DCA	DAYHD
4014	4040	JMS	CRLF
4015	1375	TAD	(FORMT1
4016	4776'	JMS	MESSAGE

4017 4040
 4020 1774'
 4021 3241
 4022 1641
 4023 4773'
 4024 1372
 4025 4031
 4026 1771'
 4027 3241
 4030 1641
 4031 4773'
 4032 4040
 4033 7604
 4034 0370
 4035 7650
 4036 7402
 4037 5000
 4040 0000
 4041 0000
 4042 0401
 4043 2401
 4044 4005
 4045 2222
 4046 1722
 4047 0000

JMS CRLF
 DATNUM, TAD GPNTR
 DCA DATPNT
 TAD I DATPNT
 JMS OPRINT
 TAO (240
 JMS TYPE
 TAO TPNTR
 DCA DATPNT
 TAD I DATPNT
 JMS OPRINT
 JMS CRLF
 LAS
 AND (200
 SNA CLA
 DATHLT, HLT
 JMP I DATERR
 DATHD, 0
 DATPNT, 0
 DATMES, TEXT "DATA ERROR"

/SUBROUTINE TO TYPE OUT HEADER FOR DATA TESTS

4050 0000
 4051 4040
 4052 1367
 4053 4776'
 4054 1372
 4055 4031
 4056 6776
 4057 7710
 4060 7001
 4061 1366
 4062 4031
 4063 4040
 4064 1365
 4065 4776'
 4066 1372
 4067 4031
 4070 1027
 4071 4773'
 4072 4040
 4073 1025
 4074 4776'
 4075 4040
 4076 1026
 4077 4776'
 4100 4040
 4101 5650
 4102 2516
 4103 1124

HEADTP, 0
 JMS CRLF
 TAO (UMESS
 JMS MESSAGE
 TAO (240
 JMS TYPE
 IOT139, SDRG
 SPA CLA
 IAO
 TAO (240
 JMS TYPE
 JMS CRLF
 TAO (UMESS
 JMS MESSAGE
 TAO (240
 JMS TYPE
 TAO BLK
 JMS OPRINT
 JMS CRLF
 TAO HEAD1
 JMS MESSAGE
 JMS CRLF
 TAO HEAD2
 JMS MESSAGE
 JMS CRLF
 JMP I HEADTP
 UMESS, TEXT "UNIT"

4104 0000
 4105 0214
 4106 1703
 4107 1300

BMESS, TEXT "BLOCK"

/CHECKSUM ERROR HANDLER

4110 0000
 4111 3331
 4112 6776
 4113 0364
 4114 6774
 4115 4250
 4116 1363
 4117 4776
 4120 4040
 4121 7604
 4122 0370
 4123 7640
 4124 5710
 4125 1331
 4126 7402
 4127 7200
 4130 5710
 4131 0000
 4132 0310
 4133 0503
 4134 1323
 4135 2515
 4136 4005
 4137 2222
 4140 1722
 4141 0000

CHKERR, 0
 OCA CHKDAT
 IOT140, SRC (4000) /STOP TAPE
 AND
 IOT141, SDC
 JMS HEADTP
 TAD (CHKMES
 JMS MESSAGE
 JMS CRLF
 LAS
 AND (200
 SEA CLA
 JMP I CHKERR
 TAD CHKDAT
 CHKHLT, HLT
 CLA
 JMP I CHKERR
 CHKDAT, 0
 CHKMES, TEXT "CHECKSUM ERROR"

4163 4132
 4164 4000
 4165 4105
 4166 0260
 4167 4102
 4170 0200
 4171 3226
 4172 0240
 4173 2316
 4174 3225
 4175 0532
 4176 2264
 4177 0400
 4200

PAGE

/WRITE LOCK OUT ERROR

4200 4777
 4201 6776

WROERR, JMS HEADTP
 IOT142, SRC

/STOP TAPE

```

4202 0376          AND          (4000
4203 6774      IOT143, SDLC
4204 1375          TAD          (WRDMES
4205 4774'        JMS          MESSAGE
4206 4040          JMS          CRLF
4207 7604          LAS
4210 0373          AND          (200
4211 7600          SNA CLA
4212 7402      WROHLT, HLT
4213 5772'        JMP          WRRL1+2

4214 2516      WRDMES, TEXT      "UNIT WRITE LOCKED"
4215 1124
4216 4027
4217 2211
4220 2405
4221 4014
4222 1703
4223 1305
4224 0400

```

/SELECT ERROR HANDLER

```

4225 4777'      SELERR, JMS      HEADTP
4226 1371        TAD          (SELMES
4227 4774'        JMS          MESSAGE
4230 4040          JMS          CRLF
4231 7604          LAS
4232 0373          AND          (200
4233 7600          SNA CLA
4234 7402      SELHLT, HLT
4235 5772'        JMP          WRRL1+2

4236 2505      SELMES, TEXT      "SELECT ERROR"
4237 1400
4240 0324
4241 4000
4242 2222
4243 1722
4244 0000

```

/TIMING ERROR HANDLER

```

4245 0000      TYMERR, 0
4246 6776      IOT144, SDRC
4247 0376          AND          (4000
4250 6774      IOT145, SDLC
4251 4777'        JMS          HEADTP
4252 1370        TAD          (TYMHES
4253 4774'        JMS          MESSAGE
4254 4040          JMS          CRLF
4255 7604          LAS
4256 0373          AND          (200

/STOP TAPE

```

4257 7650
 4260 7402
 4261 5767
 4262 2411
 4263 1511
 4264 1607
 4265 4005
 4266 2222
 4267 1722
 4270 0000

SNA CLA
 TYMHLT, HLT
 JMP WREAD+3
 TYMMES, TEXT "TIMING ERROR"

4271 0000
 4272 6773
 4273 3272
 4274 6772
 4275 7610
 4276 4161
 4277 6776
 4300 0366
 4301 6774
 4302 5671

/SUBROUTINE TO CLEAR WRITE AFTER QUAD LINE FLAG
 CLRWT, 0
 IOT168, SDSQ /WAIT FOR QUAD LINE FLAG
 JMP I=I
 IOT154, SDST /TIMING ERROR
 SKP CLA /NO
 JMS CTYMERR /YES
 IOT169, SDRC /READ THE COMMAND REGISTER
 AND (7000) /MASK OFF WRITE BIT
 IOT170, SDLC /LOAD THE COMMAND REGISTER
 JMP I CLRWT /EXIT

4303 0000
 4304 1703
 4305 3765
 4306 2303
 4307 1703
 4310 3017
 4311 2303
 4312 1703
 4313 3322
 4314 2303
 4315 1417
 4316 4764
 4317 2322
 4320 5315
 4321 5703
 4322 0000

CSUMRT, 0
 TAD I CSUMRT
 DCA CHKSUM
 ISR CSUMRT
 TAD I CSUMRT
 DCA AUTO
 ISR CSUMRT
 TAD I CSUMRT
 DCA XXX
 ISR CSUMRT
 TAD I AUTO
 JMS SBOXOR
 ISR XXX
 JMP I=I
 JMP I CSUMRT
 XXX, 0

4323 0000
 4324 4303
 4325 0000
 4326 7377
 4327 7075
 4330 1765
 4331 7040
 4332 0363
 4333 7440
 4334 4560
 4335 5723

CHKCHK, 0
 JMS CSUMRT
 0
 BUFR2=1
 -203
 TAD CHKSUM
 CMA
 AND (77)
 SZA
 JMS I [CHKERR /CHECK SUM ERROR
 JMP I CHKCHK /RETURN

4363 0077
 4364 4714

4365 4744
 4366 7000
 4367 3027
 4370 4262
 4371 4236
 4372 3037
 4373 0200
 4374 2264
 4375 4214
 4376 4000
 4377 4000
 4400

PAGE

/TDS=EA READ=WRITE=AND=SEARCH SUBROUTINES
 /COPYRIGHT 1971, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS, 01704
 /DECTAPE COMMANDS

