

TS11

TS11 CTRL LGC
CZTSIB0

AH E458B MC

COPYRIGHT 1979

FICHE 1 OF 1

NOV 1979

digital

MADE IN USA

The main body of the document is a large, dark grid of data points or a very faded table, likely representing a control logic or data matrix for the TS11 system. The grid is composed of many small, rectangular cells, each containing faint, illegible text or symbols. The overall appearance is that of a high-resolution scan of a physical document, where the individual characters are too light to be read clearly against the dark background.

IDENTIFICATION

PRODUCT CODE: AC-E457B-MC
PRODUCT NAME: CZTS1B0 TS11 CTRL LGC
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: ROBERT F. WERY

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

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

COPYRIGHT (C) 1979 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

USER DOCUMENTATION

USER DOCUMENTATION TABLE OF CONTENTS

GLOSSARY

- 1.0 GENERAL INFORMATION
 - 1.1 PROGRAM ABSTRACT
 - 1.1.1 FUNCTIONAL DESCRIPTION
 - 1.1.2 STRUCTURE OF PROGRAM
 - 1.1.3 MEMORY MAP
 - 1.1.4 DIAGNOSTIC INFORMATION
 - 1.2 SYSTEM REQUIREMENTS
 - 1.2.1 HARDWARE REQUIREMENTS
 - 1.2.2 SOFTWARE REQUIREMENTS
 - 1.3 RELATED DOCUMENTS AND STANDARDS
 - 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 - 1.5 ASSUMPTIONS
 - 1.6 DIAGNOSTIC HISTORY
- 2.0 OPERATING INSTRUCTIONS
 - 2.1 HARDWARE PARAMETERS
 - 2.2 SOFTWARE PARAMETERS
 - 2.4 EXECUTION TIMES
 - 2.4.1 SYSTEM CONFIGURATION
 - 2.4.2 TEST EXECUTION TIMES
- 3.0 ERROR INFORMATION
 - 3.1 ERROR REPORTING
 - 3.2 ERROR HALTS
- 4.0 DEVICE INFORMATION TABLES
 - 4.1 DIAG REGISTERS
 - 4.2 GENERAL
 - 4.3 UNIBUS INTERFACE SPECIFICATIONS
 - 4.4 BIT DEFINITIONS FOR TS11/TS04 REGISTERS

- 5.0 TEST SUMMARIES
- 5.1 TEST 1 - PDP-11/TS11 WRAP TEST
- 5.2 TEST 2 - PDP-11/TS04 WRAP TEST
- 5.3 TEST 3 - SET TS04 CHARACTERISTIC
- 5.4 FORMATTER BOARD DATA WRAP TESTS (TESTS 4-7).
 - 5.4.1 TRACK ACTIVE/TRACK INACTIVE TESTS (TEST 4)
 - 5.4.1.1 SUBTEST 1 - TRACK INACTIVE TEST
 - 5.4.1.2 SUBTEST 2 - TRACK ACTIVE TEST
 - 5.4.1.3 SUBTEST 3 - TRACK INACTIVE TEST
 - 5.4.1.4 SUBTEST 4 - TRACK ACTIVE TEST
 - 5.4.2 P.E. DATA TESTS (TEST 5)
 - 5.4.2.1 SUBTEST 1 - P.E. DATA TEST/0 PATTERN
 - 5.4.2.2 SUBTEST 2 - P.E. DATA TEST/1 PATTERN
 - 5.4.2.3 SUBTEST 3 - P.E. DATA TEST/SHIFTING 1 PATTERN
 - 5.4.2.4 SUBTEST 4 - P.E. DATA TEST/SHIFTING 0 PATTERN
 - 5.4.3 PE SKEW TEST (TEST 6)
 - 5.4.3.1 SUBTEST 1 - P.E. SKEW TEST
 - 5.4.3.2 SUBTEST 2 - P.E. SKEW TEST
 - 5.4.4 DEAD TRACK LOGIC TEST (TEST 7)
 - 5.4.4.1 SUBTEST 1 - P.E. DEAD TRACK TEST
 - 5.4.4.2 SUBTEST 2 - P.E. DEAD TRACK TEST
- 5.5 TEST 8 - ROM LOOKUP TABLE TEST.
- 5.6 TEST 9 - INLINE MICRO DIAG TEST
- 5.7 TEST 10 - INIT MICRO DIAG TEST

GLOSSARY

SEQ 0004

ACT	AUTOMATED COMPUTER TEST
APT	AUTOMATED PRODUCT TEST SYSTEM USED IN MANUFACTURING.
BYTE/RECORD/FILE COUNT BRF	IS STORED IN THE 4TH WORD OF THE COMMAND PACKET AND IT'S USE BY THE TS04 DEPENDS ON THE TYPE OF COMMAND.
CMD	TS04 COMMAND
COMMAND PACKET CMDPKT	FOUR WORD PACKET IN THE CPU MEMORY WHICH CONTAINS ALL INFORMATION NEEDED BY THE TS04 TO EXECUTE A COMMAND.
EXTENDED STATUS	FOUR WORDS OF TS04 STATUS WHICH ARE TRANSFERRED AS PART OF THE MESSAGE PACKET AT THE COMPLETION OF A COMMAND.
MESSAGE PACKET	SEVEN WORD PACKET IN THE CPU MEMORY INTO WHICH THE TS04 STORES STATUS AT THE COMPLETION OF A COMMAND.
PC	PROGRAM COUNTER
PSW	PROCESSOR STATUS WORD
RESIDUAL FRAME COUNT RFC	THIS COUNT IS PART OF THE MESSAGE PACKET AND CONTAINS THE NUMBER OF BYTES/RECORDS /FILES REMAINING TO BE PROCESSED AT THE COMPLETION OF A COMMAND.
TERMINATION CLASS CODE TCC	THREE BIT CODE IN THE TSSR WHICH INDICATES THE TYPE OF COMMAND TERMINATION.
TSBA	TAPE SYSTEM BUS ADDRESS REGISTER.
TSDB	TAPE SYSTEM DATA BUFFER REGISTER.
TSSR	TAPE SYSTEM STATUS REGISTER.
XST0	EXTENDED STATUS REGISTER 0
XST1	EXTENDED STATUS REGISTER 1
XST2	EXTENDED STATUS REGISTER 2
XST3	EXTENDED STATUS REGISTER 3
XXDP+	XXDP+ IS A "CATCH-ALL" NAME FOR A GROUP OF PDP-11 DIAGNOSTIC PACKAGES AVAILABLE ON MULTIMEDIA.

1.0 GENERAL INFORMATION

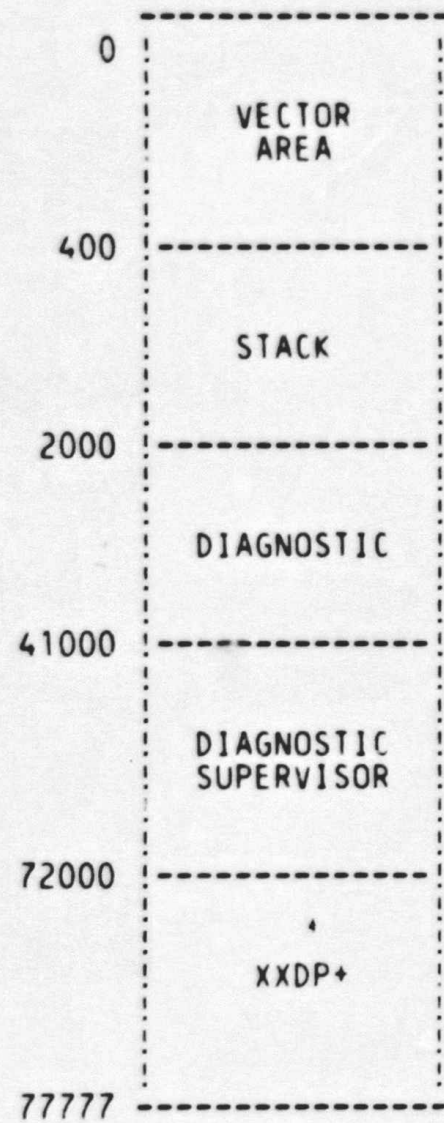
1.1 PROGRAM ABSTRACT

1.1.1 FUNCTIONAL DESCRIPTION

THIS PROGRAM PERFORMS CORRECTIVE MAINTENANCE BY EXECUTING TS11 AND TS04 WRAPAROUNDS FOR THE PURPOSE OF IDENTIFYING FAILED MODULES.

1.1.2 STRUCTURE OF PROGRAM

THIS DIAG IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE USER, BUT IT CONTAINS A CONTROL MODULE RELEASED INDEPENDENTLY AS A DIAG SUPERVISOR.

1.1.3 MEMORY MAP

FREE MEMO SPACE RESIDES BETWEEN THE DIAG AND THE SUPERVISOR.

1.1.4 DIAGNOSTIC INFORMATION

THIS DIAGNOSTIC TESTS ONE UNIT AT A TIME, BUT WILL SEQUENTIALLY TEST UP TO 4 UNITS. THE 4 UNITS ARE ASSIGNED LOGICAL UNIT NUMBERS 0 - 3 BY THE DIAGNOSTIC. THE UNITS DO NOT HAVE TO BE ON LINE AND A TAPE DOES NOT HAVE TO BE LOADED TO RUN THIS DIAGNOSTIC.

RECOMMANDATION:

IT IS RECOMMENDED TO RUN THIS DIAG WITH NO TAPE LOADED BECAUSE OF MUCH FASTER EXECUTION TIME (SEE 2.4).

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

PDP-11 PROCESSOR WITH 16K OR MORE OF MEMORY
CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
PROGRAM LOAD DEVICE

1.2.2 SOFTWARE REQUIREMENTS

DIAGNOSTIC SUPERVISOR

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP+ USERS MANUAL MD-11-CHQUS
DIAGNOSTIC SUPERVISOR PROGRAM LISTING
PDP-11 DIAGNOSTIC SUPERVISOR INTERFACE SPECIFICATION.
PDP-11 DIAGNOSTIC SUPERVISOR PROGRAMMER'S GUIDE
TS11/TS04 PROGRAMMING SPECIFICATION.
TS04/TS11 ENGINEERING SPECIFICATION.
TS11/TS04 COMMAND PACKET SPECIFICATION.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THIS PROGRAM SHOULD BE RUN BEFORE ANY OTHER HOST CPU DIAGNOSTIC PROGRAM.

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DO NOT FUNCTION PROPERLY.

1.6 DIAGNOSTIC HISTORY

REV A - FEB 79
- ORIGINAL RELEASE
REV B - AUG 79
-ADDED AN INTERRUPT TEST
-REVISED ALL TESTS BECAUSE OF AN HARDWARE ECO:
WATCH FOR TAPE MOTION DURING A LOAD SEQUENCE RESULTING FROM AN INIT SENT TO THE TS11.
-CONVERTED DIAG TO SUPERVISOR REV: C.
ADDED SEVERAL SECTIONS: PROTECT TABLE, AUTO-DROP CODE, HARD CODED PARAMETER TABLE...

2.0 OPERATING INSTRUCTIONS

SEQ 0008

2.1 HARDWARE PARAMETERS

ON A START COMMAND, THE DIAG ASKS: "CHANGE HW?".
ON A "N" ANSWER, THERE SHALL BE NO HARDWARE DIALOGUE AND THE
DIAG SHALL RUN ASSUMING A UNIT AT TSSR = 177522 WITH VECTOR OF
224.

ON A "Y" ANSWER TO "CHANGE HW?" QUESTION, THEN AND ONLY THEN,
THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE
VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT
VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

TSSR ADDRESS (172522) ?

VECTOR (224) ?

THE VALIDITY OF THESE PARAMETERS CAN BE CHECKED BEFORE RUNNING
THE TESTS BY SETTING THE FLAG "ADR" ON A STA, RES, OR CON COMMAND.
THE SO CALLED AUTO-DROP CODE SHALL THEN BE EXECUTED AFTER THE INIT
CODE AND BEFORE THE HARDWARE TESTS ARE RUN.
THAT CODE FIRST TESTS THE ADDRESS OF THE TSSR. IF NO RESPONSE FROM
THE INTERFACE, THE UNIT IS DROPPED IMMEDIATELY WITH THE FOLLOWING
MESSAGE:

BUS TRAP AT XXXXXX
INTERFACE BAD OR NOT SETTO ABOVE AD

(XXXXXX = TSSR AD)

ON A RESPONSE FROM THE INTERFACE, THE READY STATUS OF THE
UNIT IS CHECKED. IF NOT READY, THE UNIT IS DROPPED IMMEDIATELY.

2.2 SOFTWARE PARAMETERS

THE FOLLOWING QUESTION WILL BE ASKED ON REQUEST - I.E., ANSWERING
"Y" TO "CHANGE SW (L)?" QUESTION - ON A START, RESTART OR CONTINUE COMMAND:

ENABLE DATA COMPARE ERROR PRINTS FOR TESTS 4-7(L) N ?

IF "Y" IS THE RESPONSE TO THIS SOFTWARE QUESTION, THEN WILL DATA
COMPARE ERRORS BE PRINTED, PROVIDED IER FLAG IS RESET.

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION

PDP11/70
MOS MEMORY
LA36
TS11/TS04

2.4.2 TESTS EXECUTION TIMES

TAPE NOT LOADED:	40	SEC
TAPE LOADED, UNIT OFF-LINE:	90	SEC
TAPE LOADED, UNIT ON-LINE:	145	SEC

3.0 ERROR INFORMATION

SEQ 0010

3.1 ERROR REPORTING

- 1 - COMMAND PACKET ADR NOT ON MODULO 4 BOUNDARY
- 2 - TS04 NOT READY-SSR NOT SET
- * 3 - PDP11-TS11 WRAP FAILURE
- * 4 - PDP11-TS11 WRAP FAILURE ON TSSR EXT ADDR BITS
- * 5 - PDP11-TS04 WRAP FAILURE-TSBA INCORRECT
- * 6 - PDP11-TS04 WRAP FAILURE-TSSR INCORRECT
- 7 - TRACK ACTIVE NOT 0 FOR 1 OR MORE TRACKS IN TEST4 SUB1
- 8 - TRACK ACTIVE NOT 0 FOR 1 OR MORE TRACKS IN TEST4 SUB3
- 9 - TRACK ACTIVE NOT 1 FOR 1 OR MORE TRACKS IN TEST4 SUB2
- 10 - TRACK ACTIVE NOT 1 FOR 1 OR MORE TRACKS IN TEST4 SUB4
- 11 - TRACK ACTIVE ERROR
- 12 - PE WRAP DATA ON 0'S PATTERN
- 13 - PE WRAP DATA ERROR ON 1'S PATTERN
- 14 - PE WRAP DATA ERROR ON SHIFTING 1 PATTERN
- 15 - PE WRAP DATA ERROR ON SHIFTING 0 PATTERN
- 16 - PE SKEW ERROR
- 17 - PE DEAD TRACK ERROR
- 18 - ROM LOOKUP TABLE ERROR
- 19 - MICRO DIAGNOSTIC ERROR
- * 20 - SET CHAR ERROR-TSSR. NBA NOT SET ON COMMANDS BEFORE SET CHAR ISSUED.
- * 21 - SET CHAR ERROR-TSSR. NBA NOT CLEARED WHEN SET CHAR ISSUED
- * 22 - SET CHAR ERROR-TSBA NOT POINTING PAST MSG PACKET
- * 23 - SET CHAR ERROR-MSG PACKET CONTENTS IN DOUBT
- * 24 - SET CHAR ERROR-I/O SEQ CROM PARITY ERR (FC=1)
- * 25 - SET CHAR ERROR-SERIAL BUS PARITY ERR (FC=2)
- * 26 - SET CHAR ERROR-SILO PARITY ERR (FC=2)
- * 27 - SET CHAR ERROR-UP CROM PARITY ERROR OR FATAL MICRO ERROR HALTS (FC=2)
- * 28 - SET CHAR ERROR-AC DETECTED LOW AT TS04 (FC=3)
- * 29 - TS11 DID NOT DROP READY UPON COMMAND ISSUANCE
- 30 - PE DATA ERROR
- 31 - NO INTERRUPT

* = DEVICE FAIAL ERRORS - CALL BAD TS11 BOARD OR IO BOARDS

OTHER ERRORS ARE HARD ERRORS

NOTE: EXPECTED AND ACTUAL DATA AND/OR PRINTOUTS WILL OCCUR WITH SOME OF THE ABOVE MESSAGES WHEN APPLICABLE.

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:MOE. THERE ARE NO OTHER HALTS.

4.0 DEVICE INFORMATION TABLES

4.1 "DIAG" REGISTERS

;THE FOLLOWING ARE REGISTER AND BIT DEFINITIONS FOR THE TS04
;REGISTERS OF INTEREST IN THIS DIAGNOSTIC.