6771 S0SS=6771 /SKIP ON SINGLE LINE FLAG
 6772 S0ST=6772 /SKIP ON TIMING ERROR
 6773 S0SQ=6773 /SKIP ON QUADRUPLE LINE FLAG
 6774 S0LC=6774 /LOAD COMMAND REGISTER
 6775 S0LD=6775 /LOAD DATA REGISTER, CLEAR FLAGS
 6776 S0RC=6776 /READ COMMAND REGISTER AND MARK TRACK, CLEAR FLAG
 6777 S0RD=6777 /READ DATA REGISTER, CLEAR FLAGS

/SEARCH SUBROUTINE
 /SUBROUTINE IS ENTERED WITH THE NUMBER OF THE DESIRED BLOCK IN THE AC
 /PROGRAM WILL EXIT WITH TAPE MOVING IN THE FORWARD DIRECTION
 /UNIT BIT IS IN SUNIT, BIT 0, BITS 1 TO 11 ARE 0

4400 0000 SEARCH, 0
 4401 3270 DCA SLOK /SAVE BLOCK NUMBER
 4402 1054 TAD M10 /SET UP A COUNT OF 10
 4403 3039 DCA BLRTRY /TO SEARCH FOR A BLOCK;
 4404 1377 TAD I3000 / PUT IN MOTION BACKWARD
 4405 1271 TAD SUNIT
 4406 6774 SRCH0, S0LC /LOAD CONTROL WITH UNIT, REV, GO, READ
 4407 6776 IOT146, S0RC /READ STATUS
 4410 0376 AND I100
 4411 7640 SBA CLA /SELECT ERROR?
 4412 0563 JMP I ISELERR /YES
 4413 4775 JMS ROQUAD /DELAY TO ASSURE
 4414 4775 JMS ROQUAD /MARK WINDOW OPEN
 4415 6771 SRCH1, S0SS /SINGLE LINE FLAG
 4416 7410 SKP /NO
 4417 6777 IOT147, S0RD /YES
 4420 6771 IOT148, S0SS /SKIP ON SINGLE LINE FLAG
 4421 5220 JMP I=1
 4422 6776 IOT149, S0RC /READ MARK TRACK AND COMMAND REGISTER
 4423 0376 AND I79 /MASK TO MARK TRACK BITS
 4424 1373 TAD I=26 /BLOCK MARK ?
 4425 7450 SNA
 4426 5240 JMP SRCH2 /YES, GO READ THE BLOCK NUMBER
 4427 1372 TAD I4 /END ZONE ?

```

4430 7640          SZA  CLA
4431 5215          JMP  SRCH1  /NO, GO GET NEXT WORD
4432 6776 IOT14A, SDRG          /READ THE COMMAND REG;
4433 7006          RTL
4434 7032          CML  RTR  /TURN THE TAPE AROUND
4435 2055          ISZ  BLKTRY /8 TRIES ?
4436 5206          JMP  SRCH0  /NO, TRY AGAIN
4437 5261          JMP  BADBLK /YES, CAN NOT FIND BLOCK
4440 6776 SRCH2, SDRG          /READ COMMAND REGISTER
4441 7006          RTL  /MOVE DIRECTION BIT INTO THE LINK
4442 6777 IOT150, SDRD          /GET BLOCK NUMBER FOUND
4443 7041          CIA
4444 1270          TAD  SLOOK /COMBINE WITH BLOCK LOOKED FOR
4445 7450          SNA          /CURRENT BLOCK?
4446 5265          JMP  LOGGED /YES, CHECK DIRECTION
4447 7041          CIA  /NO, TAKE 2'S COMPLEMENT
4450 7420          SNL          /LINK IS 1 IF BACKWARD AND NOT AT OR LOWER THAN BLOCK
4451 1371          TAD  (2  /ADD TWO TO ENABLE TURN AROUND
4452 7630          SZL  CLA  /TURN AROUND (3 BEYOND)?
4453 5215          JMP  SRCH1  /NO, DON'T TURN AROUND

4454 6776 IOT150, SDRG          /READ THE COMMAND REGISTER
4455 7006          RTL  /MOVE THE DIRECTION BIT INTO LINK
4456 7032          CML  RTR  /COMPLEMENT THE DIRECTION BIT
4457 2055          ISZ  BLKTRY /8 TRIES ?
4460 5206          JMP  SRCH0  /NO, GO SEARCH AGAIN
4461 7200 BADBLK, CLA
4462 1270          TAD  SLOOK
4463 7402          HLT          /AG=THE BLOCK BEING SEARCHED FOR BUT FAILED
4464 5263          JMP  =1  /TO FIND AFTER 8 TRIES
4465 7620 LOGGED, SNL  CLA  /FOUND BLOCK FORWARD?
4466 5215          JMP  SRCH1  /NO
4467 5000          JMP  I  SEARCH /YES, EXIT
4470 0000 SLOOK, 0  /BLOCK NUMBER LOOKED FOR
4471 0000 SUNIT, 0  /CURRENT UNIT

```

```

/WRITE SUBROUTINE
/CALLING SEQUENCE:
/   TAD  (BLKNO /FIRST BLOCK TO BE WRITTEN INTO
/   JMP  WRITE  /CALL SUBROUTINE
/   ADDRESS  /ADDRESS OF FIRST DATA WORD
/   =N      /MINUS (2'S) NUMBER OF WORDS TO TRANSFER
/   /RETURN HERE
/120 WORDS PER BLOCK WILL BE WRITTEN FROM MEMORY

```

```

4472 0000 WRITE, 0
4473 3270 DCA  WCNT  /SAVE BLOCK NUMBER
4474 1072 TAD  I  WRITE
4475 3344 DCA  WADR  /SAVE ADDRESS
4476 2272 ISZ  WRITE
4477 1072 TAD  I  WRITE
4500 3345 DCA  WCUNT /SAVE WORD COUNT
4501 2272 ISZ  WRITE
4502 4770 JMP  CSUMRY
4503 0025 25

```

4504	7177		BUFF1=1	
4505	7000		=200	
4506	4767	JMS	SBCXOR	
4507	4766	JMS	WRFLCK	/CHECK FOR WRITE LOCKOUT
4510	1270	TAD	WCNT	
4511	4200	JMS	SEARCH	/FIND BLOCK
4512	4765	WRITE1, JMS	REVGRD	/WAIT FOR REVERSE GUARD
4513	4775	JMS	ROGUAD	/DELAY TWO=THIRDS THRU LOCK
4514	6776	IOT152, SDRG		
4515	1364	TAD	(400	
4516	6774	IOT153, SOLC		/LOAD CONTROL WITH UNIT, FWD, GO, WRITE
4517	1363	TAD	(25	
4520	4762	JMS	WRQUAD	/WRITE REVERSE CHECKSUM
4521	1744	WRITE2, TAD I	WADDR	/GET THE DATA WORD
4522	2344	ISE	WADDR	/INCREMENT ADDRESS
4523	7000	NOP		/SAFETY NOP
4524	4762	JMS	WRQUAD	/WRITE DATA WORD ON TAPE
4525	2345	ISE	WCOUNT	/WORD 1287
4526	5321	JMP	WRITE2	/NO
4527	4762	JMS	WRQUAD	/YES WRITE A 0 (WORD 129)
4530	1761	TAD	CHRSUM	
4531	7040		CMA	
4532	0374		AND	(77
4533	7106		RTL	CLL
4534	7006		RTL	
4535	7006		RTL	
4536	4762	JMS	WRQUAD	/WRITE CHECKSUM
4537	4760	JMS	CLRNT	/WAIT FOR CHECKSUM TO BE WRITTEN, CLEAR "WRITE"
4540	6776	IOT155, SDRG		
4541	0387		AND	(4000
4542	6774	IOT156, SOLC		/STOP TAPE
4543	5672	JMP I	WRITE	
	4470	WCNT=BLOCK		/BLOCK NUMBER, ALSO BLOCK DATA COUNTER
4544	0000	WADDR, 0		/WORD ADDRESS
4545	0000	WCOUNT, 0		/WORD COUNT
4557	4000			
4560	4271			
4561	4744			
4562	4701			
4563	0000			
4564	0400			
4565	4041			
4566	3751			
4567	4714			
4570	4303			
4571	0002			
4572	0004			
4573	7752			
4574	0077			
4575	4707			
4576	0100			
4577	3000			
	4000			

```

/READ SUBROUTINE
/CALLING SEQUENCE:
/   TAD      (BLKNO /FIRST BLOCK TO BE READ FROM
/   JMS      READ   /CALL SUBROUTINE
/   ADDRESS  /ADDRESS FOR FIRST DATA WORD
/   =N       /MINUS (2'S) NUMBER OF WORDS TO TRANSFER
/           /RETURN HERE
/128 WORDS PER BLOCK WILL BE READ INTO MEMORY