;FMCTLO - FORMATTER MAJOR STATE CONTROL REG
FMCTLO=4
;WRITE REGISTER 4

;THIS REGISTER IS SET UP FOR THE PORTION OF THE
;RECORD WE ARE PRESENTLY READING

FC.RD= 200 ;WE ARE DOING NORMAL READ. IF 0, WE DISABLE SOME ERROR
;CORRECTION LOGIC SO WE'RE MORE DISCRIMINATING
;FOR READ AFTER WRITE.
FC.FLO=100 ;SETTING THIS BIT CAUSES .FMFLO ON BBUS TO BE TRUE
;(NEEDED BY THE RD PE ROUTINE)
FC.DAT=10 ;DATA MODE
FC.PRE=4 ;PREAMBLE MODE
FC.VCO=2 ;VCO SYNC MODE
FC.NRZ=1 ;NRZI MODE (FORCE SKEW WINDOW TO STAY OPEN)
;CAUSES FMTAUI TO BE CLRED WHEN YOU WRITE
;TO THE FMCLDO REGISTER.

;RDCTLO - READ CONTROL REGISTER
RDCTLO=20
;WRITE REGISTER 20

;THIS REGISTER CONTROLS THE FORMATTER MODE AND THRESHOLD.

RD.REV=200 ;1 FOR REV MOTION, 0=FWD
RD.MAI=100 ;I/O FORMATTER DATA WRAPAROUND
RD.SPC=40 ;WE ARE SPACING RECORDS. (THIS BIT IS UNUSED IN
;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)
RD.SKP=20 ;WE ARE SKIPPING FILES. (THIS BIT IS UNUSED IN
;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)

;THE FOLLOWING THRESHOLDS ARE AVAILABLE:
;NORM USE DIAG USE
; HI PREAMP GAIN
; LO PREAMP, BAD TAPE CLEAN
; RESIDUAL ERASE CHK
; FWD/REV AMP BALANCE

RD.110=7
RD.90=6
RD.75=5
RD.68=4
RD.40=3
RD.20=2
RD.12=1
RD.07=0
;NRZ WRT
;NRZ RD, PE WRT
;PE READ, NRZ ERR RECOV RD/WRT CROSSTALK
;PE ERR RECOV ERASE FUNCTION
;DATA PORTION ONLY

```
*****
;IOSCO - I/O SEQUENCER SILO CONTROL BUFFER OUT
```

```
IOSCO=14
```

```
;WRITE
```

```
REGISTER 14
```

```
;THIS REGISTER CONTAINS THE SILO CONTROL BITS FOR DATA WRITTEN
;BY THE MAIN OR I/O MICRO TO THE SILO. THE DATA IN THIS REG
;IS PAIRED WITH THE IOSDO REG AND PUT IN THE SILO WHENEVER
;THE IOSICO REG IS CLOCKED. THIS REGISTER NEED ONLY BE WRITTEN
;ONCE IF THE SAME OLD DATA IS OK TO BE WRITTEN IN THE SILO.
```

```
;NOTE THAT IF THE I/O IS WRITING THE SILO, THE MAIN MUST PUT EVEN
;PARITY IN THE BITS 357 OR THE I/O WILL WRITE PAR ERRS IN THE SILO.
;(THE IS.DAP BIT IS DON'T CARE HERE FOR I/O WRITING THE SILO)
```

```
IS.PAR= 200 ;ODD PAR BIT FOR ALL 16 BITS (CNRTL AND DATA)
;NOTE THAT THE BITS MASKED BY 357 MUST BE
;EVEN PARITY BECAUSE THE 9 DATA BITS ARE ODD
```

```
IS.IVP= 100 ;INVERT CNTRL SILO PAR BIT BEFORE MOVING
;TO WRITE BOARD (WRITE EVEN PARITY ON TAPE)
```

```
IS.NRZ= 40 ;INVERT WRITE BUFFER BIT IF ASSOCIATED SILO
;DATA BIT IS A 1. (IF IS.NRZ=0, WE'RE IN PE MODE)
```

```
IS.DAP= 20 ;ODD PARITY FOR THE 8 DATA BITS IN IOSDO
```

```
IS.LRC= 10 ;CAUSES SYNCHRONOUS CLR ON WRT BOARD TO WRITE THE LRC CHAR
IS.WRF= 4 ;THIS FLG BIT SHOWS UP AT THE WRITE BOARD
;WITH THE CORRESPONDING DATA. THE FUNCTION OF
;THE BIT WILL BE DEFINED BY THE WRITE BOARD.
;NOTE HOWEVER THAT IF THE WRITE BOARD SEES
;THE BIT 1, THE PA.WRF BIT IN THE
;PRATNI REG WILL ALSO BE 1 (IF ENABLED
;TO AFFECT THE ATTN REG).
```

```
;THE 2 LOW BITS ARE WRITABLE AND AFFECT THE PARITY TREES BUT
;OTHERWISE ARE UNIMPLEMENTED
```

4.2 GENERAL

```
-----
THE TS04 TAPE SUBSYSTEM CONSISTS OF A TS11 UNIBUS TO SERIAL BUS
CONTROLLER CONNECTED TO A TS04 DRIVE. FROM A SOFTWARE VIEWPOINT
THIS CONFIGURATION IS UNIQUE (FOR A UNIBUS DEVICE) IN A NUMBER
OF WAYS:
```

- A. ONLY ONE REGISTER MAY BE WRITTEN - TSDB (TAPE SYSTEM DATA BUFFER),
- B. TWO REGISTERS MAY BE READ - TSSR AND TSBA (TAPE SYSTEM STATUS REGISTER AND TAPE SYSTEM BUS ADDRESS REGISTER),
- C. COMMANDS ARE NOT WRITTEN TO THE DRIVE; RATHER, COMMAND POINTERS ARE WRITTEN WHICH POINT TO COMMAND PACKETS SOMEWHERE IN CPU MEMORY. THE COMMAND POINTER IS USED BY THE TS04 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:

1. COMMAND WORD
2. LOW ORDER BUFFER ADDRESS
3. HIGH ORDER BUFFER ADDRESS
4. BYTE COUNT

- D. THE TSSR CONTAINS ALL THE INFORMATION WHICH WILL BE NECESSARY TO DETERMINE WHETHER:
1. THE DRIVE IS READY TO ACCEPT ANOTHER COMMAND,
 2. THE PREVIOUS COMMAND WAS EXECUTED WITHOUT ERROR.
- IF EITHER OF THE ABOVE CONDITIONS IS UNTRUE AT "JOB DONE" OR "COMMAND INITIATION" TIME, IT MAY BE NECESSARY TO GET THE EXTENDED STATUS REGISTERS TO DETERMINE WHAT ACTION IS TO BE TAKEN AND/OR LOG THE ERROR INFORMATION.
- E. EXTENDED STATUS REGISTERS ARE NOT READ DIRECTLY FROM DRIVE REGISTERS; RATHER, A "GET STATUS" COMMAND IS ISSUED WHICH WILL CAUSE THE TS04 TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FOUR EXTENDED STATUS REGISTERS. SEE 6.3.
- F. THE TSDB MUST BE WRITTEN WITH A DATO INSTRUCTION TO PROPERLY WRITE THE COMMAND POINTER. A DATOB WILL CAUSE A MAINTENANCE FUNCTION. A DATO TO THE TSSR WILL CAUSE SUBSYSTEM INIT.
- G. COMMAND PACKETS MUST RESIDE ON DIVIDE BY FOUR MEMORY BOUNDARIES (AS OPPOSED TO DIVIDE BY 2 OR WORD BOUNDARIES) .

<u>TS11/ TS04</u>	<u>INT. VECTOR</u>	<u>UNIBUS ADDRESS</u>	<u>REGISTER</u>
FIRST	224	772520 772522	TSBA/TSDB TSSR
SECOND	154	772524 772526	TSBA/TSDB TSSR
THIRD	160	772530 772532	TSBA/TSDB TSSR
FOURTH	164	772534 772536	TSBA/TSDB TSSR

4.4 BIT DEFINITIONS FOR TS11/TS04 REGISTERS

4.4.1 TS11/TS04 REGISTER SUMMARY

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
TSBA	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00	
TSDB	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P17	P16	
TSSR	SC	UPE	SPE	RMR	NXM	NBA	A17	A16	SSR	OFL	FC1	FC0	TC2	TC1	TC0		
XST0	TMK	RLS	LET	RLI	WLE	NEF	ILC	ILA	MOT	ONL	IE	VCK	PED	WLK	BOT	EOT	
XST1	DLT		COR	CRS	TIG	DBF	SCK		IPR	SYN	IPO	IED	POS	POL	UNC	MTE	
XST2	OPM	SIP	BPE	CAF		WCF		DTP	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
XST3	MICRO DIAGNOSTIC ERROR CODE							LMX	OPI	REV	CRF	DCK	NOI	LXS	RIB		

TERMINATION CLASS CODES (TSSR TC0-TC2):

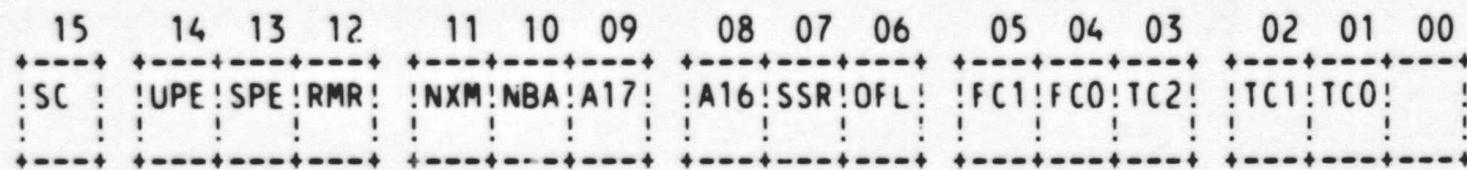
- 0 = NORMAL TERMINATION
- 1 = ATTENTION CONDITION
- 2 = TAPE STATUS ALERT
- 3 = FUNCTION REJECT
- 4 = RECOVERABLE ERROR - TAPE POSITION = ONE RECORD
DOWN TAPE FROM START OF FUNCTION
- 5 = RECOVERABLE ERROR - TAPE NOT MOVED
- 6 = UNRECOVERABLE ERROR - TAPE POSITION LOST
- 7 = FATAL CONTROLLER ERROR

FATAL CLASS CODES (TSSR FC0-FC1):

- 0 = MICRO DIAGNOSTIC FAILURE (DISPLAYED IN TS04 OPERATOR PANEL AND XST3).
- 1 = I/O SEQUENCER CROM PARITY ERROR.
- 2 = MICROPROCESSOR CROM PARITY ERROR.
SILO PARITY ERROR.
SERIAL BUS PARITY ERROR DETECTED AT TS11 (SPE).
SERIAL BUS PARITY ERROR DETECTED AT TS04 (BPE).
FATAL ERROR HALTS 1750-1777 IN TS04 PROGRAM COUNTER DISPLAY.
- 3 = LOSS OF AC POWER HAS BEEN DETECTED.

4.4.2 TS11 STATUS REGISTER (TSSR)

UNIBUS ADDRESS + 2 - READ ONLY



BIT	NAME	TCC	DEFINITION
15	SC	S	SPECIAL CONDITION. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT. SPECIFICALLY, EITHER AN ERROR WAS DETECTED OR AN EXCEPTION CONDITION OCCURRED. EXCEPTION CONDITIONS CAN BE TAPE MARKS ON READ COMMANDS, REVERSE MOTION AND AT BOT, EOT WHILE WRITING, ETC. MAY ALSO BE SET BY THE ERROR BITS CONTAINED IN THE TSSR REGISTER: UPE, SPE, RMR, AND NXM. THE TERMINATION CLASS BITS ARE SOMETHING OTHER THAN 0 (UNLESS RMR IS THE ONLY ERROR - SEE RMR).
14	UPE	4/5	UNIBUS PARITY ERROR. SET BY THE TS11 WHEN IT DETECTS A PARITY ERROR ON THE UNIBUS DATA WHEN TRANSFERRING TO OR FROM THE CPU'S MEMORY.
13	SPE	7	SERIAL BUS PARITY ERROR. THIS BIT IS SET BY THE TS11 WHEN IT DETECTS A SERIAL BUS PARITY ERROR ON DATA RECEIVED FROM THE TS04.
12	RMR	S	REGISTER MODIFICATION REFUSED. SET BY THE TS11 WHEN A COMMAND POINTER IS LOADED INTO TSDB AND SUB-SYSTEM READY (SSR) IS NOT SET. NOTE THAT THIS BIT CAUSES SPECIAL CONDITION BUT NO TERMINATION CLASS (IN FACT, THE TS04 NEVER SEES THIS ERROR) BECAUSE ON A SYSTEM WITH NO BUGS, THIS BIT MAY COME UP ON AN ATTENTION MESSAGE. IF ATTNS ARE NOT ENABLED, THIS BIT COMING UP IS AN INDICATION OF EITHER A FATAL CONTROLLER ERROR OR A SOFTWARE BUG.
11	NXM	4/5	NON-EXISTENT MEMORY. SET BY THE TS11 WHEN TRYING TO TRANSFER TO OR FROM A MEMORY LOCATION WHICH DOES NOT EXIST. MAY OCCUR WHEN FETCHING THE COMMAND PACKET, FETCHING OR STORING DATA, OR STORING THE MESSAGE PACKET.
10	NBA	S	NEED BUFFER ADDRESS. WHEN SET, INDICATES THAT THE TS04 NEEDS A MESSAGE BUFFER ADDRESS. THIS BIT IS CLEARED DURING THE SET CHARACTERISTICS COMMAND (IF A GOOD ADDRESS WAS GIVEN).
09	A17	S	BUS ADDRESS BIT 17. A17 AND A16 (BIT 08) TRACK

E 2

THE VALUES OF BITS 17 AND 16 OF THE TSBA REGISTER.

SEQ 0017

08	A16	S	BUS ADDRESS BIT 16. SEE A17 (BIT 09).
07	SSR	S	SUB-SYSTEM READY. WHEN SET, INDICATES THAT THE TS11/TS04 SUBSYSTEM IS NOT BUSY AND IS READY TO ACCEPT A NEW COMMAND POINTER.
06	OFL	S,1,3	OFF-LINE. WHEN SET, INDICATES THAT THE TS04 IS OFF-LINE AND UNAVAILABLE FOR ANY TAPE MOTION COMMANDS. THIS BIT CAN CAUSE A TERMINATION CLASS OF 1 (ON ATTN INTERRUPT) OR 3 (RESULTS IN NEF).
05	FC1	7	FATAL TERMINATION CLASS 01. FC1 AND FC0 (BIT 04) ARE USED TO INDICATE THE TYPE OF FATAL ERROR WHICH HAS OCCURRED ON THE TS04. THESE BITS ARE VALID ONLY WHEN SC IS SET AND THE TERMINATION CLASS CODE BITS ARE ALL SET (111).
04	FC0	7	FATAL TERMINATION CLASS 00. SEE FC1 (BIT 05).
03	TC2	S	TERMINATION CLASS BIT 02. THIS BIT, ALONG WITH THE TC1 AND TC0 BITS, ACT AS AN OFFSET VALUE WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS ON A COMMAND. EACH OF THE EIGHT POSSIBLE VALUES OF THIS FIELD REPRESENT A PARTICULAR CLASS OF ERRORS OR EXCEPTIONS. THE CONDITIONS IN EACH CLASS HAVE SIMILAR SIGNIFICANCE AND, AS APPLICABLE, RECOVERY PROCEDURES. THE CODE PROVIDED IN THIS FIELD IS EXPECTED TO BE UTILIZED AS AN OFFSET INTO A DISPATCH TABLE FOR HANDLING OF THE CONDITION.
02	TC1	S	TERMINATION CLASS BIT 01. SEE TC2 (BIT 03).
01	TC0	S	TERMINATION CLASS BIT 00. SEE TC2 (BIT 03).
00	-	-	NOT USED.