```

```

4600 0000 READ, 0
4601 3256 DCA RCNT /SAVE BLOCK NUMBER
4602 1600 TAD I READ
4603 3257 DCA RADDR /SAVE ADDRESS
4604 2200 ISZ READ
4605 1600 TAD I READ
4606 3260 DCA RCOUNT /SAVE WORD COUNT
4607 2200 ISZ READ
4610 1256 TAD RCNT
4611 4777 JMS SEARCH /FIND BLOCK
4612 6771 READ1, SDSS /WAIT FOR REVERSE GUARD
4613 5212 JMP I=-I
4614 6776 IOT16A, SDRG /READ THE MARK TRACK
4615 0376 AND (79
4616 1375 TAD (=32
4617 7450 SNA /REVERSE GUARD
4620 5225 JMP I,+5 /YES, EXIT
4621 1374 TAD (10 /NO
4622 7640 SZA CLA /END ZONE ?
4623 5212 JMP READ1 /NO
4624 5274 JMP IOT162 /YES STOP TAPE
4625 4307 JMS RDQUAD
4626 4307 JMS RDQUAD /WAIT FOR
4627 4307 JMS RDQUAD /REVERSE CHECKSUM
4630 0376 AND (79 /MASK
4631 7410 SKP /STORE THE WORD
4632 4307 READ2, JMS RDQUAD /GET DATA WORD
4633 3657 DCA I RADDR
4634 2257 ISZ RADDR
4635 7000 NOP /SAFETY NOP
4636 2260 ISZ RCOUNT /128 DATA WORDS?
4637 5232 JMP READ2 /NO
4640 4307 JMS RDQUAD /YES, GET WORD 129
4641 3657 DCA I RADDR /STORE IT
4642 2257 ISZ RADDR
4643 4307 JMS RDQUAD /GET FORWARD CHECKSUM

4644 0373 AND (7700
4645 3657 DCA I RADDR
4646 6772 IOT157, SDST
4647 7410 SKP
4650 4561 JMS I CTYMER /TIMING ERROR
4651 6776 IOT158, SDRG
4652 0372 AND (4000

```



```

4653 6774 IOT159, SOLC /STOP TAPE
4654 4771 JMS CHKCHK /CALCULATE AND CHECK CHECK SUM
4655 5600 JMP I READ
4656 0000 RCNT, 0 /BLOCK NUMBER, ALSO BLOCK DATA COUNTER
4657 0000 RADDR, 0 /WORD ADDRESS
4660 0000 RCOUNT, 0 /WORD COUNT

/WAIT FOR REVERSE GUARD SUBROUTINE
4661 0000 REVGRO, 0
4662 6771 IOT160, SDBS /WAIT FOR MARK TRACK CHANGE
4663 5262 JMP ,=1
4664 6776 IOT161, SDRS /READ MARK TRACK
4665 0376 AND (77
4666 1375 TAD (=52
4667 7450 SNA /REVERSE GUARD?
4670 5661 JMP I REVGRO /YES, EXIT
4671 1374 TAD (10 /NO
4672 7440 SEA CLA /END ZONE?
4673 5262 JMP REVGRO+1 /NO
4674 6776 IOT162, SDRS /YES, STOP TAPE
4675 0372 AND (4800
4676 6774 IOT163, SOLC
4677 7402 HLT
4700 5277 JMP ,=1 /NON-RECOVERABLE ERROR, PROGRAM
/FOUND ENDOONE WHILE LOOKING FOR REV GRO
/BLOCK PROBABLY ABOVE 2701

/WRITE A "QUAD WORD" (12 BIT WORD) SUBROUTINE
4701 0000 WRQUAD, 0
4702 6773 IOT164, SDBS /WAIT FOR NEXT QUAD FLAG
4703 5302 JMP ,=1
4704 6775 IOT165, SDRS /LOAD DATA BUFFER TO WRITE ON TAPE
4705 7600 M0200A, CLA+400 /CLEAR AC
4706 5701 JMP I WRQUAD /EXIT

/READ A "QUAD WORD" (12 BIT WORD) SUBROUTINE
4707 0000 R0QUAD, 0
4710 6773 IOT166, SDBS /WAIT FOR QUAD FLAG
4711 5310 JMP ,=1
4712 6777 IOT167, SDRS /READ DATA BUFFER, CLEAR FLAG
4713 5707 JMP I R0QUAD

/SIXBIT COMPLEMENT XOR SUBROUTINE
/SUBROUTINE IS ENTERED WITH DATA WORD TO BE XORED IN AC
/TWO 6-BIT COMPLEMENT XORS WILL TAKE PLACE TO LOCATION CHKSUM
/WITH THE RESULT IN CHKSUM

4714 0000 SBCKOR, 0
4715 7040 CHA /COMPLEMENT WORD
4716 3345 DCA SBWORD /AND SAVE
4717 1345 TAD SBWORD
4720 0344 AND CHKSUM
4721 7041 CIA
4722 7104 CLL RAL
4723 1345 TAD SBWORD
4724 1344 TAD CHKSUM

```

4725 3344
 4726 1345
 4727 7112
 4730 7012
 4731 7012
 4732 3345
 4733 1345
 4734 0344
 4735 7041
 4736 7104
 4737 1345
 4740 1344
 4741 0376
 4742 3344
 4743 5714
 4744 0000
 4745

DCA CHKSUM
 TAD SBWORD
 RTR CLL; RTR; RTR

DCA SBWORD
 TAD SBWORD
 AND CHKSUM
 CIA
 CLL RAL
 TAD SBWORD
 TAD CHKSUM
 AND (7)
 DCA CHKSUM
 JMP I SBEXOR

CHKSUM, 0
 SBWORD,

4771 4323
 4772 4000
 4773 7700
 4774 0010
 4775 7746
 4776 0077
 4777 4400
 5000

PAGE

/MESSAGES

MESS1, TEXT "LOAD AND READ DATA REGISTER ERROR"

5000 1417
 5001 0104
 5002 4001
 5003 1604
 5004 4022
 5005 0501
 5006 0440
 5007 0401
 5010 2401
 5011 4022
 5012 0507
 5013 1125
 5014 2405
 5015 2240
 5016 0522
 5017 2217
 5020 2200
 5021 1417
 5022 0104
 5023 4001
 5024 1604
 5025 4022
 5026 0501
 5027 0440

MESS2, TEXT "LOAD AND READ COMMAND REGISTER ERROR"

5030	0317		
5031	1515		
5032	0116		
5033	0440		
5034	2205		
5035	0711		
5036	2324		
5037	0522		
5040	4005		
5041	2222		
5042	1722		
5043	0000		
5044	1116	MESS3, TEXT	"INITIALIZE TEST"
5045	1124		
5046	1101		
5047	1411		
5050	3205		
5051	4024		
5052	0523		
5053	2400		
5054	0301	MESS4, TEXT	"CAF DID NOT CLEAR COMMAND REGISTER"
5055	0640		
5056	0411		
5057	0440		
5060	1617		
5061	2440		
5062	0314		
5063	0501		
5064	2340		
5065	0317		
5066	1515		
5067	0116		
5070	0440		
5071	2205		
5072	0711		
5073	2324		
5074	0522		
5075	0000		
5076	0310	MESS5, TEXT	"CHECK SOLO, SOLO, SORD, SORD AND AC CLEAR"
5077	0503		
5100	1340		
5101	2304		
5102	1403		
5103	5440		
5104	2304		
5105	1404		
5106	5440		
5107	2304		
5110	2203		
5111	5440		
5112	2304		
5113	2204		
5114	4001		
5115	1404		

5117	0340		
5120	0314		
5121	0501		
5122	2200		
5123	2304	MESS6, TEXT	"SDLC DID NOT CLEAR AC"
5124	1403		
5125	4004		
5126	1104		
5127	4016		
5130	1724		
5131	4003		
5132	1405		
5133	0122		
5134	4001		
5135	0300		
5136	2304	MESS7, TEXT	"SDRC DID NOT CLEAR AC"
5137	2203		
5140	4004		
5141	1104		
5142	4016		
5143	1724		
5144	4003		
5145	1405		
5146	0122		
5147	4001		
5150	0300		
5151	2304	MESS8, TEXT	"SOLD CLEARED AC"
5152	1404		
5153	4003		
5154	1405		
5155	0122		
5156	0504		
5157	4001		
5160	0300		
5161	2304	MESS9, TEXT	"SORD DID NOT CLEAR AC"
5162	2204		
5163	4004		
5164	1104		
5165	4016		
5166	1724		
5167	4003		
5170	1405		
5171	0122		
5172	4001		
5173	0300		
5174	2311	MESS10, TEXT	"SINGLE LINE FLAG SKIP INSTRUCTION AND LOGIC"
5175	1007		
5176	1405		
5177	4014		
5200	1116		
5201	0540		
5202	0614		
5203	0107		
5204	4003		
5205	1311		

5206 2040
5207 1116
5210 2324
5211 2225
5212 0324
5213 1117
5214 1640
5215 0116
5216 0440
5217 1617
5220 0711
5221 0300
5222 2311
5223 1607
5224 1405
5225 4014
5226 1116
5227 0540
5230 0614
5231 0107
5232 4001
5233 1427
5234 0131
5235 2340
5236 2305
5237 2440
5240 1722
5241 4023
5242 0423
5243 2340
5244 0114
5245 2701
5246 3123
5247 4023
5250 1311
5251 2023
5252 0000
5253 2311
5254 1607
5255 1405
5256 4014
5257 1116
5260 0540
5261 0614
5262 0107
5263 4004
5264 1705
5265 2340
5266 1617
5267 2440
5270 2305
5271 2440
5272 1722
5273 4023
5274 0423

MESS11: TEXT

"SINGLE LINE FLAG ALWAYS SET OR SSSS ALWAYS SKIPS"

MESS12: TEXT

"SINGLE LINE FLAG DOES NOT SET OR SSSS DOES NOT SKIP"

5275 2340
5276 0417
5277 0523
5300 4016
5301 1724
5302 4023
5303 1311
5304 2000
5305 2311
5306 1607
5307 1405
5310 4014
5311 1116
5312 0540
5313 0614
5314 0107
5315 4003
5316 1405
5317 0122
5320 0504
5321 4002
5322 3140
5323 2304
5324 2323
5325 0000
5326 2311
5327 1607
5330 1405
5331 4014
5332 1116
5333 0540
5334 0614
5335 0107
5336 4016
5337 1724
5340 4003
5341 1405
5342 0122
5343 0504
5344 4002
5345 3140
5346 0301
5347 0600
5350 2311
5351 1607
5352 1405
5353 4014
5354 1116
5355 0540
5356 0614
5357 0107
5360 4016
5361 1724
5362 4003
5363 1405

MESS13, TEXT

"SINGLE LINE FLAG CLEARED BY S0SS"

MESS14, TEXT

"SINGLE LINE FLAG NOT CLEARED BY GAF"

MESS15, TEXT

"SINGLE LINE FLAG NOT CLEARED BY \$OLD"

5364 0122
 5365 0504
 5366 4002
 5367 3140
 5370 2304
 5371 1404
 5372 0000
 5373 2311
 5374 1607
 5375 1405
 5376 4014
 5377 1116
 5400 0540
 5401 0614
 5402 0107
 5403 4016
 5404 1724
 5405 4003
 5406 1405
 5407 0122
 5410 0504
 5411 4002
 5412 3140
 5413 2304
 5414 2303
 5415 0000
 5416 2311
 5417 1607
 5420 1405
 5421 4014
 5422 1116
 5423 0540
 5424 0614
 5425 0107
 5426 4016
 5427 1724
 5430 4003
 5431 1405
 5432 0122
 5433 0504
 5434 4002
 5435 3140
 5436 2304
 5437 2204
 5440 0000
 5441 2311
 5442 1607
 5443 1405
 5444 4014
 5445 1116
 5446 0540
 5447 0614
 5450 0107
 5451 4003
 5452 1405

MESS16, TEXT "SINGLE LINE FLAG NOT CLEARED BY SDR0"

MESS17, TEXT "SINGLE LINE FLAG NOT CLEARED BY SDR0"

MESS18, TEXT "SINGLE LINE FLAG CLEARED BY SDST, SDS0, OR SOLC"

5453 0122
5454 0504
5455 4002
5456 3140
5457 2304
5460 2324
5461 3440
5462 2304
5463 2321
5464 3440
5465 1722
5466 4023
5467 0414
5470 0300
5471 2125
5472 0104
5473 4014
5474 1116
5475 0540
5476 0614
5477 0107
5500 4023
5501 1311
5502 2040
5503 1116
5504 2324
5505 2225
5506 0324
5507 1117
5510 1040
5511 0116
5512 0440
5513 1417
5514 0711
5515 0300
5516 2125
5517 0104
5520 4014
5521 1116
5522 0540
5523 0614
5524 0107
5525 4001
5526 1427
5527 0131
5530 2340
5531 2305
5532 2440
5533 1722
5534 4023
5535 0423
5536 2140
5537 0114
5540 2701
5541 3123

MESS19, TEXT "QUAD LINE FLAG SKIP INSTRUCTION AND LOGIC"

MESS20, TEXT "QUAD LINE FLAG ALWAYS SET OR BDSQ ALWAYS SKIPS"

5542 4023
 5543 1311
 5544 2023
 5545 0000
 5546 2125
 5547 0104
 5550 4014
 5551 1116
 5552 0540
 5553 0614
 5554 0107
 5555 4023
 5556 0324
 5557 4024
 5560 1717
 5561 4023
 5562 1717
 5563 1600
 5564 2125
 5565 0104
 5566 4014
 5567 1116
 5570 0540
 5571 0614
 5572 0107
 5573 4004
 5574 1705
 5575 2340
 5576 1617
 5577 2440
 5600 2305
 5601 2440
 5602 1722
 5603 4023
 5604 0423
 5605 2140
 5606 0417
 5607 0523
 5610 4016
 5611 1704
 5612 4023
 5613 1331
 5614 2000
 5615 2125
 5616 0104
 5617 4014
 5620 1116
 5621 0540
 5622 0614
 5623 0107
 5624 4023
 5625 1405
 5626 0122
 5627 0504
 5630 4002

MESS21, TEXT "QUAD LINE FLAG SET TOO SOON"

MESS22, TEXT "QUAD LINE FLAG DOES NOT SET OR SDSQ DOES NOT SKIP"

MESS23, TEXT "QUAD LINE FLAG CLEARED BY SDSQ"