UNIBUS ADDRESS + 2 - WRITE ONLY

SUBSYSTEM INITIALIZE

4.4.3 EXTENDED STATUS REGISTER 0 (XSTAT0)

```

15  14 13 12  11 10 09  08 07 06  05 04 03  02 01 00
+---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+
|TMK| |RLS| |LET| |RLL| |WLE| |NEF| |ILC| |ILA| |MOT| |ONL| |IE | |VCK| |PED| |WLK| |BOT| |EOT|
|---| |---| |---| |---| |---| |---| |---| |---| |---| |---| |---| |---| |---|
+---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+

```

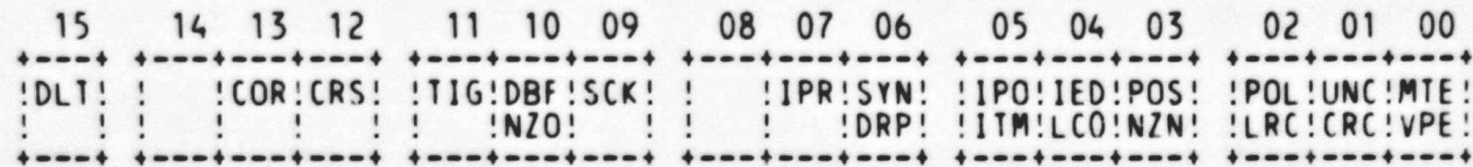
BIT	NAME	TCC	DEFINITION
15	TMK	S,2	TAPE MARK DETECTED. SET WHENEVER A TAPE MARK WAS DETECTED DURING A READ, SPACE, OR SKIP COMMAND AND AS A RESULT OF THE WRITE TAPE MARK OR WITE TAPE MARK RETRY COMMANDS.
14	RLS	2	RECORD LENGTH SHORT. THIS BIT INDICATES THAT EITHER THE RECORD'S LENGTH WAS SHORTER THAN THE BYTE COUNT ON READ OPERATIONS, A SPACE RECORD OPERATION ENCOUNTERED A TAPE MARK OR BOT BEFORE THE POSITION COUNT WAS EXHAUSTED, OR A SKIP TAPE MARKS COMMAND WAS TERMINATED BY ENCOUNTERING BOT OR A DOUBLE TAPE MARK (IF THAT OPERATIONAL MODE IS ENABLED, SEE LET) PRIOR TO EXHAUSTING THE POSITION COUNTER.
13	LET	2	LOGICAL END OF TAPE. SET ONLY ON THE SKIP TAPE MARKS COMMAND WHEN EITHER TWO CONTIGUOUS TAPE MARKS ARE DETECTED OR WHEN MOVING OFF OF BOT AND THE FIRST RECORD ENCOUNTERED IS A TAPE MARK. THE SETTING OF THIS BIT WILL NOT OCCUR UNLESS THIS MODE OF TERMINATION IS ENABLED THROUGH USE OF THE SET CHARACTERISTICS COMMAND.
12	RLL	2	RECORD LENGTH LONG. WHEN SET, THIS BIT INDICATES THAT THE RECORD READ WAS LONGER THAN THE BYTE COUNT SPECIFIED.
11	WLE	3,6	WRITE LOCK ERROR. WHEN SET, INDICATES THAT A WRITE OPERATION WAS ISSUED BUT THE MOUNTED TAPE DID NOT CONTAIN A WRITE ENABLE RING OR THE WRT LOCK SWITCH ACTIVATED DURING THE OPERATION.
10	NEF	3	NON-EXECUTABLE FUNCTION. WHEN SET, INDICATES THAT THE COMMAND COULD NOT BE EXECUTED DUE TO ONE OF THE FOLLOWING CONDITIONS: <ul style="list-style-type: none"> - THE COMMAND SPECIFIED REVERSE TAPE DIRECTION BUT THE TAPE WAS ALREADY POSITIONED AT BOT. - THE ISSUING OF ANY COMMAND, EXCEPT REWIND,

UNLOAD, OR A COMMAND WITH THE CLEAR VOLUME CHECK (CVC) BIT SET, WHEN THE VOLUME CHECK BIT IS SET.

- ANY COMMAND, EXCEPT GET STATUS OR DRIVE INITIALIZE, WHEN THE TS04 IS OFF-LINE.
- ANY WRITE COMMAND WHEN THE TAPE DOES NOT CONTAIN A WRITE ENABLE RING (WRITE LOCK STATUS - WLS).

09	ILC	3	ILLEGAL COMMAND. SET WHEN A COMMAND IS ISSUED AND EITHER ITS COMMAND FIELD OR ITS COMMAND MODE FIELD CONTAINS CODES WHICH ARE NOT SUPPORTED BY THE TS04.
08	ILA	3	ILLEGAL ADDRESS. (MORE THAN 18 BITS OR ODD WHEN AN EVEN ADDRESS IS REQUIRED.)
07	MOT	S	TAPE IS MOVING.
06	ONL	S	ON LINE. WHEN SET, INDICATES THAT THE TS04 IS ON-LINE AND OPERABLE.
05	IE	S	INTERRUPT ENABLE. REFLECTS THE STATE OF THE INTERRUPT ENABLE BIT SUPPLIED ON THE LAST COMMAND.
04	VCK	S	VOLUME CHECK. WHEN SET, INDICATES THAT THE DRIVE HAS BEEN EITHER POWERED DOWN OR TURNED OFF-LINE. CLEARED BY THE CLEAR VOLUME CHECK (CVC) BIT IN THE COMMAND HEADER WORD. THIS BIT CAN CAUSE A TERMINATION CLASS OF 3.
03	PED	S	PHASE ENCODED DRIVE. WHEN SET, INDICATES THAT THE TS04 IS CAPABLE OF READING AND WRITING ONLY 1600 BPI PHASE ENCODED DATA. WHEN RESET, INDICATES THAT THE TS04 HAS ONLY 800 BPI NRZI DATA CAPABILITIES.
02	WLK	S,3	WRITE LOCKED. WHEN SET, INDICATES THAT THE MOUNTED REEL OF TAPE DOES NOT HAVE A WRITE-ENABLE RING INSTALLED. THE TAPE IS, THEREFORE, WRITE PROTECTED.
01	BOT	S,3	BEGINNING OF TAPE. WHEN SET, INDICATES THAT THE TAPE IS POSITIONED AT THE LOAD POINT AS DENOTED BY THE BOT REFLECTIVE STRIP ON THE TAPE.
00	EOT	S,2	END OF TAPE. THIS BIT IS SET WHENEVER THE TAPE IS POSITIONED AT OR BEYOND THE END OF TAPE REFLECTIVE STRIP. DOES NOT RESET UNTIL THE TAPE PASSES OVER THE REFLECTIVE STRIP IN THE REVERSE DIRECTION UNDER PROGRAM CONTROL.

4.4.4 EXTENDED STATUS REGISTER 1 (XSTAT1)

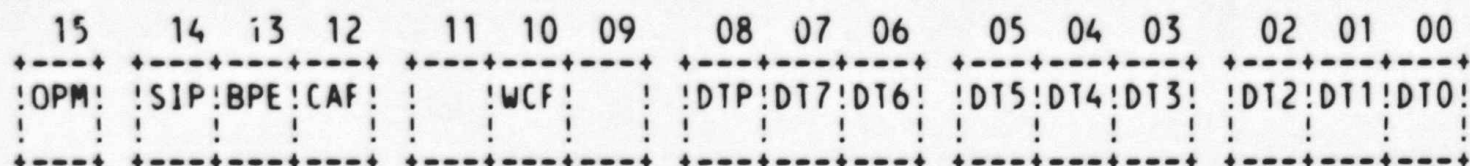


BIT	NAME	TCC	DEFINITION
15	DLT	4	DATA LATE. SET WHEN THE I/O SILO IS FULL ON A READ OR EMPTY ON A WRITE. THESE CONDITIONS OCCUR WHENEVER THE UNIBUS LATENCY EXCEEDS THE DATA TRANSFER RATE OF THE TS04.
14	-	-	NOT USED.
13	COR	5	CORRECTABLE DATA. IN PHASE ENCODED MODE, A CORRECTABLE DATA ERROR HAS BEEN ENCOUNTERED.
12	CRS	4	CREASE DETECTED. FOR NRZI, ALL DATA TRACKS DROPPED OUT FOR MORE THAN THREE CHARACTER TIMES BUT FOR LESS THAN .1 INCHES OF TAPE. FOR PE, EIGHT OUT OF NINE DATA TRACKS WENT DEAD FOR LESS THAN .1 INCHES BEFORE A VALID POSTAMBLE WAS DETECTED.
11	TIG	4	TRASH IN THE GAP. NON-ERASED DATA WAS DETECTED IN A GAP DURING A READ, WRITE, WRITE TAPE MARK, OR ERASE COMMAND.
10	DBF	4	EXCESSIVE SKEW. FOR NRZI, DATA OCCURRED BETWEEN THE 50% MARK AND THE 100% MARK OF THE NRZI DATA WINDOW. FOR PE, IT TOOK MORE THAN FIVE CHARACTERS IN READ-AFTER-WRITE OR TEN CHARACTERS IN READ TO PROPERLY CENTER THE WINDOWS OF THE FORMAT CHANNEL LOGIC.
	NZO	4	NRZ FIFO OVERRUN.
09	SCK	4	SPEED CHECK. TAPE SPEED WAS OFF BY MORE THAN 5% DURING A WRITE DATA OPERATION. NOTE THAT SPEED AVERAGED OVER 8 TICKS AND THE AVERAGE MUST BE OFF 5% TO CAUSE THIS ERROR.
08	-	-	NOT USED.
07	IPR	5,4	INVALID PREAMBLE. SET ON A PE DRIVE IF THE PREAMBLE APPEARS TO BE SHORTER THAN 36 CHARACTERS OR LONGER THAN 44 CHARACTERS. ALSO SET IF THE PREAMBLE IS INCORRECTLY ENCODED BEYOND THE FIFTEENTH CHARACTER IN READ OR THE TENTH CHARACTER IN READ-AFTER-WRITE.

06	SYN	4	SYNCH FAILURE. SET ON A PE DRIVE IF THE FORMATTER WAS UNABLE TO ACHIEVE SYNCHRONIZATION IN THE PREAMBLE.
	DRP	4	NRZ RECORD DROPPED A CHARACTER (THE NEXT CHARACTER WAS TO BE CONSIDERED CRC).
05	IPO	S,4	INVALID POSTAMBLE. SET ON A PE DRIVE DURING READ OR WRITE IF ANY OF THE FIRST 39 CHARACTERS OF THE POSTAMBLE ARE NOT READ CORRECTLY.
	ITM	S,4	ILLEGAL TAPE MARK FOR NRZ.
04	IED	4	INVALID END DATA. FOR PE, EIGHT OUT OF NINE TRACKS WENT DEAD BEFORE THE POSTAMBLE WAS DETECTED.
	LRO	4	FOR NRZI, DATA WAS NOT DETECTED IN EITHER THE LRCC OR CRCC WINDOWS. (LRC WAS ZERO)
03	POS	S,4	POSTAMBLE SHORT. SET ON PE DRIVES DURING A READ OR WRITE WHEN LESS THAN 38 ALL-ZEROES CHARACTERS ARE READ FOLLOWING THE ALL-ONES CHARACTER.
	NZN	S,4	NRZ NOISE RECORD (FEWER THAN 13(10) FRAMES).
02	POL	4	POSTAMBLE LONG. SET ON PE DRIVES DURING READ OR WRITE OPERATIONS WHEN THE POSTAMBLE EXCEEDS 42 CHARACTERS.
	LRC	4	LRC ERROR. SET ON NRZI DRIVES WHEN THE LRCC CHARACTER WAS FOUND IN ERROR.
01	UNC	4	UNCORRECTABLE DATA. SET ON PE DRIVES WHEN A PARITY ERROR OCCURRED WITHOUT A CORRESPONDING DEAD TRACK INDICATION.
	CRC	4	CRC ERROR. SET ON NRZI DRIVES WHEN THE CRC CHARACTER WAS FOUND TO BE IN ERROR.
00	MTE	4	MULTI-TRACK ERROR. SET ON PE DRIVES WHEN MORE THAN ONE DEAD TRACK OCCURRED IN THE PREAMBLE OR IN THE DATA FIELD.
	VPE	4	VERTICAL PARITY ERROR. SET ON NRZI DRIVES WHEN A CHARACTER DID NOT CONTAIN AN ODD NUMBER OF ONE BITS.

4.4.5 EXTENDED STATUS REGISTER 2 (XSTAT2)

SEQ 0022



BIT	NAME	TCC	DEFINITION
15	OPM	S	OPERATION IN PROGRESS. (TAPE MOVING)
14	SIP	7	SIL0 PARITY ERROR. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
13	BPE	7	SERIAL BUS PARITY ERROR AT DRIVE. SET BY THE TS04 WHEN A PARITY ERROR IS DETECTED ON DATA TRANSMITTED FROM THE TS11 TO THE TS04. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
12	CAF	7	CAPSTAN ACCELERATION FAIL. AFTER ACCELERATING TAPE FOR .2 INCHES, THE TAPE SPEED WAS CHECKED AND FOUND TO BE OUT OF TOLERANCE BY MORE THAN 10%.
11	-	-	NOT USED.
10	WCF	7	DESKEW BUFFER FAIL. ONE OF THE DESKEW BUFFERS FAILED TO ASSERT "OUTPUT READY" WITHIN 20 MICROSECONDS AFTER BEING ENABLED. THE DEAD TRACK BITS WILL INDICATE ON WHICH TRACKS THIS FAILURE OCCURRED.
09	-	-	NOT USED.
08	DTP	S	DEAD TRACK PARITY. THE BITS DTP THROUGH DT0 INDICATE WHICH TRACK(S) WENT DEAD, IF ANY, DURING THE LAST DATA TRANSFER OPERATION. IF DESKEW BUFFER FAIL (DBF) IS SET, THESE BITS INDICATE WHICH CHANNEL FAILED.
07	DT7	S	DEAD TRACK 7. SEE DTP.
06	DT6	S	DEAD TRACK 6. SEE DTP.
05	DT5	S	DEAD TRACK 5. SEE DTP.
04	DT4	S	DEAD TRACK 4. SEE DTP.
03	DT3	S	DEAD TRACK 3. SEE DTP.
02	DT2	S	DEAD TRACK 2. SEE DTP.

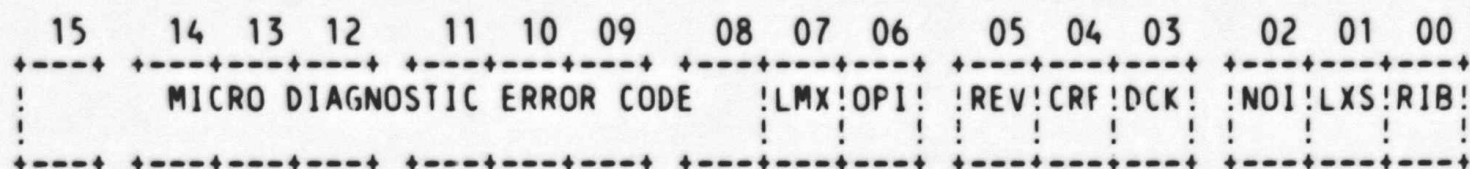
01 DT1 S DEAD TRACK 1. SEE DTP.

00 DT0 S DEAD TRACK 0. SEE DTP.

SEQ 0023

NOTE: ON A SET CHARACTERISTICS COMMAND, THE UCODE LEVEL IS RETURNED IN DT7 THRU DT0. ON A GET STATUS COMMAND, THE RESIDUAL CAPSTAN TICK COUNT (INTERNALLY R7) IS RETURNED IN DT7 THRU DT0.

4.4.6 EXTENDED STATUS REGISTER 3 (XSTAT3)



BIT	NAME	TCC	DEFINITION
15 TO 08			MICRO DIAGNOSTIC ERROR CODE. (SEE LIST OF CODES BELOW). ALL ERROR CODES IN THE TABLE WILL BE DISPLAYED ON THE TSO4 CONTROL PANEL BUT ONLY CODES HIGHER THAN 110 WILL BE AVAILABLE TO CPU DIAGNOSTICS FOR PRINTOUT IN THE MICRO DIAGNOSTIC ERROR CODE FIELD OF XSTAT3. THIS ERROR CODE FIELD IS VALID ONLY WHEN THE TERMINATION CLASS CODE IN THE TSSR EQUALS 7 AND THE FATAL CLASS CODE IN THE TSSR EQUALS 0, INDICATING AN INTERNAL DIAGNOSTIC FAILURE.
07	NTL	6	LIMIT EXCEEDED. SET WHEN THE TAPE TENSION ARMS HAVE EXCEEDED THEIR ALLOWABLE TRAVEL AND HAVE CAUSED THE ACTIVATION OF THE LIMIT SWITCHES. NO TENSION EXISTS ON THE MOUNTED TAPE.
06	OPI	6	OPERATION INCOMPLETE. SET WHEN A READ, SPACE, OR SKIP OPERATION HAS MOVED 25 FEET OF TAPE WITHOUT DETECTING ANY DATA ON THE TAPE.
05	REV	5	DIRECTION OF CURRENT OPERATION WAS REVERSE (BUT IS 0 IF REWIND OR FORWARD)
04	CRF	7	CAPSTAN RESPONSE FAILURE. A MOTION COMMAND WAS GIVEN TO THE CAPSTAN BUT WE DID NOT GET A TICK BACK WITHIN A REASONABLE AMOUNT OF TIME.
03	DCK	5,6	DENSITY CHECK. SET ON PE DRIVES WHEN A PE IDENTIFICATION BURST WAS NOT DETECTED WHEN MOVING OFF OF BOT. SET ON NRZI DRIVES WHEN A NON-NRZI IDENTIFICATION BURST WAS FOUND WHEN MOVING OFF OF BOT.
02	NOI	6	NOISE RECORD. SET DURING A READ OR SPACE OPERATION WHEN A BURST OF FLUX CHANGES, WHICH DO NOT QUALIFY AS A RECORD (BUT TOO MANY TO IGNORE), ARE DETECTED: NRZI: AT LEAST TWO CHARACTERS IN A ROW BUT LESS THAN TWELVE, FOLLOWED BY A CHARACTER IN EITHER THE CRCC OR LRCC WINDOWS. PE: AT LEAST 24 CHARACTERS IN A ROW THAT DO NOT QUALIFY AS A TAPE MARK OR A DATA PREAMBLE.