5631	3140		
5632	2304		
5633	2321		
5634	0000		
5635	2125	MESS24, TEXT	"QUAD LINE FLAG NOT CLEARED BY CAP"
5636	0104		
5637	4014		
5640	1116		
5641	0540		
5642	0614		
5643	0107		
5644	4016		
5645	1724		
5646	4003		
5647	1405		
5650	0122		
5651	0504		
5652	4002		
5653	3140		
5654	0301		
5655	0000		
5656	2125	MESS25, TEXT	"QUAD LINE FLAG NOT CLEARED BY SOLD"
5657	0104		
5660	4014		
5661	1116		
5662	0540		
5663	0614		
5664	0107		
5665	4016		
5666	1724		
5667	4003		
5670	1405		
5671	0122		
5672	0504		
5673	4002		
5674	3140		
5675	2304		
5676	1404		
5677	0000		
5700	2125	MESS26, TEXT	"QUAD LINE FLAG NOT CLEARED BY SRCB"
5701	0104		
5702	4014		
5703	1116		
5704	0540		
5705	0614		
5706	0107		
5707	4016		
5710	1724		
5711	4003		
5712	1405		
5713	0122		
5714	0504		
5715	4002		
5716	3140		
5717	2304		

6007 1411
 6010 2057
 6011 2014
 6012 1720
 6013 4016
 6014 1724
 6015 4020
 6016 2217
 6017 2005
 6020 2214
 6021 3140
 6022 0314
 6023 0501
 6024 2205
 6025 0400

6026 2411
 6027 1511
 6030 1607
 6031 4005
 6032 2222
 6033 1722
 6034 4023
 6035 1311
 6036 2040
 6037 1116
 6040 2324
 6041 2225
 6042 0324
 6043 1117
 6044 1640
 6045 0116
 6046 0440
 6047 1417
 6050 0711
 6051 0300
 6052 2411
 6053 1511
 6054 1607
 6055 4005
 6056 2222
 6057 1722
 6060 4001
 6061 1427
 6062 0131
 6063 2340
 6064 2305
 6065 2440
 6066 1722
 6067 4023
 6070 0423
 6071 2440
 6072 0114

MESSAGE, TEXT "TIMING ERROR SKIP INSTRUCTION AND LOGIC"

MESSAGE, TEXT "TIMING ERROR ALWAYS SET OR SOST ALWAYS SKIPS"

6073 2701
 6074 3123
 6075 4023
 6076 1311
 6077 2023
 6100 0000
 6101 2411
 6102 1511
 6103 1607
 6104 4003
 6105 2222
 6106 1722
 6107 4004
 6110 1703
 6111 2340
 6112 1617
 6113 2440
 6114 2303
 6115 2440
 6116 1110
 6117 4022
 6120 0501
 6121 0440
 6122 1517
 6123 0403
 6124 4017
 6125 2240
 6126 2304
 6127 2304
 6130 4004
 6131 1703
 6132 2340
 6133 1617
 6134 2440
 6135 2313
 6136 1120
 6137 0000
 6140 2411
 6141 1511
 6142 1607
 6143 4000
 6144 2222
 6145 1722
 6146 4003
 6147 1403
 6150 0122
 6151 0504
 6152 4002
 6153 3140
 6154 2304
 6155 2324
 6156 0000
 6157 2411
 6160 1511
 6161 1607

MESS32, TEXT "TIMING ERROR DOES NOT SET IN READ MODE OR SOST DOES NOT SKIP"

MESS33, TEXT "TIMING ERROR CLEARED BY SOST"

MESS34, TEXT "TIMING ERROR NOT CLEARED BY CAF"

6162 4005
6163 2222
6164 1722
6165 4016
6166 1724
6167 4003
6170 1405
6171 0122
6172 0304
6173 4002
6174 3140
6175 0301
6176 0600
6177 2411
6200 1511
6201 1607
6202 4005
6203 2222
6204 1722
6205 4023
6206 2401
6207 2425
6210 2340
6211 0211
6212 2440
6213 1617
6214 2440
6215 2305
6216 2440
6217 1116
6220 4003
6221 1719
6222 1501
6223 1604
6224 4022
6225 0507
6226 1123
6227 2405
6230 2200
6231 2411
6232 1511
6233 1607
6234 4005
6235 2202
6236 1722
6237 4016
6240 1724
6241 4003
6242 1405
6243 0122
6244 0304
6245 4002
6246 3140
6247 2304
6250 1403

MESS35: TEXT "TIMING ERROR STATUS BIT NOT SET IN COMMAND REGISTER"

MESS36: TEXT "TIMING ERROR NOT CLEARED BY SOLC"

6251 0000
 6252 2411
 6253 1511
 6254 1607
 6255 4005
 6256 2222
 6257 1722
 6260 4004
 6261 1705
 6262 2340
 6263 1617
 6264 2440
 6265 2305
 6266 2440
 6267 1116
 6270 4027
 6271 2211
 6272 2005
 6273 4013
 6274 1704
 6275 0500
 6276 2411
 6277 1511
 6300 1607
 6301 4005
 6302 2222
 6303 1722
 6304 4004
 6305 1705
 6306 2340
 6307 1617
 6310 2440
 6311 0314
 6312 0501
 6313 2240
 6314 2722
 6315 1124
 6316 0540
 6317 0614
 6320 1100
 6321 3706
 6322 1617
 6323 2000
 6324 2411
 6325 1511
 6326 1607
 6327 4005
 6330 2222
 6331 1722
 6332 4016
 6333 1724
 6334 4023
 6335 0524
 6336 4002
 6337 3140

MESS37, TEXT "TIMING ERROR DOES NOT SET IN WRITE MODE"

MESS38, TEXT "TIMING ERROR DOES NOT CLEAR WRITE FLIP/FLOP"

MESS39, TEXT "TIMING ERROR NOT SET BY SRC BORO, OR SOLC"

6340 2304
6341 2203
6342 4023
6343 0422
6344 0434
6345 4017
6346 2240
6347 2304
6350 1403
6351 0000
6352 2520
6353 4024
6354 1740
6355 2320
6356 0505
6357 0440
6360 0311
6361 2203
6362 2511
6363 2422
6364 3140
6365 0310
6366 0503
6367 1340
6370 2523
6371 1116
6372 0740
6373 4703
6374 1405
6375 0122
6376 4015
6377 0122
6400 1340
6401 2422
6402 0103
6403 1340
6404 2205
6405 0711
6406 2324
6407 0522
6410 4740
6411 0605
6412 0124
6413 2522
6414 0500
6415 1501
6416 2213
6417 4024
6420 2201
6421 0313
6422 4022
6423 0507
6424 1123
6425 2405
6426 2240

MESS43, TEXT "UP TO SPEED CIRCUITRY CHECK USING 'CLEAR MARK TRACK REGISTER' FEATURE"

MESS44, TEXT "MARK TRACK REGISTER NOT CLEARED BY 'GO'"

6427 1617
6430 2440
6431 0314
6432 0501
6433 2205
6434 0440
6435 0231
6436 4047
6437 0717
6440 4700
6441 1501
6442 2213
6443 4024
6444 2201
6445 0313
6446 4022
6447 0507
6450 1123
6451 2405
6452 2240
6453 1617
6454 2440
6455 0314
6456 0501
6457 2205
6460 0440
6461 0231
6462 4047
6463 2324
6464 1720
6465 4700
6466 1501
6467 2213
6470 4024
6471 2201
6472 0313
6473 4022
6474 0507
6475 1123
6476 2405
6477 2240
6500 1617
6501 2440
6502 0314
6503 0501
6504 2205
6505 0440
6506 0231
6507 4047
6510 2205
6511 2647
6512 4024
6513 1740
6514 4706
6515 2704

MESS45: TEXT

"MARK TRACK REGISTER NOT CLEARED BY /STOP/"

MESS46: TEXT

"MARK TRACK REGISTER NOT CLEARED BY /REV/ TO /FWD/"

6516 4700
 6517 1501
 6520 2213
 6521 4024
 6522 2201
 6523 0313
 6524 4022
 6525 0507
 6526 1123
 6527 2405
 6530 2240
 6531 1617
 6532 2440
 6533 0314
 6534 0501
 6535 2205
 6536 0440
 6537 0231
 6540 4047
 6541 0627
 6542 0447
 6543 4024
 6544 1740
 6545 4722
 6546 0526
 6547 4700
 6550 1501
 6551 2213
 6552 4024
 6553 2201
 6554 0313
 6555 4022
 6556 0507
 6557 1123
 6560 2405
 6561 2240
 6562 1617
 6563 2440
 6564 0314
 6565 0501
 6566 2205
 6567 0440
 6570 0231
 6571 4047
 6572 2516
 6573 1124
 6574 4060
 6575 4740
 6576 2417
 6577 4047
 6600 2516
 6601 1124
 6602 4061
 6603 4700
 6604 1501

MESS47, TEXT "MARK TRACK REGISTER NOT CLEARED BY 'FWD' TO 'REV'"

MESS48, TEXT "MARK TRACK REGISTER NOT CLEARED BY 'UNIT 0' TO 'UNIT 1'"

MESS49, TEXT "MARK TRACK REGISTER NOT CLEARED BY 'UNIT 1' TO 'UNIT 0'"

6605 2213
6606 4024
6607 2201
6610 0313
6611 4022
6612 0507
6613 1123
6614 2403
6615 2240
6616 1617
6617 2440
6620 0314
6621 0501
6622 2203
6623 0440
6624 0231
6625 4047
6626 2316
6627 1124
6630 4001
6631 4740
6632 2417
6633 4047
6634 2316
6635 1124
6636 4000
6637 4700
6640 4723
6641 0514
6642 0503
6643 2440
6644 0503
6645 2217
6646 2247
6647 4004
6650 0503
6651 2403
6652 0000
6653 2411
6654 1011
6655 1607
6656 4000
6657 2222
6660 1722
6661 4023
6662 0524
6663 0000
6664 1617
6665 4023
6666 0514
6667 0503
6670 2440
6671 0522
6672 2217
6673 2240

MESSAGE, TEXT "SELECT ERROR! TESTS"

MESSAGE1, TEXT "TIMING ERROR SET"

MESSAGE2, TEXT "NO SELECT ERROR STATUS FROM UNIT I"

6674 2324
6675 0124
6676 2523
6677 4006
6700 2217
6701 1540
6702 2516
6703 1124
6704 4061
6705 0000
6706 4727
6707 2211
6710 2405
6711 4740
6712 1617
6713 2440
6714 0314
6715 0501
6716 2205
6717 0440
6720 0231
6721 4023
6722 0514
6723 0503
6724 2440
6725 0522
6726 2217
6727 2200
6730 4727
6731 2211
6732 2405
6733 4014
6734 1703
6735 1340
6736 1725
6737 2447
6740 4024
6741 0523
6742 2423
6743 0000
6744 1617
6745 4027
6746 2211
6747 2405
6750 4014
6751 1703
6752 1340
6753 1725
6754 2440
6755 2324
6756 0124
6757 2523
6760 4006
6761 2617
6762 1340

MESS53, TEXT "/WRITE' NOT CLEARED BY SELECT ERROR"

MESS54, TEXT "/WRITE LOCK OUT' TESTS"

MESS55, TEXT "NO WRITE LOCK OUT STATUS FROM UNIT 0"

6763 2516
6764 1124
6765 4060
6766 0000
6767 4727
6770 2211
6771 2405
6772 4740
6773 1617
6774 2440
6775 0314
6776 0501
6777 2205
7000 0440
7001 0331
7002 4027
7003 2211
7004 2405
7005 4014
7006 1703
7007 1340
7010 1725
7011 2400
7012 0401
7013 2401
7014 4027
7015 2211
7016 2424
7017 0516
7020 4006
7021 1722
7022 2701
7023 2204
7024 0000
7025 2722
7026 1124
7027 0540
7030 0401
7031 2401
7032 4006
7033 1722
7034 2701
7035 2204
7036 0000
7037 2205
7040 0104
7041 4004
7042 0124
7043 0140
7044 0617
7045 2227
7046 0122
7047 0400
7050 2205
7051 0104

MESS56, TEXT "WRITE' NOT CLEARED BY WRITE LOCK OUT"

MESS57, TEXT "DATA WRITTEN FORWARD"

MESS58, TEXT "WRITE DATA FORWARD"

MESS59, TEXT "READ DATA FORWARD"

MESS60, TEXT "READ DATA BACKWARD"

7052 4004
 7053 0124
 7054 0140
 7055 0201
 7056 0313
 7057 2701
 7060 2204
 7061 0000
 7062 0401
 7063 2401
 7064 4027
 7065 2211
 7066 2424
 7067 0516
 7070 4002
 7071 0103
 7072 1327
 7073 0122
 7074 0400
 7075 2722
 7076 1124
 7077 0540
 7100 0401
 7101 2401
 7102 4002
 7103 0103
 7104 1327
 7105 0122
 7106 0400
 7107 1617
 7110 4029
 7111 1611
 7112 2440
 7113 6040
 7114 2305
 7115 1405
 7116 0324
 7117 0504
 7120 0000

MESS61, TEXT "DATA WRITTEN BACKWARD"

MESS62, TEXT "WRITE DATA BACKWARD"

MESS63, TEXT "NO UNIT 0 SELECTED"

7200

PAGE
 /ROUTINE TO CHANGE ALL T08E JOYS IN PROGRAM FOR MULTIPLE UNIT
 /ROUTINE IS STARTED AT LOCATION "MODIFY" WITH AC6,7 AND 8 INDICATING
 /DEVICE SELECTOR BITS 6, 7, AND 8 OF THE CONTROL TO BE TESTED (4,5,6, OR 7)

7200 7604
 7201 0221
 7202 3223
 7203 1230
 7204 3224
 7205 1227
 7206 3226
 7207 1624

MODIFY, LAR
 AND MODMS1
 DCA MODCON
 TAD MODTAB
 DCA MOOPT1
 TAD MODSIE
 DCA MODCNT
 MODLUP, TAD I MOOPT1

7210	3225	DCA	MODPT2
7211	1625	TAD I	MODPT2
7212	0222	AND	MODMS2
7213	1223	TAD	MODCON
7214	3625	DCA I	MODPT2
7215	2224	ISE	MODPT1
7216	2226	ISE	MODCNT
7217	5207	JMP	MODLUP
7220	7402	HLT	
7221	0030	MODMS1,	30
7222	7747	MODMS2,	7747
7223	0000	MODCON,	0
7224	0000	MODPT1,	0
7225	0000	MODPT2,	0
7226	0000	MODCNT,	0
7227	7441	MODSIZE,	=MOOEND+MODTAB
7230	7231	MODTAB,	1+1
7231	0207		IOT1
7232	0212		IOT2
7233	0214		IOT3
7234	0244		IOT4
7235	0246		IOT5
7236	0313		IOT6
7237	0320		IOT7
7240	0406		IOT8
7241	0423		IOT9
7242	0425		IOT10
7243	0442		IOT11
7244	0457		IOT12
7245	0461		IOT13
7246	0600		IOT14
7247	0612		IOT15
7250	0621		IOT16
7251	0656		SING1
7252	0649		SING2
7253	0653		IOT17
7254	0662		IOT18
7255	0665		SING3
7256	0670		IOT19
7257	0674		IOT20
7260	0704		SING4
7261	0707		IOT21
7262	0713		IOT22
7263	0723		SING5
7264	0726		IOT23
7265	0732		IOT24
7266	1002		SING6
7267	1004		IOT25
7270	1006		IOT26
7271	1011		IOT27
7272	1012		IOT28
7273	1032		IOT29
7274	1036		IOT30

7275	1041	QUAD0
7276	1042	IOT31
7277	1044	IOT32
7300	1050	IOT33
7301	1053	IOT34
7302	1057	IOT35
7303	1061	IOT36
7304	1067	IOT37
7305	1072	IOT38
7306	1101	QUAD1
7307	1110	QUAD2
7310	1116	IOT39
7311	1125	IOT40
7312	1130	QUAD3
7313	1133	IOT41
7314	1137	IOT42
7315	1202	QUAD4
7316	1205	IOT43
7317	1211	IOT44
7320	1221	QUAD5
7321	1224	IOT45
7322	1230	IOT46
7323	1240	QUAD6
7324	1242	IOT47
7325	1244	IOT48
7326	1250	IOT49
7327	1254	IOT50
7330	1270	IOT51
7331	1271	IOT52
7332	1273	IOT53
7333	1276	IOT54
7334	1305	IOT55
7335	1324	IOT56
7336	1325	IOT57
7337	1327	IOT58
7340	1334	TIME0
7341	1342	IOT59
7342	1353	TIME1
7343	1402	TIME2
7344	1410	IOT60
7345	1417	IOT61
7346	1422	TIME3
7347	1424	IOT62
7350	1442	IOT63
7351	1445	TIME4
7352	1447	IOT64
7353	1453	IOT65
7354	1465	IOT66
7355	1466	IOT67
7356	1470	IOT68
7357	1475	IOT69
7360	1500	IOT70
7361	1504	IOT71