01 LXS S LIMIT EXCEEDED STATICALLY. THIS BIT IS SET ANY TIME THE LIMIT SWITCHES ARE EXCEEDED. THIS BIT CAN ONLY BE CLEARED BY MANUALLY LOADING THE TAPE. M 2

SEQ 0025

00 RIB 2 REVERSE INTO BOT. A READ, SPACE, OR SKIP COMMAND ALREADY IN PROGRESS HAS ENCOUNTERED THE BOT MARKER WHEN MOVING TAPE IN THE REVERSE DIRECTION. TAPE MOTION WILL BE HALTED AT BOT.

4.4.7 MICRO DIAGNOSTIC ERROR CODES

FOLLOWING IS A LIST OF THE ERRORS WHICH ARE DISPLAYED IN THE MICRO DIAGNOSTIC ERROR CODE (XSTAT3 BITS 15 - 08) AND ALSO IN THE LIGHTS ON THE TSO4 CONTROL PANEL, DUE TO FAILURES ON THE CAPSTAN BOARD, I/O BOARDS, WRITE BOARD, READ BOARD, OR FORMATTER BOARD. THE MICRO WILL BE IN A TIGHT LOOP IN THE DISPM PROGRAM, WAITING FOR OPERATOR OR CPU INTERVENTION WHILE THE ERROR IS BEING DISPLAYED IN THE CONSOLE LIGHTS. IT IS APPARENT THAT AN ERROR IS BEING DISPLAYED IF THE "UOK" LIGHT IS NOT LIGHTED, THE PROCESSOR IS NOT STOPPED, AND AN OCTAL NUMBER (100-377) IS BEING DISPLAYED IN THE LIGHTS. TO SCOPE LOOP THESE TESTS, ENTER MAINTENANCE MODE (ON-LINE SWITCH TO "OFF" POSITION, MAINTENANCE SWITCH UP, PRESS RESET), ENTER THE OFF-LINE TEST NUMBER (SEE SCOPE LOOP COLUMN BELOW) IN THE OPERATOR CONSOLE LIGHTS (ENTER ONES WITH LEFT-MOST SWITCH, ENTER ZEROES WITH RIGHT-MOST SWITCH), AND PRESS ON-LINE BUTTON. TEST WILL LOOP UNTIL ON-LINE SWITCH IS RETURNED TO OFF-LINE POSITION, ERRORS WILL BE DISPLAYED CONTINUOUSLY.

ERROR (DISPLAY)	PROGRAM	ERROR DESCRIPTION	LIKELY MODULE	SCOPE LOOP
337	OPERATIONAL CODE	CAPSTAN RUNAWAY ERROR (H3.RNY). CAPSTAN DIDN'T STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND.		
100	IOTSM	BASIC I/O MICRO FAILURE (PARITY ERROR, IOATN, HANDSHAKING, AND DATA WINDOW TEST BETWEEN THE I/O AND MAIN MICROS. NOTE: CAN ALSO BE CAUSED BY THE SERIAL BUS .SHIN (SHIFT IN) STUCK ASSERTED.	M8967	14
101	IOTSM	ERROR IN I/O CONTROL REGISTER TEST	M8966 M8967	15
102	IOTSM	FAILURE OF FRAME COUNTER TEST	M8966	15
103	IOTSM	FAILURE OF I/O SILO NON-PARITY ERROR DATA TEST OR THE WRITE FLAG.	M8966 M8963	16
104	IOTSM	FAILURE OF I/O SILO PARITY ERROR	M8966	17

		TEST OR DATA LATE TEST.		
105	IOTSM	FAILURE OF SHIFT LOOP WITH ZEROES.	M8965	20
106	IOTSM	FAILURE OF SHIFT LOOP WITH ONES.	M8965	21
107	IOTSM	FAILURE OF SHIFT LENGTH MUX.	M8965	22
110	IOTSM	FAILURE TO RECEIVE CORRECT OP-CODE FROM TS11 WHEN WE SENT DATA OVER THE SERIAL BUS.	M8965 TS11 MOTHER BD SBUS CABLE	47
111	CATSM	FAILURE OF 1 KHZ CLOCK TEST. TSTS TAC SYNC FLOP AND ATTN, TOO.	G159 CBUS CABLE M8963	2
112	CATSM	LIGHT REGISTER CHANGED WHEN MOTION REGISTER WAS CLEARED.	G159	3,4
113	CATSM	FWD OR MVG BITS WRONG AFTER 1 TICK OF SIMULATED COMMAND AND TACH PULSES.	G159	3,4
114	CATSM	FAILURE OF SIMULATED CAPSTAN SPEED TEST. THE CAPSTAN SPEED COUNTER WAS OUT OF RANGE WHEN TAPE MOTION AT SPEED WAS SIMULATED.	G159	3,4
115	CATSM	FAILURE OF SIMULATED SLOW CAPSTAN TEST. SPEED COUNTER DID NOT LATCH UP WITH MAX COUNT WHEN SLOW TACH TICKS WERE SIMULATED.	G159	3,4
116	CATSM	FAILURE OF SIMULATED CAPSTAN DECEL TEST. COUNTER NOT ZERO FOR FORWARD OR 377 FOR REVERSE WHILE DECELERATING, OR MVG BIT NOT 1.	G159	3,4
117	CATSM	FAILURE OF MOVING FLOP TO GO TO ZERO AFTER STOPPING (DIRECTION REVERSAL FOR ONE TACH TICK).	G159	3,4
120	PETSM	FAILURE OF WRITE BOARD TO TURN ON AND EMPTY THE SILO, OR DATA LATE BIT DOESN'T WORK.	M8929 M8966	23
121	PETSM	FAILURE OF WRITE BOARD TO EMPTY SILO AT CORRECT SPEED.	M8929	23
124	PETSM	FORMATTER FLAG DOESN'T WORK ON THE M8922.	M8922	24
125	PETSM	FORMATTER SILO FILLING AND DATA ERROR	M8922 M8923 M8924	24
126	PETSM	PEAK SHIFT TEST ERROR	M8922 M8923 M8924	25

127

PETSM

FORMATTER TABLE LOOKUP ROM CHECKSUM
TEST ERROR

B 3
M8922 26
M8923
M8924

SEQ 0027

5.1 TEST 1 - PDP-11/TS11 WRAP TEST

TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND THE TS11 BY WRAPPING THE FOLLOWING PATTERNS:

A 1 IN A FIELD OF 0'S A 0 IN A FIELD OF 1'S.

WHEN DATA IS WRITTEN TO THE TSDB HI BYTE, THE DATA IS WRAPPED AROUND WITHIN THE TS11 AND APPEARS IN THE TSBA LO AND HI BYTES. THE 2 LOW ORDER BITS OF THE DATA WILL BE REFLECTED IN THE TSSR EXTENDED ADDRESS BITS.

R4 CONTAINS A COPY OF THE DATA SENT.

R3 CONTAINS THE EXPECTED TSBA RESULTS

R2 CONTAINS THE EXPECTED STATE OF THE TWO EXTENDED ADDRESS BITS IN THE TSSR.

5.2 TEST 2 - PDP-11/TS04 WRAP TEST

TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND THE TS04 BY WRAPPING THE FOLLOWING PATTERN:

A 1 IN A FIELD OF 0'S; A 0 IN A FIELD OF 1'S

WHEN THE DATA IS WRITTEN TO THE TSDB LO BYTE, THE DATA IS SENT TO THE TS04, VIA THE SERIAL LINE, WHERE IT IS WRAPPED AROUND BACK OVER THE SERIAL LINE TO THE TS11. THE DATA THEN APPEARS IN THE TSBA LO AND TSSR LO BYTES.

R4 CONTAINS THE EXPECTED TSBA RESULTS AND THE EXPECTED TSSR RESULTS.

5.3 TEST 3 - SET TS04 CHARACTERISTIC

THE FUNCTION OF THIS TEST IS TO ISSUE A "SET CHARACTERISTIC" COMMAND TO TELL THE TS04 WHERE IN CORE THE MESSAGE PACKET RESIDES AND TO VERIFY THAT A MESSAGE PACKET WAS STORED.

5.4

SEQ 0029

TESTS 4 - 7 PERFORM DATA WRAPS ON THE P.E. READ FORMATTER
 BOARDS. COMMUNICATION BETWEEN THE PDP11 AND TS04 OCCURS BY USING
 THE DIA (DIAGNOSTIC) COMMAND WHICH SENDS A COPY OF THE DIABLK TABLE,
 RESIDING IN CORE, TO THE TS04 CONTROLLER. THE FORMAT OF THE DIABLK IS
 SHOWN IN THE FOLLOWING TABLE. NOTE THAT THE TABLE IS FILLED IN REVERSE
 ORDER, THAT IS, THE LAST LOGICAL ENTRY OF THE TABLE IS LABELED DIABLK,
 WHILE THE FIRST LOGICAL ENTRY OF THE TABLE IS LABELED DIABLK+DIAEXT, WHERE
 DIAEXT IS THE LENGTH (EXTENT) OF THE TABLE IN BYTES.

WHEN THE DIA COMMAND IS EXECUTED, THE DIABLK IS LOADED ONTO THE TS04 STACK
 WITH THE FIRST LOGICAL ENTRY AT THE TOP OF THE STACK, AS SHOWN BELOW.
 THE TS04 THEN JUMPS TO THE P.E. WRAP TASK, IN ROM, WHERE THE FUNCTION IS
 EXECUTED USING THE REMAINING STACK ENTRIES AS ARGUMENTS.

DIABLK+DIAEXT:	TS04 PE	WRAP TASK ADDR	LO
	TS04 PE	WRAP TASK ADDR	HI
		READ CONTROL	(RDCTLO)
		FORMAT CONTROL	(FMCTLO)
		DATA	
		CONTROL	(IOSCO)
		DATA	
		CONTROL	(IOSCO)
		DATA	
		CONTROL	(IOSCO)
		DATA	
		CONTROL	(IOSCO)
		DATA	
		CONTROL	(IOSCO)
DIABLK:		DATA	
		CONTROL	(IOSCO)

5.4.1 TRACK ACTIVE/TRACK INACTIVE TESTS (TEST 4)

TEST 4 CHECKS THAT THE TRACK ACTIVE FLOP CAN SET AND CLEAR IN NRZI MODE FOR ALL CHANNELS. IF THE DATA DOES NOT MAKE A TRANSITION WHEN THE WRITE FLAG IS UP, THE TRACK ACTIVE FLOP WILL CLEAR. HOWEVER, IT WILL SET IF THERE IS A DATA TRANSITION WHILE THE WRITE FLAG IS UP.

5.4.1.1 SUBTEST 1 - TRACK INACTIVE TEST

THIS TEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING ALL 0'S DATA. THE PATTERN IS AS FOLLOWS FOR EACH CHANNEL:

DATA: 000000
 WRTFLG: 011100
 TRACK ACTIVE: SHOULD BE 0 FOR ALL TRACKS

5.4.1.2 SUBTEST 2 - TRACK ACTIVE TEST

THIS TEST FORCES TRACK ACTIVE TO SET BY WRITING THE FOLLOWING PATTERN ON EACH CHANNEL:

DATA: 110000
 WRTFLG: 011100
 TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.

5.4.1.3 SUBTEST 3 - TRACK INACTIVE TEST

THIS TEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING THE FOLLOWING PATTERN ON EACH CHANNEL:

DATA: 111111
 WRTFLG: 011100
 TRACK ACTIVE: SHOULD BE 0 FOR ALL TRACKS

5.4.1.4 SUBTEST 4 - TRACK ACTIVE TEST

THIS TEST FORCES THE TRACK ACTIVE FLOP TO SET BY WRITING THE FOLLOWING PATTERN ON EACH CHANNEL:

DATA: 001111
 WRTFLG: 011100
 TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.

5.4.2 P.E. DATA TEST (TEST 5)

TEST 5 WRAPS A DATA PATTERN TO CHECK EACH TRACK FOR BIT PICKUPS AND DROPS.

REGISTER USAGE IS AS FOLLOWS:

R2 = PREAMBLE DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
 R3 = 1ST BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
 THIS IS THE DATA INTEREST AFTER EXECUTING THE TS04 DIA
 COMMAND.
 R4 = 2ND BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.

5.4.2.1 SUBTEST 1 - P.E. DATA TEST/0 PATTERN.

THIS TEST WRAPS AN ALL 0'S PATTERN.

5.4.2.2 SUBTEST 2 - P.E. DATA TEST/1 PATTERN.

THIS TEST WRAPS AN ALL 1'S PATTERN.

5.4.2.3 SUBTEST 3 - P.E. DATA TEST/SHIFTING 1 PATTERN.

THIS TEST RIPPLES A 1 IN A FIELD OF 0'S.

5.4.2.4 SUBTEST 4 - P.E. DATA TEST/SHIFTING 0 PATTERN.

THIS TEST RIPPLES A 0 IN A FIELD OF 1'S.

5.4.3 TEST 6 SKEWS THE DATA ON A TRACK BY ONE BYTE WITH RESPECT TO ALL THE OTHER TRACKS. THAT IS, THE DATA IS ONE BYTE LATE ON THE ONE TRACK. EACH TRACK IS TESTED FOR SKEW IN THIS MANNER. REGISTER ASSIGNMENTS ARE AS FOLLOWS:

R2 = PREAMBLE DATA

R3 = BYTE 1 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK.
THAT TRACK CONTAINS PREAMBLE DATA)

R4 = BYTE 2 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK.
THAT TRACK CONTAINS BYTE 1 DATA)

5.4.3.1 SUBTEST 1 - P.E. SKEW TEST

THIS TEST WRITE AN ALL 1'S PREAMBLE (SKEWED), AN ALL 0'S BYTE 1 (SKEWED), AND AN ALL 1'S BYTE 2 DATA (SKEWED).

5.4.3.2 SUBTEST 2 - P.E. SKEW TEST

THIS TEST SENDS AN ALL 1'S PREAMBLE (SKEWED), AN ALL 1'S BYTE 1 DATA (SKEWED) AND AN ALL 0'S BYTE 2 DATA (SKEWED).

5.4.4 TEST 7 CHECKS THE DEAD TRACK LOGIC BY RIPPLING A DEAD TRACK THRU A FIELD OF LIVE TRACKS AND ONE LIVE TRACK THRU A FIELD OF DEAD TRACKS. ADDITIONALLY, EACH SUBTEST WILL SEND 1'S DATA AND 0'S DATA IN ORDER TO TEST THE 1'S OR DEAD REGISTER AND THE 0'S OR DEAD REGISTER.

REGISTER USAGE:

R2 = PREAMBLE ALL 1'S CHARACTER

R3 = 1ST DATA BYTE (BITS 0-8)

R4 = 2ND DATA BYTE (BITS 0-8)

DTKIDN = DEAD TRACK DEFINED IN BITS 0-8 (0=LIVE TRK; 1=DEAD TRK)

5.4.4.1 SUBTEST 1 - P.E. DEAD TRACK TEST

THIS TEST RIPPLES A DEAD TRACK IN A FIELD OF LIVE TRACKS.

5.4.4.2 SUBTEST 2 - P.E. DEAD TRACK TEST

THIS TEST RIPPLES A LIVE TRACK IN A FIELD OF DEAD TRACKS.

5.5 TEST 8 - LOOKUP TABLE TEST

THIS TEST VERIFIES THAT THE CONTENTS OF THE ROM LOOKUP TABLE ARE CORRECT. THE ROM CONTENTS IN ADDRESSES 1777-0 ARE CHECKED.

DATA AND REGISTER USAGE:

ROMLKI = ROM LOOKUP TABLE ADDRESS.
ERRLFG = ERROR FLAG.
R5 = DIABLK INDEX.

5.6 TEST 9 - INLINE MICRO DIAG TEST

ALLOWS INLINE MICRO DIAGS TO RUN, THEN CHECKS THE STATUS THEY RETURN.

5.7 TEST 10 - INIT MICRO DIAG TEST

ALLOWS INIT MICRO DIAGS TO RUN, THEN CHECKS THE STATUS THEY RETURN.

3	PROGRAM HEADER
105	DISPATCH TABLE
126	DESCRIPTIVE TEXT
143	DEFAULT HARDWARE P-TABLE
161	SOFTWARE P-TABLE
184	GLOBAL EQUATES SECTION
473	GLOBAL DATA SECTION
616	GLOBAL TEXT SECTION
632	GLOBAL ERROR REPORT SECTION
1145	GLOBAL SUBROUTINES SECTION
2575	LOAD DEV PROTECTION TABLE
2590	INITIALIZE SECTION
2681	AUTO DROP SECTION
2764	CLEANUP CODING SECTION
2791	DROP UNIT SECTION
2817	ADD UNIT SECTION
2849	TEST 1: PDP11/TS11 WRAP TEST.
3002	TEST 2: PDP11/TS04 WRAP TEST.
3078	TEST 3: SET TS04 CHARACTERISTIC VERIFICATION.
3280	TEST 4: TRACK INACTIVE/ACTIVE TEST.
3582	TEST 5: P.E. DATA TEST.
3792	TEST 6: P.E. SKEW TEST
3949	TEST 7: P.E. DEAD TRACK TEST.
4117	TEST 8: LOOKUP TABLE TEST
4192	TEST 9: IN-LINE MICRO DIAGNOSTIC TEST
4217	TEST 10: INIT MICRO DIAGNOSTIC TEST
4239	HARDWARE PARAMETER CODING SECTION
4277	SOFTWARE PARAMETER CODING SECTION
4321	HARD CODED P-TABLE

1

```

2      .TITLE PROGRAM HEADER AND TABLES
3      .SBTTL PROGRAM HEADER
4
5      .ENABL ABS,AMA
6      =      2000
7
8      002000      BGNMOD
9
10     :++
11     : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
12     : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
13     :--
14
15     002000      POINTER BGNAU,BGNDU,BGNSW,BGNSFT,BGNSETUP
16
17
18     002000      HEADER CZTSI,B,0,25,0
19     002000      L$NAME::      ;DIAGNOSTIC NAME
20     002000      103      .ASCII /C/
21     002001      132      .ASCII /Z/
22     002002      124      .ASCII /T/
23     002003      123      .ASCII /S/
24     002004      111      .ASCII /I/
25     002005      000      .BYTE 0
26     002006      000      .BYTE 0
27     002007      000      .BYTE 0
28     002010      L$REV::      ;REVISION LEVEL
29     002010      102      .ASCII /B/
30     002011      L$DEPO::      ;0
31     002011      060      .ASCII /O/
32     002012      L$UNIT::      ;NUMBER OF UNITS
33     002012      000001      .WORD T$PTHV
34     002014      L$TIML::      ;LONGEST TEST TIME
35     002014      000025      .WORD 25
36     002016      L$HPCP::      ;PTR. TO H.W. PTABLE
37     002016      034066      .WORD L$HARD
38     002020      L$SPCP::      ;PTR. TO S.W. PTABLE
39     002020      034140      .WORD L$SOFT
40     002022      L$HPTP::      ;PTR. TO DEF. H.W. PTABLE
41     002022      002204      .WORD L$HW
42     002024      L$SPTP::      ;PTR. TO S.W. PTABLE
43     002024      002212      .WORD L$SW
44     002026      L$LADP::      ;DIAG. END ADDRESS
45     002026      034534      .WORD L$LAST
46     002030      L$STA::      ;RESERVED FOR APT STATS
47     002030      000000      .WORD 0
48     002032      L$CO::      .WORD 0
49     002032      000000      .WORD 0
50     002034      L$DTYP::      ;DIAGNOSTIC TYPE
51     002034      000000      .WORD 0
52     002036      L$APT::      ;APT EXPANSION
53     002036      000000      .WORD 0
54     002040      L$DTP::      ;PTR. TO DISPATCH TABLE
55     002040      002124      .WORD L$DISPATCH
56     002042      L$PRIO::      ;DIAGNOSTIC RUN PRIORITY
57     002042      000000      .WORD 0
  
```

58	002044		L\$EXP1::	;EXPANSION WORDS	.WORD	0
59	002044	000000				
60	002046		L\$EXP2::		.WORD	0
61	002046	000000				
62	002050		L\$MREV::	;SVC REV AND EDIT #	.BYTE	C\$REVISION
63	002050	003			.BYTE	C\$EDIT
64	002051	002				
65	002052		L\$EF::	;DIAG. EVENT FLAGS	.WORD	0
66	002052	000000			.WORD	0
67	002054	000000				
68	002056		L\$SPC::		.WORD	0
69	002056	000000				
70	002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST	.WORD	L\$DVTYP
71	002060	002174				
72	002062		L\$REPP::	;PTR. TO REPORT CODE	.WORD	0
73	002062	000000				
74	002064		L\$EXP4::		.WORD	0
75	002064	000000				
76	002066		L\$EXP5::		.WORD	0
77	002066	000000				
78	002070		L\$AUT::	;PTR. TO ADD UNIT CODE	.WORD	L\$AU
79	002070	030112				
80	002072		L\$DUT::	;PTR. TO DROP UNIT CODE	.WORD	L\$DU
81	002072	030074				
82	002074		L\$LUN::	; LUN FOR EXERCISERS TO FILL	.WORD	0
83	002074	000000				
84	002076		L\$DESP::	; POINTER TO DIAG. DESCRIPTION	.WORD	L\$DESC
85	002076	002150				
86	002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT	EMT	E\$LOAD
87	002100	104035				
88	002102		L\$ETP::	;POINTER TO ERRtbl	.WORD	0
89	002102	000000				
90	002104		L\$IcP::	;PTR. TO INIT CODE	.WORD	L\$INIT
91	002104	027316				
92	002106		L\$CCP::	;PTR. TO CLEAN-UP CODE	.WORD	L\$CLEAN
93	002106	030052				
94	002110		L\$ACP::	;PTR. TO AUTO CODE	.WORD	L\$AUTO
95	002110	027600				
96	002112		L\$PRT::	;PTR. TO PROTECT TABLE	.WORD	L\$PROT
97	002112	027310				
98	002114		L\$TEST::	;TEST NUMBER	.WORD	0
99	002114	000000				
100	002116		L\$DLY::	;DELAY COUNT	.WORD	0
101	002116	000000				
102	002120		L\$HIME::	;PTR. TO HIGH MEM	.WORD	0
103	002120	000000				
104						

```
105      .SBTTL DISPATCH TABLE
106
107      ;++
108      ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
109      ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
110      ;--
111
112      DISPATCH 10
113      002122 000012                                .WORD 10
114      002124
115      002124 030130                                .WORD T1
116      002126 030464                                .WORD T2
117      002130 030644                                .WORD T3
118      002132 031336                                .WORD T4
119      002134 032232                                .WORD T5
120      002136 032674                                .WORD T6
121      002140 033256                                .WORD T7
122      002142 033640                                .WORD T8
123      002144 033774                                .WORD T9
124      002146 034042                                .WORD T10
125
126      .SBTTL DESCRIPTIVE TEXT
127
128      ;++
129      ; 2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAG AND THE DEVICE
130      ;--
131
132      DESCRIPT      <CONTROL LOGIC TEST>
133      L$DESC::
134      002150 047503 052116 047522                .ASCIZ /CONTROL LOGIC T
135      002156 020114 047514 044507
136      002164 020103 042524 052123
137      002172      000
138      002174                                .EVEN
139      DEVTYP      <TS11>
140      L$DVTYP::
141      002174 051524 030461 000                .ASCIZ /TS11/
142      002202                                .EVEN
```

143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160

002202
002202 000002
002204
002204
002204 172522
002206 000224
002210
002210

.SBTTL DEFAULT HARDWARE P-TABLE

;++
; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
:--

BGNHW DFPTBL

.WORD L10000-L\$HW/2

L\$HW::
DFPTBL::

172522 ;TSSR ADDRESS
224 ;VECTOR ADDRESS

ENDHW

L10000:

```
161          .SBTTL  SOFTWARE P-TABLE
162
163          ;++
164          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
165          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
166          ;--
167
168          BGNSW  SFPTBL
169 002210          .WORD  L10001-L$SW/2
170 002210 000001
171 002212          L$SW::
172 002212          SFPTBL::
173
174 002212          000          CMPFLG:: .BYTE 0          ;ENABLE DATA COMPARE ERROR PRINT FLAG
175          ;0=DO NOT ENABLE IS DEFAULT
176 002213          000          .BYTE 0          ;SPARE FLAG
177
178          ENDSW
179 002214          L10001:
180
181 002214          ENDMOD
```



```
182
183
184 .TITLE GLOBAL AREAS
185 002214 .SBTTL GLOBAL EQUATES SECTION
186 BGNMOD
187
188 :++
189 : THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
190 : ARE USED IN MORE THAN ONE TEST.
191 :--
192 002214
193
194 : EQUALS
195 : BIT DIFINITIONS
196 100000 BIT15== 100000
197 040000 BIT14== 40000
198 020000 BIT13== 20000
199 010000 BIT12== 10000
200 004000 BIT11== 4000
201 002000 BIT10== 2000
202 001000 BIT09== 1000
203 000400 BIT08== 400
204 000200 BIT07== 200
205 000100 BIT06== 100
206 000040 BIT05== 40
207 000020 BIT04== 20
208 000010 BIT03== 10
209 000004 BIT02== 4
210 000002 BIT01== 2
211 000001 BIT00== 1
212 :
213 001000 BIT9== BIT09
214 000400 BIT8== BIT08
215 000200 BIT7== BIT07
216 000100 BIT6== BIT06
217 000040 BIT5== BIT05
218 000020 BIT4== BIT04
219 000010 BIT3== BIT03
220 000004 BIT2== BIT02
221 000002 BIT1== BIT01
222 000001 BIT0== BIT00
223 :
224 : EVENT FLAG DEFINITIONS
225 : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
226 :
227 000040 EF.START== 32. : START COMMAND WAS ISSUED
228 000037 EF.RESTART== 31. : RESTART COMMAND WAS ISSUED
229 000036 EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED
230 000035 EF.NEW== 29. : A NEW PASS HAS BEEN STARTED
231 000034 EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED
232 :
233 :
234 : PRIORITY LEVEL DEFINITIONS
235 :
236 000340 PRI07== 340
237 000300 PRI06== 300
```

238	000240	PRI05== 240
239	000200	PRI04== 200
240	000140	PRI03== 140
241	000100	PRI02== 100
242	000040	PRI01== 40
243	000000	PRI00== 0
244		;
245		;OPERATOR FLAG BITS
246		;
247	000004	EVL== 4
248	000010	LOT== 10
249	000020	ADR== 20
250	000040	IDU== 40
251	000100	ISR== 100
252	000200	UAM== 200
253	000400	BOE== 400
254	001000	PNT== 1000
255	002000	PRI== 2000
256	004000	IXE== 4000
257	010000	IBE== 10000
258	020000	IER== 20000
259	040000	LOE== 40000
260	100000	HOE== 100000

261
262 ;BIT DEFINITIONS USED TO SPECIFY THE ROM ADDRESS LINES IN THE ROMLOK
263 ;SUBROUTINE

264		
265	001000	A9== 1000
266	000400	A8== 400
267	000200	A7== 200
268	000100	A6== 100
269	000040	A5== 40
270	000020	A4== 20
271	000010	A3== 10
272	000004	A2== 4
273	000002	A1== 2
274	000001	A0== 1

275
276
277 :*****

278 :*****

279 : THE FOLLOWING DEFINITIONS MAY CHANGE ON MICRO-CODE REASSEMBLY:

280			
281			
282	000005	WRPLO==5	;WRITE WRAP TASK ADR LO.
283	000200	WRPHI==200	;WRITE WRAP TASK ADR HI.
284	000000	POPJHI==0	;TS04 POPJ ADDRESS HI (RTS)
285	000033	POPJLO==33	;TS04 POPJ ADDRESS LO (RTS)

286 :*****

287 :*****

288 :*****

289

```
290 ; TSSR REGISIER BIT DEFINITIONS.
291
292 001000 TS.XA1==1000 ;EXTENDED ADDRESS BIT 1
293 000400 TS.XA0==400 ;EXTENDED ADDRESS BIT 0
294 000200 TS.SSR==200 ;SUBSYSTEM READY BIT.
295 002000 TS.NBA==2000 ;NEED BUFFER ADRESS-CLEARED BY SET CHAR
296 ; -SET BY COMD WITHOUT SET CHAR
297 020000 TS.SPE==20000 ;SERIAL BUS PARITY ERROR AT TS11
298 177717 FCMASK==177717 ;FATAL CLASS CODE MASK
299 177701 TCFMK==177701 ;TERMINATION AND FATAL CLASSES CODE MASK
300
301 ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
302
303 100000 ACK.C==100000 ;ACKNOWLEDGE BIT
304 040000 CVC.C==40000 ;CLEAR VOLUME CHECK.
305 020000 OPP.C==20000 ;OPPOSITE BIT
306 010000 SWB.C==10000 ;SWAP BYTE BIT
307 004000 MOD.C3==4000 ;MODE BIT 3
308 002000 MOD.C2==2000 ;MODE BIT 2
309 001000 MOD.C1==1000 ;MODE BIT 1
310 000400 MOD.C0==400 ;MODE BIT 0
311 000200 IE.C==200 ;INTERRUPT ENABLE
312 000100 FMT.C1==100 ;FORMAT BIT 1
313 000040 FMT.C0==40 ;FORMAT BIT 0.
314 000020 CMD.C4==20 ;COMMAND BIT 4
315 000010 CMD.C3==10 ;COMMAND BIT 3
316 000004 CMD.C2==4 ;COMMAND BIT 2
317 000002 CMD.C1==2 ;COMMAND BIT 1
318 000001 CMD.C0==1 ;COMMAND BIT 0
319
320 ; BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
321
322 000200 CH.ESS==200
323 000040 CH.EA1==40
324 000020 CH.ERI==20
325
326 ;ROM LOOKUP TABLE BIT DEFINITIONS
327
328 000200 .MULT==200 ;MULT TRACKS
329 000100 .RDFMK==100 ;READ FILE MARK PATTERN
330 000040 .PREAM==40 ;PREAMBLE
331 000020 .9OF9==20 ;9 OF 9 TRACKS
332 000010 .0OF9==10 ;NONE OF 9 TRACKS
333 000004 .CORD==4 ;CORRECTABLE DATA
334 000002 .INCOR==2 ;INCORRECTABLE DATA
335 000001 .8OF9==1 ;8 OF 9 TRACKS
336
337 ; MISCELLANEOUS DEFINITIONS.
338
339 000340 INTPRI==PRI07 ;PRIORITY TO BE USED IN THE INTERRUPT STATE.
340 000010 SCHEXT==10 ;BUFFER LENGTH. (EVEN #)
341 000016 MSGEXT==16 ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
342 000020 DIAEXT==20 ;DIABLK EXTENT IN OCTAL.
343 100006 DIA==ACK.C!CMD.C2!CMD.C1 ;DIA CMD WORD.
344 140004 SCH==ACK.C!CVC.C!CMD.C2 ;SCH (SET CHAR) CMD WORD.
345 100017 GES==ACK.C!CMD.C0!CMD.C1!CMD.C2!CMD.C3 ;GET STATUS COMMAND
```

```
346      020000      BPE==20000      ;XSTAT2, SERIAL BUS PARITY ERROR AT TS04
347      040000      SIP==40000      ;XSTAT2, SILO PARITY ERROR
348      000200      MOT==200        ;XSTAT0, TAPE MOVING
349      177777      ENDTBL==177777 ;END OF A TABLE FLAG
350
351      ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
352      ;IN THE MESSAGE BUFFER.
353
354      000004      MSSRFC==4      ;RESIDUAL FRAME COUNT.
355      000006      MS$XS0==6     ;EXT STATUS REG 0
356      000010      MS$XS1==10    ;EXT STATUS REG 1
357      000012      MS$XS2==12    ;EXT STATUS REG 2
358      000014      MS$XS3==14    ;EXT STATUS REG 3
359      000004      C18$OR==4     ;INDEX FOR OUTPUT READY INFO (CHAN 1-8).
360      000005      C18$1D==5     ;INDEX FOR 1 OR DEAD INFO (CHAN 1-8)
361      000007      C18$TA==7     ;INDEX FOR TRACK ACTIVE INFO. (CHAN 1-8)
362      000010      C18$DA==10    ;INDEX FOR DATA INFO. (CHAN 1-8)
363      000011      C18$TD==11    ;INDEX FOR TRACK DEAD INFO. (CHAN 1-8)
364      000012      C18$OD==12    ;INDEX FOR 0 OR DEAD INFO. (CHAN 1-8)
365      000013      ROM$LK==13    ;INDEX FOR LOOKUP TABLE
366      000014      PRCHST==14    ;STATUS OF THE PARITY CHANNEL (CHANNEL 9)
367      000001      CH9.OR==1     ;BIT POSITION FOR OUTPUT RDY FOR CHAN 9
368      000002      CH9.1D==2     ;BIT POS FOR 1 OR DEAD INFO FOR CHAN 9.
369      000004      CH9.TA==4     ;BIT POSITION OF THE TRACK ACTIVE INFO CHAN 9.
370      000010      CH9.DA==10    ;BIT POS OF DATA INFO FOR CHAN 9.
371      000020      CH9.TD==20    ;BIT POS OF TRK DEAD INFO FOR CHAN 9.
372      000040      CH9.OD==40    ;BIT POS OF 0 OR DEAD INFO FOR CHAN 9.
373
374      ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
375      ;PACKET ENTRIES.
376
377      000000      CP$CMD==0      ;CMDPKT+0==TS04 COMMAND.
378      000002      CP$ADL==2     ;CMDPKT+2==BUFFER ADDRESS LOW.
379      000004      CP$ADH==4     ;CMDPKT+4==BUFFER ADDRESS HIGH.
380      000006      CP$CNT==6     ;CMDPKT+6==BYTE/FILE/RECORD COUNT.
381
```

```

382 ;THE FOLLOWING ARE REGISTER AND BIT DEFINITIONS FOR THE TS04
383 ;REGISTERS OF INTEREST IN THIS DIAGNOSTIC.
384
385 ;*****
386 ;FMCTLO - FORMATTER MAJOR STATE CONTROL REG
387 000004 FMCTLO==4
388 ;WRITE REGISTER 4
389
390 ;THIS REGISTER IS SET UP FOR THE PORTION OF THE
391 ;RECORD WE ARE PRESENTLY READING
392
393
394 000200 FC.RD== 200 ;WE ARE DOING NORMAL READ. IF 0, WE DISABLE SOME ERROR
395 ;CORRECTION LOGIC SO WE'RE MORE DISCRIMINATING
396 ;FOR READ AFTER WRITE.
397
398 000100 FC.FLO==100 ;SETTING THIS BIT CAUSES .FMFLO ON BBUS TO BE TRUE
399 ;(NEEDED BY THE RD PE ROUTINE)
400 000010 FC.DAT==10 ;DATA MODE
401 000004 FC.PRE==4 ;PREAMBLE MODE
402 000002 FC.VCO==2 ;VCO SYNC MODE
403 000001 FC.NRZ==1 ;NRZI MODE (FORCE SKEW WINDOW TO STAY OPEN)
404 ;CAUSES FMTAUI TO BE CLRED WHEN YOU WRITE
405 ;TO THE FMCLDO REGISTER.
406 ;*****
407
408 ;RDCTLO - READ CONTROL REGISTER
409 000020 RDCTLO==20
410 ;WRITE REGISTER 20
411
412 ;THIS REGISTER CONTROLS THE FORMATTER MODE AND THRESHOLD.
413
414 000200 RD.REV==200 ;1 FOR REV MOTION, 0==FWD
415 000100 RD.MAI==100 ;I/O FORMATTER DATA WRAPAROUND
416
417 000040 RD.SPC==40 ;WE ARE SPACING RECORDS. (THIS BIT IS UNUSED IN
418 ;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)
419
420 000020 RD.SKP==20 ;WE ARE SKIPPING FILES. (THIS BIT IS UNUSED IN
421 ;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)
422
423 ;THE FOLLOWING THRESHOLDS ARE AVAILABLE:
424 ;NORM USE DIAG USE
425 000007 RD.110==7 ; HI PREAMP GAIN
426 000006 RD.90==6 ; LO PREAMP, BAD TAPE CLEAN
427 000005 RD.75==5 ; RESIDUAL ERASE CHK
428 000004 RD.68==4 ; FWD/REV AMP BALANCE
429 000003 RD.40==3 ; NRZ WRT
430 000002 RD.20==2 ; NRZ RD, PE WRT
431 000001 RD.12==1 ; PE READ, NRZ ERR RECOV RD/WRT CROSSTALK
432 000000 RD.07==0 ; PE ERR RECOV ERASE FUNCTION
433 ;DATA PORTION ONLY
434