7362	1510	I0T72
7363	1523	I0T73
7364	1524	I0T74
7365	1531	I0T75
7366	1532	I0T76
7367	1533	I0T77
7370	1543	I0T78
7371	1605	I0T82
7372	1607	I0T83
7373	1613	I0T84
7374	1626	I0T85
7375	1627	I0T86
7376	1631	I0T87
7377	1635	I0T88
7400	1641	I0T89
7401	1654	I0T90
7402	1655	I0T91
7403	1657	I0T92
7404	1664	I0T93
7405	1670	I0T94
7406	1703	I0T95
7407	1704	I0T96
7410	1706	I0T97
7411	1713	I0T98
7412	1717	I0T99
7413	2004	I0T100
7414	2005	I0T101

7415	2007	I0T102
7416	2014	I0T103
7417	2020	I0T104
7420	2037	I0T105
7421	2040	I0T106
7422	2042	I0T107
7423	2047	I0T108
7424	2053	I0T109
7425	2049	I0T110
7426	2204	I0T111
7427	2206	DISLUP
7430	2211	I0T112
7431	2213	I0T113
7432	2227	I0T114
7433	2241	I0T115
7434	2253	I0T116
7435	2402	I0T117
7436	2410	I0T118
7437	2413	I0T119
7440	2510	I0T120
7441	2512	I0T121
7442	2520	I0T122
7443	2522	I0T123
7444	2605	I0T124
7445	2606	I0T125
7446	2614	I0T126

7447	2616	IOT127
7450	2635	IOT128
7451	2641	IOT129
7452	2654	SELECT2
7453	2660	IOT130
7454	2701	IOT131
7455	2720	IOT132
7456	2724	IOT133
7457	3606	RSRCH0
7460	3607	IOT134

7461	3615	RSRCH1
7462	3617	IOT135
7463	3620	IOT136
7464	3632	IOT13A
7465	3642	IOT137
7466	3622	IOT138
7467	3640	RSRCH2
7470	3654	RSTURN
7471	4056	IOT139
7472	4112	IOT140
7473	4114	IOT141
7474	4201	IOT142
7475	4203	IOT143
7476	4246	IOT144
7477	4250	IOT145
7500	4406	SRCH0
7501	4407	IOT146
7502	4415	SRCH1
7503	4417	IOT147
7504	4420	IOT148
7505	4422	IOT149
7506	4432	IOT14A
7507	4442	IOT150
7510	4454	IOT150
7511	4440	SRCH2
7512	3733	IOT151
7513	3734	IOT15A
7514	4514	IOT152
7515	4516	IOT153
7516	4274	IOT154
7517	4540	IOT155
7520	4542	IOT156
7521	4646	IOT157
7522	4651	IOT158
7523	4653	IOT159
7524	4662	IOT160
7525	4614	IOT16A
7526	4612	READ1
7527	4664	IOT161
7530	4674	IOT162
7531	4676	IOT163
7532	4702	IOT164
7533	4704	IOT165

7534	4710	10T166
7535	4712	10T167
7536	4272	10T168
7537	4277	10T169
7540	4301	10T170
7541	2110	10T171
7542	0064	10T172
7543	0071	10T173
7544	0073	10T174
7545	0102	10T175
7546	0104	10T176
7547	0106	10T177

MOBEND,

S

0160	4110
0161	4245
0162	4200
0163	4225
0164	2701
0165	1000
0166	3070
0167	6777
0170	7752
0171	7756
0172	0077
0173	4707
0174	2234
0175	3000
0176	0212
0177	0215

0000	00000000	00000001	11111111	11111111	11111111	11111111	11111111	11111111
0100	11111111	11110000	00000000	00000000	00000000	00000000	11111111	11111111
0200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0300	11111111	11111111	11111111	11100000	00000000	00000000	00000011	11111111
0400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0500	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
0700	11111111	11111111	11111111	11111111	10000000	00000000	00000111	11111111
1000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1100	11111111	11111111	11111111	11111111	11111100	00000000	00001111	11111111
1200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1300	11111111	11111111	11111111	11111111	11111111	11111111	10011111	11111111
1400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1500	11111111	11111111	11111111	11111111	11111111	11100001	11111111	11111111
1600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
1700	11111111	11111111	11111110	00000000	00000000	00000000	00000011	11111111
2000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11100000
2100	11111111	11111111	11111111	11111111	11111111	11111000	00001111	11111111
2200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2300	11111111	11111111	11111111	11111111	11000000	00000000	00000111	11111111
2400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2500	11111111	11111111	11111110	00000000	00000000	00000000	00111111	11111111
2600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
2700	11111111	11111111	11111111	11111111	11111100	00000000	11111111	11111111
3000	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3100	11111111	11111111	11111111	11111111	11110111	11111111	11111111	11111111
3200	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3300	11111111	11111111	11111111	11000000	00000000	00000000	00000000	00000111
3400	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3500	11111111	11111111	11100000	00000000	00000000	00000000	00000000	00001111
3600	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
3700	11111111	11111111	11111111	11111111	10000000	00001111	11111111	11111111

AUTO	0017	DISEND	2224	IOT102	2007	IOT149	4422
BADBLK	4461	DISLUP	2206	IOT103	2014	IOT14A	4432
BLK	0027	DISTRK	2235	IOT104	2020	IOT15	0612
BLKCH	2100	EHLT	0530	IOT105	2037	IOT150	4442
BLKCHK	2400	ENDZ	2404	IOT106	2040	IOT151	3733
BLKCN	0060	ERR1HT	0525	IOT107	2042	IOT152	4514
BLKEND	0070	ERR2HT	0556	IOT108	2047	IOT153	4516
BLKERR	2146	ERROR1	0496	IOT109	2053	IOT154	4274
BLKMRK	2426	ERROR2	0537	IOT11	0442	IOT155	4540
BLKREV	0061	FBLKCT	2113	IOT110	2065	IOT156	4542
BLKSER	0101	FILDCT	3440	IOT111	2204	IOT157	4646
BLKTRY	0055	FILDOO	3442	IOT112	2211	IOT158	4651
BMESS	4105	FILDEC	3421	IOT113	2213	IOT159	4653
BUFF1	7200	FILDPT	3441	IOT114	2227	IOT15A	3734
BUFF2	7400	FILICT	3416	IOT115	2241	IOT15B	4454
CAF	6007	FILIDT	3420	IOT116	2253	IOT16	0621
CCNTR	3224	FILINC	3400	IOT117	2402	IOT160	4662
CHKCHK	4323	FILIPT	3417	IOT118	2410	IOT161	4664
CHKCLA	0400	FILL0	3227	IOT119	2413	IOT162	4674
CHKDAT	4131	FILL0C	3241	IOT12	0457	IOT163	4676
CHKERR	4110	FILL0P	3242	IOT120	2510	IOT164	4702
CHKHLT	4126	FILL1	3243	IOT121	2512	IOT165	4704
CHKMES	4132	FILL1C	3206	IOT122	2520	IOT166	4710
CHKSUM	4744	FILL1P	3257	IOT123	2522	IOT167	4712
CLOOP	0266	FILL2S	3260	IOT124	2605	IOT168	4272
CLRWT	4271	FILL2C	3293	IOT125	2606	IOT169	4277
CNTR1	0022	FILL2K	3294	IOT126	2614	IOT16A	4614
CNTR2	0023	FILL2P	3295	IOT127	2616	IOT17	0653
COMLUP	3212	FILLC1	3323	IOT128	2635	IOT170	4301
COMP	3516	FILLC2	3324	IOT129	2641	IOT171	2110
COMPAR	3200	FILLP1	3321	IOT13	0461	IOT172	0044
COMREG	0236	FILLP2	3322	IOT130	2660	IOT173	0071
CRLF	0040	FILPAT	3296	IOT131	2701	IOT174	0073
CSDLC	0405	FILPL1	3303	IOT132	2720	IOT175	0102
CSDLD	0441	FILPL2	3307	IOT133	2724	IOT176	0104
CSDRC	0422	FILPNT	0000	IOT134	3607	IOT177	0106
CSDRD	0456	FILTC	3351	IOT135	3617	IOT18	0642
CSUMRT	4303	FILTP	3325	IOT136	3620	IOT19	0670
DATA	2452	FORMT1	0532	IOT137	3642	IOT2	0212
DATERR	4000	FWDEXP	2402	IOT138	3622	IOT20	0674
DATHO	4040	GOOD	0024	IOT139	4056	IOT21	0707
DATHLT	4036	GNTR	3225	IOT13A	3052	IOT22	0713
DATLUP	0232	GUARD	2492	IOT14	0606	IOT23	0726
DATMES	4042	HEAD1	0025	IOT140	4112	IOT24	0732
DATNUM	4020	HEAD2	0026	IOT141	4114	IOT25	1004
DATPNT	4041	HEADTP	4000	IOT142	4201	IOT26	1006
DATREG	0201	IN	0021	IOT143	4203	IOT27	1011
DBLOCK	2200	INITST	0305	IOT144	4246	IOT28	1012
DISBL	0056	IOT1	0207	IOT145	4250	IOT29	1032
DISBLK	2233	IOT10	0425	IOT146	4407	IOT3	0214
DISDA	0057	IOT100	2004	IOT147	4417	IOT30	1036
DISDAT	2236	IOT101	2005	IOT148	4420	IOT31	1042