```

```

435 ;*****
436 ;IOSCO - I/O SEQUENCER SILO CONTROL BUFFER OUT
437 000014 IOSCO==14
438 ;WRITE REGISTER 14
439
440 ;THIS REGISTER CONTAINS THE SILO CONTROL BITS FOR DATA WRITTEN
441 ;BY THE MAIN OR I/O MICRO TO THE SILO. THE DATA IN THIS REG
442 ;IS PAIRED WITH THE IOSDO REG AND PUT IN THE SILO WHENEVER
443 ;THE IOSICO REG IS CLOCKED. THIS REGISTER NEED ONLY BE WRITTEN
444 ;ONCE IF THE SAME OLD DATA IS OK TO BE WRITTEN IN THE SILO.
445 ;NOTE THAT IF THE I/O IS WRITING THE SILO, THE MAIN MUST PUT EVEN
446 ;PARITY IN THE BITS 357 OR THE I/O WILL WRITE PAR ERRS IN THE SILO.
447 ;(THE IS.DAP BIT IS DON'T CARE HERE FOR I/O WRITING THE SILO)
448
449 000200 IS.PAR== 200 ;OLD PAR BIT FOR ALL 16 BITS (CNRTL AND DATA)
450 ;NOTE THAT THE BITS MASKED BY 357 MUST BE
451 ;EVEN PARITY BECAUSE THE 9 DATA BITS ARE ODD
452
453 000100 IS.IVP== 100 ;INVERT CNTRL SILO PAR BIT BEFORE MOVING
454 ;TO WRITE BOARD (WRITE EVEN PARITY ON TAPE)
455
456 000040 IS.NRZ== 40 ;INVERT WRITE BUFFER BIT IF ASSOCIATED SILO
457 ;DATA BIT IS A 1. (IF IS.NRZ==0, WE'RE IN PE MODE)
458
459 000020 IS.DAP== 20 ;ODD PARITY FOR THE 8 DATA BITS IN IOSDO
460
461 000010 IS.LRC== 10 ;CAUSES SYNCHRONOUS CLR ON WRT BOARD TO WRITE THE LRC CHA
462 000004 IS.WRF== 4 ;THIS FLG BIT SHOWS UP AT THE WRITE BOARD
463 ;WITH THE CORRESPONDING DATA. THE FUNCTION OF
464 ;THE BIT WILL BE DEFINED BY THE WRITE BOARD.
465 ;NOTE HOWEVER THAT IF THE WRITE BOARD SEES
466 ;THE BIT 1, THE PA.WRF BIT IN THE
467 ;PRATNI REG WILL ALSO BE 1 (IF ENABLED
468 ;TO AFFECT THE ATTN REG).
469 ;THE 2 LOW BITS ARE WRITABLE AND AFFECT THE PARITY TREES BUT
470 ;OTHERWISE ARE UNIMPLEMENTED
471 ;*****
472

```

```

473      .SBTTL  GLOBAL DATA SECTION
474
475      ;++
476      ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
477      ; IN MORE THAN ONE TEST.
478      ;--
479
480      ;      TS04 REGISTER ADDRESSES.
481
482 002214 000000  TSDB:: 0      ;TS04 DATA BUFFER ADDRESS.
483 002216 000000  TSDBHI:: 0     ;TSDB HI BYTE ADDRESS.
484 002220 000000  TSBA:: 0      ;TS04 BUFFER ADDRESS REG ADDRESS.
485 002222 000000  TSBAHI:: 0    ;TSBA HI BYTE ADDRESS.
486 002224 000000  TSSR:: 0     ;TS04 STATUS REGISTER ADDRESS.
487 002226 000000  TSVCT:: 0    ;TS04 VECTOR ADDRESS.
488
489      ;      THE FOLLOWING IS THE TS04 COMMAND PACKET.
490      ;      (MUST BE ON A MODULO 4 BOUNDARY)
491
492      002230      .=.+3&177774      ;FORCES CMD PKT ON MODULO 4 BOUNDARY.
493
494 002230 000000  CMDPKT::0      ;1ST WORD::  TS04 COMMAND
495 002232 000000      0      ;2ND WORD::  BUFFER LOW ADDR.
496 002234 000000      0      ;3RD WORD::  BUFFER HIGH ADDR.
497 002236 000000      0      ;4TH WORD::  BYTE/RECORD/FILE COUNT.
498
499      ;SET CHAR PACKET
500
501      002240      .=.+3&177774
502
503 002240 140004  SCHPKT::SCH      ;SET CHAR CMD HEADER
504 002242 002276      SCHBLK      ;SET CHAR BLK AD LO
505 002244 000000      000000      ;SET CHAR BLK AD HI
506 002246 000010      SCHEXT      ;BPCR=LENGTH OF SCHAR BLOCK
507
508      ;DIAG PACKET
509
510      002250      .=.+3&177774
511
512 002250 100006  DIAPKT::DIA      ;DIAG CMD HEADER
513 002252 002316      DIABLK      ;DIAG BLOCK AD LO
514 002254 000000      000000      ;DIAG BLOCK AD HI
515 002256 000020      DIAEXT      ;BPCR=DIABLK LENGTH
516
517      ;      THIS IS THE MESSAGE PACKET.
518
519 002260 000000  MSGPKT:: 0      ;1ST WORD::  MESSAGE WORD.
520 002262 000000  MSGDFL:: 0     ;2ND WORD::  DATA FIELD LENGTH
521 002264 000000  RFC:: 0      ;3RD WORD::  RESIDUAL FRAME COUNT.
522 002266 000000  XSTAT0:: 0    ;4TH WORD::  EXTENDED STATUS REG 0.
523 002270 000000  XSTAT1:: 0    ;5TH WORD::  EXTENDED STATUS REG 1.
524 002272 000000  XSTAT2:: 0    ;6TH WORD::  EXTENDED STATUS REG 2.
525 002274 000000  XSTAT3:: 0    ;7TH WORD::  EXTENDED STATUS REG 3.
526      002276      MSGEND== .      ;PAST MSG PKT POINTER
527
528      ;      THE SET CHARACTERISTIC BLOCK.

```

```
529
530 002276 002260 SCHBLK:: MSGPKT ;1ST WORD:: MSGPKT ADDR LO.
531 002300 000000 0 ;2ND WORD MSGPKT ADDR HI.
532 002302 000016 MSGEXT ;3RD WORD:: MSG BUFFER LENGTH(BYTES).
533 002304 000000 0 ;4TH WORD:: CHARACTERISTIC WORD.
534 ; ADDRESSES OF INTERRUPT HANDLING ROUTINES.
535
536 002306 014370 TS4INT:: TS4IN0 ;DEVICE 0.
537 002310 014400 TS4IN1 ;DEVICE 1.
538 002312 014410 TS4IN2 ;DEVICE 2.
539 002314 014420 TS4IN3 ;DEVICE 3.
540
541
542 ; THE DIAGNOSTIC COMMAND BUFFER. WHEN A DIAG COMMAND IS
543 ; EXECUTED, THE TS04 PLACES THE CONTENTS OF THIS BUFFER ONTO THE TS04
544 ; STACK IN REVERSE ORDER, THEN EXECUTES WHAT IS ON THE STACK BY
545 ; PERFORMING A COMMAND SIMILAR TO AN RTS.
546 ; THE BUFFER IS INITIALLY LOADED WITH THE TS04 CODE FOR
547 ; AN RTS COMMAND.
548
549
550 002316 000 DIABLK:: .BYTE POPJHI
551 002317 033 .BYTE POPJLO
552 002320 000 .BYTE POPJHI
553 002321 033 .BYTE POPJLO
554 002322 000 .BYTE POPJHI
555 002323 033 .BYTE POPJLO
556 002324 000 .BYTE POPJHI
557 002325 033 .BYTE POPJLO
558 002326 000 .BYTE POPJHI
559 002327 033 .BYTE POPJLO
560 002330 000 .BYTE POPJHI
561 002331 033 .BYTE POPJLO
562 002332 000 .BYTE POPJHI
563 002333 033 .BYTE POPJLO
564 002334 000 .BYTE POPJHI
565 002335 033 .BYTE POPJLO
```



```

566                                     ;TABLES OF FORMATTER AND WRITE CONTROL REG ACTUAL & EXPECTED
567
568 002336      EXPTBL::                ;TABLE 1  EXPECTED REGS
569 002336 000000  EXORDY:: 0          ;EXPECTED OUTPUT READY INFO
570 002340 000000  EX1DTR:: 0         ;EXPECTED 1 OR DEAD TRACK INFO
571 002342 000000  EXTRAC:: 0         ;EXPECTED TRACK ACTIVE INFO
572 002344 000000  EXDATA:: 0         ;EXPECTED DATA INFO
573 002346 000000  EXTRDD:: 0         ;EXPECTED TRACK DEAD INFO
574 002350 000000  EXODTR:: 0         ;EXPECTED 0 OR DEAD TRACK INFO.
575 002352 000000  EXROML:: 0         ;EXPECTED ROM LOOKUP TABLE INFO.
576
577 002354      ACTTBL::                ;TABLE 2  ACTUAL REGS
578 002354 000000  ACORDY:: 0          ;ACTUAL OUTPUT READY INFO
579 002356 000000  AC1DTR:: 0         ;ACTUAL 1'S OR DEAD TRACK INFO
580 002360 000000  ACTRAC:: 0         ;ACTUAL TRACK ACTIVE INFO
581 002362 000000  ACDATA:: 0         ;ACTUAL DATA INFO
582 002364 000000  ACTRDD:: 0         ;ACTUAL TRACK DEAD INFO
583 002366 000000  ACODTR:: 0         ;ACTUAL 0 OR DEAD TRACK INFO
584 002370 000000  ACROML:: 0         ;ACTUAL ROM LOOKUP TABLE INFO
585
586 002372      ORDTBL::                ;TABLE 3  SUBTESTS' ORED ACTUAL REGS
587 002372 000000  ORORDY:: 0         ;OUTPUT READY
588 002374 000000  OR1DTR:: 0         ;1 OR DEAD TRACK
589 002376 000000  ORTRAC:: 0         ;TRACK ACTIVE
590 002400 000000  ORDATA:: 0         ;DATA
591 002402 000000  ORTRDD:: 0         ;TRACK DEAD
592 002404 000000  ORODTR:: 0         ;0 OR DEAD TRACK
593
594 002406 000000  ACTRK1:: 0         ;ACTUAL TRACK ACTIVE TEST 4,SUB 1
595 002410 000000  ACTRK2:: 0         ;ACTUAL TRACK ACTIVE TEST 4,SUB 2
596 002412 000000  ACTRK3:: 0         ;ACTUAL TRACK ACTIVE TEST 4,SUB 3
597 002414 000000  ACTRK4:: 0         ;ACTUAL TRACK ACTIVE TEST 4,SUB 4
598
599 002416 000000  DTKIDN:: 0         ;DEAD TRACK IDENTIFICATION REG. (BITS 0 THRU 8)
600                                     ;1=DEAD; 0=LIVE.
601 002420 000000  ROMLKI:: 0         ;ROM LOOKUP TABLE ADDRESS
602 002422 000000  UNIT:: 0           ;CURRENT UNIT # FOR PRINTS.
603 002424 000000  TS4CL:: 0         ;TS04 MICRO-CODE LEVEL - STORED IN SCH TEST.
604 002426 000000  TIME:: 0          ;TIMEOUT COUNTER.
605 002430 000000  TEMPO:: 0         ;GENERAL PURPOSE LOCATION 0.
606 002432 000000  TEMP1:: 0         ;GENERAL PURPOSE LOCATION 1.
607 002434 000000  TEMP2:: 0         ;GENERAL PURPOSE LOCATION 2.
608
609
610
611                                     ; PROGRAM CONTROL FLAGS.
612
613 002436      000      ERRFLG:: .BYTE 0      ;0 = NO ERROR
614 002437      000      CTLFLG:: .BYTE 0      ;ALL PURPOSE PROGRAM CONTROL FLAG
615                                     .EVEN

```

616
617
618
619
620
621
622
623
624
625

.SBTTL GLOBAL TEXT SECTION

;THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
;MESSAGES, AND ASCII INFORMATION THAT ARE USED IN MORE THAN
;ONE TEST.

; ASCII MESSAGES USED IN ERROR REPORT HEADERS.

.NLIST BEX

002440	051524	032117	041440	DESCM::	.ASCIZ	/TS04 CONTROL LOGIC TEST PROGRAM/
002500	047503	046515	047101	MODUER::	.ASCIZ	/COMMAND PACKET ADDR NOT ON MODULO 4 BOUNDARY: RELOAD!//
002566	051524	032060	047040	SSROFF::	.ASCIZ	/TS04 NOT READY - SSR NOT SET/
002623	124	030123	020064	SSRON::	.ASCIZ	/TS04 DID NOT DROP READY ON COMD ISSUANCE-SSR NOT 0/
002706	042120	030520	020061	WRPER1::	.ASCIZ	/PDP11 - TS11 WRAP FAILURE/
002740	042120	030520	020061	WRPER2::	.ASCIZ	/PDP11 - TS11 WRAP FAIL ON TSSR EXT ADDR BITS/
003015	120	050104	030461	WRPER3::	.ASCIZ	/PDP11 - TS04 WRAP FAILURE/
003047	123	052105	041440	SCHERR::	.ASCIZ	/SET CHARACTERISTIC ERROR/
003100	051124	041501	020113	TAER1::	.ASCIZ	/TRACK ACTIVE NOT 0 FOR 1 OR MORE TRACKS/
003150	051124	041501	020113	TAER2::	.ASCIZ	/TRACK ACTIVE NOT 1 FOR 1 OR MORE TRACKS/
003220	051124	041501	020113	TAERR::	.ASCIZ	/TRACK ACTIVE ERROR/
003243	120	042456	020056	DATER::	.ASCIZ	/P.E. DATA WRAP ERROR/
003270	027120	027105	051440	SKEWER::	.ASCIZ	/P.E. SKEW ERROR/
003310	027120	027105	042040	DDER::	.ASCIZ	/P.E. DEAD TRACK ERROR/
003336	040504	040524	047440	DASKDD::	.ASCIZ	/DATA OR SKEW OR DEAD TRK ERR/
003373	122	046517	046040	ROMER::	.ASCIZ	/ROM LOOKUP TABLE ERROR/
003422	046511	051120	050117	SPECON::	.ASCIZ	/IMPROPER TERMINATION - SPECIAL CONDITION BIT SET/
003503	116	020117	047111	NINTM::	.ASCIZ	/NO INTERRUPT/
003520	047111	042524	051122	UINTM::	.ASCIZ	/INTERRUPT OCCURRED WHEN DISABLED/
003561	115	041511	047522	MICROE::	.ASCIZ	/MICRO DIAGNOSTIC ERROR TSSR:FC=0,TC=7/

.LIST BEX
.EVEN

626

627
628
629

:ASCII BASIC MESSAGES USED IN ERROR REPORTS

```

.NLIST BEX
003630 040445 042523 020105 TERM01:: .ASCIZ /%ASEE OP PANEL%N/
003651 045 044501 020106 TERM02:: .ASCIZ /%AIF UOK LIT THEN BAD TS11 BOARD%N/
003714 051445 022462 042501 TERM03:: .ASCIZ /%S2%AEELSE OP PANEL=MICRO IO ERROR CODE 100-110%N/
003775 045 041501 042117 TERM05:: .ASCIZ /%ACODE LOOP%S10%ADESCRIPTION%S9%AMODULE(SLOT)%N2/
004056 040445 030061 020060 TERM06:: .ASCIZ /%A100 14 IO MICRO SSTEP,IOATN,...%S5%AM8967(12),M8963(11)%N/
004155 045 030501 030460 TERM07:: .ASCIZ /%A101 15 IOCNO REG TEST%S15%AM8967(12)%N/
004231 045 030501 031060 TERM10:: .ASCIZ /%A102 15 FRAME CNTR TEST%S14%AM8966(14)%N/
004306 040445 030061 020063 TERM11:: .ASCIZ /%A103 16 SILO GOOD PAR DATA - WRT FLG%S1%AM8966(14),M8963(11)%N/
004411 045 030501 032060 TERM12:: .ASCIZ /%A104 17 SILO BAD PAR - DATA LATE%S5%AM8966(14)%N/
004476 040445 030061 020065 TERM13:: .ASCIZ /%A105 20 IO LOOP-0'S%S18%AM8965(15)%N/
004547 045 030501 033060 TERM14:: .ASCIZ /%A106 21 IO LOOP-1'S%S18%AM8965(15)%N/
004620 040445 030061 020067 TERM15:: .ASCIZ /%A107 22 IO LOOP-SHIFT LENGTH MUX%S5%AM8965(15)%N/
004705 045 030501 030061 TERM16:: .ASCIZ /%A110 47 SERIAL BUS - TS11 ALIVE%S6%AM8965(15),TS11%N/
004776 051445 034463 040445 TERM17:: .ASCIZ /%S39%AMOTHER BOARD, SBUS CABLE%N/

005037 045 041101 042101 TS11BD:: .ASCIZ /%ABAD TS11 BOARD%N/
005062 040445 040502 020104 FMTCTR:: .ASCIZ /%ABAD FORMATTER CONTROL BOARD M8922 SLOT 7%N/
005136 040445 044103 041505 .ASCIZ /%ACHECK VCO ADJUST ON M8922%N/
005174 040445 040502 020104 FMTCH6:: .ASCIZ /%ABAD FORMATTER CHANNEL BOARD M8924 SLOT 6%N/
005251 045 041101 042101 FMTCH5:: .ASCIZ /%ABAD FORMATTER CHANNEL BOARD M8924 SLOT 5%N/
005326 040445 040502 020104 FMTCH4:: .ASCIZ /%ABAD FORMATTER CHANNEL BOARD M8924 SLOT 4%N/
005403 045 041101 042101 RDCH2:: .ASCIZ /%ABAD READ CHANNEL BOARD G157 SLOT 2%N/
005452 040445 040502 020104 RDCH1:: .ASCIZ /%ABAD READ CHANNEL BOARD G157 SLOT 1%N/
005521 045 041101 042101 RDCHP3:: .ASCIZ /%ABAD READ CHANNEL BOARD M8923 SLOT 3%N/
005570 040445 044103 041505 .ASCIZ /%ACHECK SKEW AND THRESHOLD ADJUSTS ON M8923%N/
005646 040445 051105 047522 IOBRD:: .ASCIZ /%AERROR IN IO BOARDS(SLOT) OR SBUS CABLE%N/
005720 040445 034115 033071 .ASCIZ /%AM8965(15),M8966(14),M8967(12)%N2/
005763 045 047501 020513 CRMSG:: .ASCIZ /%AOK!%N/

```

630
631

005774

.LIST BEX
.EVEN

632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648 005774
649 005774
650 005774
651 005774
652 005774 021127 177777
653 006000 001415
654 006002
655 006002 011137 006040
656 006006
657 006006 013746 006040
658 006012 012746 000001
659 006016 010600
660 006020 104414
661 006022 062706 000004
662 006026
663 006026 062701 000002
664 006032
665 006032 000760
666 006034
667 006034
668 006034 000167
669 006036 000002
670 006040 000000
671 006042
672 006042
673 006042 104423
674
675 006044
676 006044
677 006044
678 006044 012746 005037
679 006050 012746 000001
680 006054 010600
681 006056 104414
682 006060 062706 000004
683 006064
684 006064 013746 002422
685 006070 013746 002432
686 006074 010346
687 006076 012746 006122

.SBTTL GLOBAL ERROR REPORT SECTION

```

; ++
; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
; THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
; --
    
```

```

; LONG MESSAGE PRINT SUBR
; ENTER WITH R1=AD OF MSG AD TABLE
; MSG AD TABLE CONTAINS AD OF MSGS TO PRINT, ONE .WORD AD
; PER MSG. THE TABLE MUST END WITH .WORD ENDTBL=177777
; AS END OF TABLE FLAG
    
```

```

BGNMSG LONMSG
LONMSG: WHILE (R1) NE #177777 DO
    
```

```

        LET MSGADR := (R1)
        PRINTB      MSGADR
    
```

```

        LET R1 := R1 + #2
    
```

```

ENDDO
    
```

```

EXIT    MSG
    
```

```

MSGADR: .WORD 0
        ENDMSG
    
```

```

L10002:
    
```

```

BGNMSG WRAPR1
WRAPR1: PRINTB #TS11BD
    
```

```

PRINTX #EEM1A,R3,TEMP1,UNIT
    
```

```

50000$:
    
```

```

        CMP      (R1),#177777
        BEQ      50001$
    
```

```

        MOV      (R1),MSGADR
    
```

```

        MOV      MSGADR,-(SP)
    
```

```

        MOV      #1,-(SP)
    
```

```

        MOV      SP,R0
    
```

```

        TRAP    C$PNTB
    
```

```

        ADD     #4,SP
    
```

```

        ADD     #2,R1
    
```

```

        BR      50000$
    
```

```

50001$:
    
```

```

        .WORD   JSJMP
        .WORD   L10002-2-
    
```

```

        TRAP    C$MSG
    
```

```

        MOV      #TS11BD,-(SP)
    
```

```

        MOV      #1,-(SP)
    
```

```

        MOV      SP,R0
    
```

```

        TRAP    C$PNTB
    
```

```

        ADD     #4,SP
    
```

```

        MOV      UNIT,-(SP)
    
```

```

        MOV      TEMP1,-(SP)
    
```

```

        MOV      R3,-(SP)
    
```

```

        MOV      #EEM1A,-(SP)
    
```

688	006102	012746	000004						MOV	#4,-(SP)
689	006106	010600							MOV	SP,R0
690	006110	104415							TRAP	C\$PNTX
691	006112	062706	000012						ADD	#12,SP
692	006116				EXIT	MSG				
693	006116	000167							.WORD	J\$JMP
694	006120	000074							.WORD	L10003-2-
695										
	006122	040445	051524	040502	EEM1A:	.NLIST	BEX			
	006150	040445	051524	040502		.ASCII	/%ATSBA EXPECTED=%B16%N/			
						.ASCII	/%ATSBA ACTUAL=%S2%B16%S3%AUNIT:%D1%N2/			
						.LIST	BEX			
						.EVEN				
						ENDMSG				
696										
697	006216				L10003:					
698	006216									
699	006216	104423							TRAP	C\$MSG
700										
701	006220					BGNMSG	WRAPR2			
702	006220				WRAPR2::					
703	006220					PRINTB	#IOBRD			
704	006220	012746	005646						MOV	#IOBRD,-(SP)
705	006224	012746	000001						MOV	#1,-(SP)
706	006230	010600							MOV	SP,R0
707	006232	104414							TRAP	C\$PNTB
708	006234	062706	000004						ADD	#4,SP
709	006240					PRINTX	#EEM2A,R4,TEMP1,UNIT			
710	006240	013746	002422						MOV	UNIT,-(SP)
711	006244	013746	002432						MOV	TEMP1,-(SP)
712	006250	010446							MOV	R4,-(SP)
713	006252	012746	006276						MOV	#EEM2A,-(SP)
714	006256	012746	000004						MOV	#4,-(SP)
715	006262	010600							MOV	SP,R0
716	006264	104415							TRAP	C\$PNTX
717	006266	062706	000012						ADD	#12,SP
718	006272					EXIT	MSG			
719	006272	000167							.WORD	J\$JMP
720	006274	000072							.WORD	L10004-2-
721										
	006276	040445	051524	040502	EEM2A:	.NLIST	BEX			
	006323	045	052101	041123		.ASCII	/%ATSBA EXPECTED=%B8%N/			
						.ASCII	/%ATSBA ACTUAL=%S2%B8%S3%AUNIT:%D1%N2/			
						.LIST	BEX			
						.EVEN				
						ENDMSG				
722										
723	006370				L10004:					
724	006370									
725	006370	104423							TRAP	C\$MSG
726										
727	006372					BGNMSG	WRAPR3			
728	006372				WRAPR3::					
729	006372					PRINTB	#IOBRD			
730	006372	012746	005646						MOV	#IOBRD,-(SP)
731	006376	012746	000001						MOV	#1,-(SP)
732	006402	010600							MOV	SP,R0
733	006404	104414							TRAP	C\$PNTB
734	006406	062706	000004						ADD	#4,SP
735	006412					PRINTX	#EEM3A,R4,TEMP2,UNIT			
736	006412	013746	002422						MOV	UNIT,-(SP)
737	006416	013746	002434						MOV	TEMP2,-(SP)

738 006422 010446
 739 006424 012746 006450
 740 006430 012746 000004
 741 006434 010600
 742 006436 104415
 743 006440 062706 000012
 744 006444
 745 006444 000167
 746 006446 000072
 747

006450 040445 051524 051123
 006475 045 052101 051523

EEM3A: .NLIST BEX
 .ASCII /%ATSSR EXPECTED=%B8%N/
 .ASCIZ /%ATSSR ACTUAL=%S2%B8%S3%AUNIT:%D1%N2/
 .LIST BEX
 .EVEN
 ENDMSG

748
 749 006542
 750 006542
 751 006542 104423
 752

L10005:

WRAPR4: BGNMSG WRAPR4
 PRINTB #TS11BD

753 006544
 754 006544
 755 006544
 756 006544 012746 005037
 757 006550 012746 000001
 758 006554 010600
 759 006556 104414
 760 006560 062706 000004
 761 006564
 762 006564 013746 002422
 763 006570 013746 002434
 764 006574 010246
 765 006576 012746 006622
 766 006602 012746 000004
 767 006606 010600
 768 006610 104415
 769 006612 062706 000012
 770 006616
 771 006616 000167
 772 006620 000072
 773

006622 040445 051524 051123
 006647 045 052101 051523

EEM4A: .NLIST BEX
 .ASCII /%ATSSR EXPECTED=%B8%N/
 .ASCIZ /%ATSSR ACTUAL=%S2%B8%S3%AUNIT:%D1%N2/
 .LIST BEX
 .EVEN
 ENDMSG

774
 775 006714
 776 006714
 777 006714 104423
 778

L10006:

TAEM: BGNMSG TAEM
 PRINTX #TAEMA,UNIT

779 006716
 780 006716
 781 006716
 782 006716 013746 002422
 783 006722 012746 007016
 784 006726 012746 000002
 785 006732 010600
 786 006734 104415
 787 006736 062706 000006

MOV R4,-(SP)
 MOV #EEM3A,-(SP)
 MOV #4,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #12,SP

.WORD JSJMP
 .WORD L10005-2-

TRAP C\$MSG

MOV #TS11BD,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #4,SP

MOV UNIT,-(SP)
 MOV TEMP2,-(SP)
 MOV R2,-(SP)
 MOV #EEM4A,-(SP)
 MOV #4,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #12,SP

.WORD JSJMP
 .WORD L10006-2-

TRAP C\$MSG

MOV UNIT,-(SP)
 MOV #TAEMA,-(SP)
 MOV #2,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #6,SP

788	006742				PRINTX #TAEMB,EXTRAC		
789	006742	013746	002342			MOV	EXTRAC,-(SP)
790	006746	012746	007067			MOV	#TAEMB,-(SP)
791	006752	012746	000002			MOV	#2,-(SP)
792	006756	010600				MOV	SP,R0
793	006760	104415				TRAP	C\$PNTX
794	006762	062706	000006			ADD	#6,SP
795	006766				PRINTX #TAEMC,ACTRAC		
796	006766	013746	002360			MOV	ACTRAC,-(SP)
797	006772	012746	007123			MOV	#TAEMC,-(SP)
798	006776	012746	000002			MOV	#2,-(SP)
799	007002	010600				MOV	SP,R0
800	007004	104415				TRAP	C\$PNTX
801	007006	062706	000006			ADD	#6,SP
802	007012				EXIT MSG		
803	007012	000167				.WORD	J\$JMP
804	007014	000156				.WORD	L10007-2-
805					.NLIST BEX		
	007016	040445	051124	041501	TAEMA: .ASCIZ		/%ATRACK:%S9%S6%AP76543210%S3%AUNIT:%D1%N/
	007067	045	052101	045522	TAEMB: .ASCIZ		/%ATRK ACT EXP DATA:%S4%B9%N/
	007123	045	052101	045522	TAEMC: .ASCIZ		/%ATRK%S%AACT%S%AACTUAL%S%ADATA:%S%B9%N2/
					.LIST BEX		
					.EVEN		
					ENDMSG		
806		007174					
807	007174				L10007:		
808	007174						
809	007174	104423				TRAP	C\$MSG
810							
811	007176				BGNMSG SKDAEM		
812	007176				SKDAEM::		
813	007176				PRINTX #DAEM,UNIT		
814	007176	013746	002422			MOV	UNIT,-(SP)
815	007202	012746	007406			MOV	#DAEM,-(SP)
816	007206	012746	000002			MOV	#2,-(SP)
817	007212	010600				MOV	SP,R0
818	007214	104415				TRAP	C\$PNTX
819	007216	062706	000006			ADD	#6,SP
820	007222				PRINTX #DAEMA		
821	007222	012746	007461			MOV	#DAEMA,-(SP)
822	007226	012746	000001			MOV	#1,-(SP)
823	007232	010600				MOV	SP,R0
824	007234	104415				TRAP	C\$PNTX
825	007236	062706	000004			ADD	#4,SP
826	007242				PRINTX #DAEMB,EXORDY,ACORDY,EX1DTR,AC1DTR		
827	007242	013746	002356			MOV	AC1DTR,-(SP)
828	007246	013746	002340			MOV	EX1DTR,-(SP)
829	007252	013746	002354			MOV	ACORDY,-(SP)
830	007256	013746	002336			MOV	EXORDY,-(SP)
831	007262	012746	007530			MOV	#DAEMB,-(SP)
832	007266	012746	000005			MOV	#5,-(SP)
833	007272	010600				MOV	SP,R0
834	007274	104415				TRAP	C\$PNTX
835	007276	062706	000014			ADD	#14,SP
836	007302				PRINTX #DAEMC,EXTRAC,ACTRAC,EXDATA,ACDATA		
837	007302	013746	002362			MOV	ACDATA,-(SP)
838	007306	013746	002344			MOV	EXDATA,-(SP)
839	007312	013746	002360			MOV	ACTRAC,-(SP)