10732 1044
 10733 1050
 10734 1053
 10735 1057
 10736 1061
 10737 1067
 10738 1072
 10739 1116
 1074 0244
 10740 1125
 10741 1133
 10742 1137
 10743 1205
 10744 1211
 10745 1224
 10746 1238
 10747 1242
 10748 1244
 10749 1250
 1075 0246
 10750 1254
 10751 1278
 10752 1271
 10753 1273
 10754 1276
 10755 1305
 10756 1324
 10757 1325
 10758 1327
 10759 1342
 1076 0313
 10760 1410
 10761 1417
 10762 1424
 10763 1442
 10764 1447
 10765 1453
 10766 1468
 10767 1466
 10768 1478
 10769 1478
 1077 0320
 10770 1500
 10771 1504
 10772 1510
 10773 1523
 10774 1524
 10775 1531
 10776 1532
 10777 1533
 10778 1543
 1078 0406

10782 1605
 10783 1607
 10784 1613
 10785 1626
 10786 1627
 10787 1631
 10788 1635
 10789 1641
 1079 0423
 10790 1654
 10791 1655
 10792 1657
 10793 1664
 10794 1670
 10795 1703
 10796 1704
 10797 1706
 10798 1713
 10799 1717
 LOCKED 4465
 LOCK 2445
 LOOP1 8846
 M0300A 4785
 M10 5854
 MESSAGE 2264
 MESS01 5000
 MESS010 5194
 MESS011 5222
 MESS012 5253
 MESS013 5355
 MESS014 5386
 MESS015 5388
 MESS016 5393
 MESS017 5416
 MESS018 5441
 MESS019 5471
 MESS02 5821
 MESS020 5516
 MESS021 5546
 MESS022 5564
 MESS023 5615
 MESS024 5685
 MESS025 5696
 MESS026 5700
 MESS027 5722
 MESS028 5744
 MESS029 5773
 MESS3 5844
 MESS30 6026
 MESS31 6052
 MESS32 6101
 MESS33 6140

MESS34 6197
 MESS35 6197
 MESS36 6231
 MESS37 6292
 MESS38 6296
 MESS39 6324
 MESS4 5854
 MESS43 6352
 MESS44 6415
 MESS45 6441
 MESS46 6446
 MESS47 6517
 MESS48 6550
 MESS49 6604
 MESS5 5876
 MESS50 6640
 MESS51 6653
 MESS52 6664
 MESS53 6706
 MESS54 6738
 MESS55 6744
 MESS56 6767
 MESS57 7012
 MESS58 7029
 MESS59 7057
 MESS6 5103
 MESS60 7050
 MESS61 7062
 MESS62 7075
 MESS63 7107
 MESS7 5136
 MESS8 5151
 MESS9 5161
 MODCNT 7226
 MODCON 7223
 MODEND 7547
 MODIFY 7200
 MODLUP 7207
 MODMS1 7221
 MODMS2 7222
 MODPT1 7224
 MODPT2 7225
 MODSTR 7227
 MODTAB 7230
 MPNTR 2315
 OCNT 2341
 OK 2744
 ONUMB 2340
 OPLoop 2324
 OPRINT 2316
 OUT 0020
 PASCNT 3497

PASS 3513
 PREFIN 2462
 QOLUP 1074
 QUAD 1024
 QUAD0 1041
 QUAD1 1101
 QUAD2 1110
 QUAD3 1130
 QUAD4 1202
 QUAD5 1221
 QUAD6 1240
 QUAD7 1265
 RADDR 4657
 RBLKCT 2132
 RCNT 4656
 RCOUNT 4660
 RD1MRK 2507
 RD6MRK 2515
 RDQUAD 4707
 READ 4600
 READ1 4612
 READ2 4632
 READR 3714
 REVBLK 2476
 REVEXP 2552
 REVGRD 4641
 RLOCED 3663
 ROCK 2237
 RSERCH 3600
 RSLOOK 3670
 RSRCH0 3606
 RSRCH1 3615
 RSRCH2 3640
 RSTURN 3634
 RVGARD 2456
 SBXOR 4714
 SBWORD 4745
 SOLC 6774
 SOLD 6773
 SDRC 6776
 SORD 6777
 SOS0 6773
 SOS5 6771
 SOST 6772
 SEARCH 4400
 SELECT1 2634
 SELECT2 2634
 SELECT 2600
 SELERR 4225
 SELHLT 4234
 SELMES 4236
 SING1 0636

SING2	0645	WRITE2	4521
SING3	0665	WRITER	3671
SING4	0704	WROERR	4200
SING5	0723	WROHLT	4212
SING6	1002	WRQUAD	4701
SING7	1021	WRTLCK	3731
SINGL0	0615	WRZL1	3035
SINGLE	0600	WRZL2	3076
SLOOK	4470	XFER	3000
SP1CT	3456	XXX	4322
SP1PT	3457		
SP2CT	3474		
SP2PT	3475		
SPEC1	3443		
SPEC10	3460		
SPEC2	3461		
SPEC20	3476		
SRCH0	4406		
SRCH1	4415		
SRCH2	4440		
SUNIT	4471		
TIME0	1334		
TIME1	1353		
TIME2	1402		
TIME3	1422		
TIME4	1445		
TIME5	1463		
TIME6	1521		
TIMING	1315		
TPNTR	3226		
TYMERR	4245		
TYMHLT	4260		
TYMMES	4262		
TYPE	0031		
UMESS	4102		
UNIT	2234		
UTSMK1	1624		
UTSMK2	1652		
UTSMK3	1701		
UTSMK4	2002		
UTSMK5	2035		
UTSMK6	2062		
UTSMRK	1600		
WADDR	4544		
WCNT	4470		
WCOUNT	4545		
WL0	2673		
WL1	2717		
WRDMES	4214		
WREAD	3024		
WRITE	4472		
WRITE1	4512		

ERRORS DETECTED: 0

LINKS GENERATED: 147

RUN-TIME: 21 SECONDS

3K CORE USED