840 007316 013746 002342
841 007322 012746 007617
842 007326 012746 000005
843 007332 010600
844 007334 104415
845 007336 062706 000014
846 007342
847 007342 013746 002366
848 007346 013746 002350
849 007352 013746 002364
850 007356 013746 002346
851 007362 012746 007701
852 007366 012746 000005
853 007372 010600
854 007374 104415
855 007376 062706 000014
856 007402
857 007402 000167
858 007404 000362
859

PRINTX #DAEMD,EXTRDD,ACTRDD,EXODTR,ACODTR

EXIT MSG

MOV EXTRAC,-(SP)
MOV #DAEMC,-(SP)
MOV #5,-(SP)
MOV SP,RO
TRAP C\$PNTX
ADD #14,SP

MOV ACODTR,-(SP)
MOV EXODTR,-(SP)
MOV ACTRDD,-(SP)
MOV EXTRDD,-(SP)
MOV #DAEMD,-(SP)
MOV #5,-(SP)
MOV SP,RO
TRAP C\$PNTX
ADD #14,SP

.WORD J\$JMP
.WORD L10010-2-

007406 051445 022471 032523 DAEM: .NLIST BEX
007461 045 052101 040522 DAEMA: .ASCIZ /%S9%S5%AEEXPECTED%S7%AACTUAL%S3%AUNIT:%D1%/

007530 040445 052517 050124 DAEMB: .ASCIZ /%ATRACK:%S8%AP76543210%S5%AP76543210%/

007564 040445 020061 051117 .ASCIZ /%AOUTPUT READY:%S%B9%S5%B9%/

007617 045 052101 040522 DAEMC: .ASCIZ /%A1 OR DEAD:%S4%B9%S5%B9%/

007653 045 042101 052101 .ASCIZ /%ATRACK ACTIVE:%S%B9%S5%B9%/

007701 045 052101 040522 DAEMD: .ASCIZ /%ADATA:%S9%B9%S5%B9%/

007734 040445 020060 051117 .ASCIZ /%A0 OR DEAD:%S4%B9%S5%B9%N2/

.LIST BEX

.EVEN

ENDMSG

860
861 007770
862 007770
863 007770 104423
864
865 007772
866 007772
867 007772
868 007772 013746 002422
869 007776 013746 002370
870 010002 013746 002352
871 010006 013746 002420
872 010012 012746 010036
873 010016 012746 000005
874 010022 010600
875 010024 104415
876 010026 062706 000014
877 010032
878 010032 000167
879 010034 000102
880

L10010:

ROMEM: BGNMSG ROMEM

PRINTX #ROMEMA,ROMLKI,EXROML,ACROML,UNIT

TRAP C\$MSG

MOV UNIT,-(SP)
MOV ACROML,-(SP)
MOV EXROML,-(SP)
MOV ROMLKI,-(SP)
MOV #ROMEMA,-(SP)
MOV #5,-(SP)
MOV SP,RO
TRAP C\$PNTX
ADD #14,SP

.WORD J\$JMP
.WORD L10011-2-

010036 040445 042101 051104 ROMEMA: .NLIST BEX
010057 045 042501 050130 .ASCIZ /%ADDRESS=%S%04%/

010077 045 040501 052103 .ASCIZ /%AEEXPECTED=%03%/

.ASCIZ /%AACTUAL=%S2%03%S3%AUNIT:%D1%N2/

.LIST BEX

.EVEN

ENDMSG

881 010140
882 010140


```

883 010140 L10011:
884 010140 104423 TRAP C$MSG
885
886 010142 040445 051524 051123 TERMA: .NLIST BEX
      .ASCIZ /%ATSSR =%S%06%S3%AUNIT:%D1%N/
      .LIST BEX
      .EVEN
887 010200
888
889 ;SET CHAR ERR MSGS SCHER1-SCHETEMPO
890
891 010200 BGNMSG SCHER1
892 010200 SCHER1::
893 010200 PRINTB #TS11BD
894 010200 012746 005037 MOV #TS11BD,-(SP)
895 010204 012746 000001 MOV #1,-(SP)
896 010210 010600 MOV SP,R0
897 010212 104414 TRAP C$PNTB
898 010214 062706 000004 ADD #4,SP
899 010220 PRINTX #SCHERA
900 010220 012746 010244 MOV #SCHERA,-(SP)
901 010224 012746 000001 MOV #1,-(SP)
902 010230 010600 MOV SP,R0
903 010232 104415 TRAP C$PNTX
904 010234 062706 000004 ADD #4,SP
905 010240 EXIT MSG
906 010240 000167 .WORD JSJMP
907 010242 000064 .WORD L10012-2-.
908 010244 040445 051524 051123 SCHERA: .NLIST BEX
      .ASCIZ /%ATSSR NBA NOT SET ON COMD BEFORE SET CHAR ISSUED%N/
      .LIST BEX
      .EVEN
      ENDMSG
909
910 010330 L10012:
911 010330
912 010330 104423 TRAP C$MSG
913
914 010332 BGNMSG SCHER2
915 010332 SCHER2::
916 010332 PRINTB #TS11BD
917 010332 012746 005037 MOV #TS11BD,-(SP)
918 010336 012746 000001 MOV #1,-(SP)
919 010342 010600 MOV SP,R0
920 010344 104414 TRAP C$PNTB
921 010346 062706 000004 ADD #4,SP
922 010352 PRINTX #SCHERB
923 010352 012746 010376 MOV #SCHERB,-(SP)
924 010356 012746 000001 MOV #1,-(SP)
925 010362 010600 MOV SP,R0
926 010364 104415 TRAP C$PNTX
927 010366 062706 000004 ADD #4,SP
928 010372 EXIT MSG
929 010372 000167 .WORD JSJMP
930 010374 000046 .WORD L10013-2-.
931 010376 040445 051524 051123 SCHERB: .NLIST BEX
      .ASCIZ /%ATSSR NBA NOT CLEARED BY SET CHAR%N/
      .LIST BEX
      .EVEN
932 010444

```



```

984
985 010706          BGNMSG  SCHER5
986 010706          SCHER5::
987 010706          PRINTB  #SCHERE
988 010706 012746 010752          MOV    #SCHERE,-(SP)
989 010712 012746 000001          MOV    #1,-(SP)
990 010716 010600          MOV    SP,R0
991 010720 104414          TRAP   C$PNTB
992 010722 062706 000004          ADD    #4,SP
993 010726          PRINTX  #SCHERF
994 010726 012746 011036          MOV    #SCHERF,-(SP)
995 010732 012746 000001          MOV    #1,-(SP)
996 010736 010600          MOV    SP,R0
997 010740 104415          TRAP   C$PNTX
998 010742 062706 000004          ADD    #4,SP
999 010746          EXIT    MSG
1000 010746 000167          .WORD  JSJMP
1001 010750 000146          .WORD  L10016-2-.
1002
    010752 040445 051524 030461  SCHERE: .NLIST  BEX
    011036 040445 042523 044522  SCHERF: .ASCIZ  /%ATS11 BOARD OR SERIAL BUS CABLE OR M8965 SLOT 15%N/
    .ASCIZ  /%ASERIAL BUS PARITY ERROR  SPE OR BPE TSSR FC=2%N/
    .LIST  BEX
    .EVEN
    ENDMSG
1003
1004 011120          L10016:
1005 011120
1006 011120 104423          TRAP   C$MSG
1007
1008 011122          BGNMSG  SCHER6
1009 011122          SCHER6::
1010 011122          PRINTB  #SCHERG
1011 011122 012746 011166          MOV    #SCHERG,-(SP)
1012 011126 012746 000001          MOV    #1,-(SP)
1013 011132 010600          MOV    SP,R0
1014 011134 104414          TRAP   C$PNTB
1015 011136 062706 000004          ADD    #4,SP
1016 011142          PRINTX  #SCHERH
1017 011142 012746 011210          MOV    #SCHERH,-(SP)
1018 011146 012746 000001          MOV    #1,-(SP)
1019 011152 010600          MOV    SP,R0
1020 011154 104415          TRAP   C$PNTX
1021 011156 062706 000004          ADD    #4,SP
1022 011162          EXIT    MSG
1023 011162 000167          .WORD  JSJMP
1024 011164 000070          .WORD  L10017-2-.
1025
    011166 040445 034115 033071  SCHERG: .NLIST  BEX
    011210 040445 044523 047514  SCHERH: .ASCIZ  /%AM8966 SLOT 14%N/
    .ASCIZ  /%ASILO PARITY ERROR  SIP TSSR FC=2%N/
    .LIST  BEX
    .EVEN
    ENDMSG
1026
1027 011256          L10017:
1028 011256
1029 011256 104423          TRAP   C$MSG
1030
1031
1032 011260          BGNMSG  SCHER7
1033 011260          SCHER7::

```

```

1034 011260          PRINTB #SCHERI
1035 011260 012746 011350          MOV #SCHERI,-(SP)
1036 011264 012746 000001          MOV #1,-(SP)
1037 011270 010600          MOV SP,R0
1038 011272 104414          TRAP C$PNTB
1039 011274 062706 000004          ADD #4,SP
1040 011300          LET R2 := #PCHLTB
1041 011300 012702 011436          MOV #PCHLTB,R2
1042 011304          WHILE (R2) NE #177777 DO
1043 011304
1044 011304 021227 177777          50002$: CMP (R2),#177777
1045 011310 001415          BEQ 50003$
1046 011312          LET PCHTAD := (R2)
1047 011312 011237 011432          MOV (R2),PCHTAD
1048 011316          PRINTB PCHTAD
1049 011316 013746 011432          MOV PCHTAD,-(SP)
1050 011322 012746 000001          MOV #1,-(SP)
1051 011326 010600          MOV SP,R0
1052 011330 104414          TRAP C$PNTB
1053 011332 062706 000004          ADD #4,SP
1054 011336          LET R2 := R2 + #2
1055 011336 062702 000002          ADD #2,R2
1056 011342          ENDDO
1057 011342 000760          BR 50002$
1058 011344          50003$:
1059 011344          EXIT MSG          .WORD JSJMP
1060 011344 000167          .WORD L10020-2-.
1061 011346 000064
1062 011350 040445 051524 051123 SCHERI: .NLIST BEX
          .ASCIZ /%ATSSR FC=2 FATAL MICRO ERR HALTS PC=1750-1777%N2/
          .LIST BEX
1063 011432 000000          PCHTAD: .WORD 0
1064 011434          .EVEN
1065 011434          .WORD 0
1066 011434          ENDMSG
1067 011434 104423          L10020: TRAP C$MSG
1068
1069 011436 011530 011624 011674 PCHLTB: .WORD PCHDR,PCHDRA,PCHDRB
1070 011444 011736 012015 012057 .WORD PCHT50,PCHT51,PCHT52
1071 011452 012133 012215 012277 .WORD PCHT53,PCHT54,PCHT55
1072 011460 012342 012403 012444 .WORD PCHT56,PCHT57,PCHT60
1073 011466 012512 012554 012624 .WORD PCHT61,PCHT62,PCHT63
1074 011474 012702 012760 013025 .WORD PCHT64,PCHT65,PCHT66
1075 011502 013073 013141 013213 .WORD PCHT67,PCHT70,PCHT71
1076 011510 013275 013353 013435 .WORD PCHT72,PCHT73,PCHT74
1077 011516 013511 013600 013665 .WORD PCHT75,PCH75A,PCHT76,PCHT77
1078 011524 013747
1079 011526 177777          .WORD ENDTBL
1080
1081          .NLIST BEX
          011530 040445 041520 042440 PCHDR: .ASCIZ /%APC ERROR HALTS BAD MODULES: M8962,M8963,M8964,OR M8967%N/
          011624 051445 031462 040445 PCHDRA: .ASCIZ /%S23%ASLOT: 8 11 9 12%N/
          011674 040445 050040 022503 PCHDRB: .ASCIZ /%A PC%S17%ACOMMENT%S13%AMODULE%N2/
          011736 040445 033461 030065 PCHT50: .ASCIZ /%A1750 OPERATION IO PROBLEM%S14%AM8967,TS11%N/
          012015 045 030501 032467 PCHT51: .ASCIZ /%A1751 SPURIOUS ATTN%S22%AM8963%N/

```

```

012057      045 030501 032467 PCHT52: .ASCIZ  /%A1752 STK PAR ERR ON OVERFLOW%S12%AM8963%N/
012133      045 030501 032467 PCHT53: .ASCIZ  /%A1753 STK NOT EMPTY AT END OF TASK      M8963%N/
012215      045 030501 032467 PCHT54: .ASCIZ  /%A1754 STK PTR REG WON'T HOLD DATA      M8963%N/
012277      045 030501 032467 PCHT55: .ASCIZ  /%A1755 UTSTM - BRANCH%S21%AM8962%N/
012342     040445 033461 033065 PCHT56: .ASCIZ  /%A1756 UTSTM - ZBIT%S23%AM8962%N/
012403      045 030501 032467 PCHT57: .ASCIZ  /%A1757 UTSTM - NBIT%S23%AM8962%N/
012444     040445 033461 030066 PCHT60: .ASCIZ  /%A1760 UTSTM - NOT Z BIT%S18%AM8962%N/
012512     040445 033461 030466 PCH161: .ASCIZ  /%A1761 UTSTM - C BIT%S22%AM8962%N/
012554     040445 033461 031066 PCHT62: .ASCIZ  /%A1762 UTSTM - PCB LO TEST%S16%AM8962%N/
012624     040445 033461 031466 PCHT63: .ASCIZ  /%A1763 UTSTM - INTERNAL REG ERROR%S9%AM8962%N/
012702     040445 033461 032066 PCHT64: .ASCIZ  /%A1764 UTSTM - INTERNAL REG ERROR%S9%AM8962%N/
012760     040445 033461 032466 PCHT65: .ASCIZ  /%A1765 UTSTM - ADD TEST%S19%AM8962%N/
013025      045 030501 033067 PCHT66: .ASCIZ  /%A1766 UTSTM - A SUBTEST%S18%AM8962%N/
013073      045 030501 033067 PCH167: .ASCIZ  /%A1767 UTSTM - BSUB TEST%S18%AM8962%N/
013141      045 030501 033467 PCHT70: .ASCIZ  /%A1770 UTSTM - SHIFT LR TEST%S14%AM8962%N/
013213      045 030501 033467 PCHT71: .ASCIZ  /%A1771 UTSTM - LOGICAL:AND,OR,XOR,NAND      M8962%N/
013275      045 030501 033467 PCHT72: .ASCIZ  /%A1772 STAKM - STACK PARITY TEST%S10%AM8963%N/
013353      045 030501 033467 PCHT73: .ASCIZ  /%A1773 STAKM - STACK UNDER-OVERFLOW      M8963%N/
013435      045 030501 033467 PCHT74: .ASCIZ  /%A1774 STAKM - STACK DATA TEST%S12%AM8963%N/
013511      045 030501 033467 PCHT75: .ASCIZ  /%A1775 STAKM - CAPSTAN BUS DATA TEST      M8963,G159%N/
013600     040445 044103 041505 PCH75A: .ASCIZ  /%ACHECK TACH PHASE,DECEL AND SPEED ADJUSTS ON G159%N/
013665      045 030501 033467 PCHT76: .ASCIZ  /%A1776 STAKM - CBUS BRANCH TEST      M8963%N/
013747      045 030501 033467 PCHT77: .ASCIZ  /%A1777 STAKM - SIMULATED LIMIT ATTN TEST      M8963%N/
    
```

```

1082      014032      .LIST      BEX
1083     014032      .EVEN
1084     014032      BGNMSG      SCHERO
1085     014032      SCHERO::
1086     014032     012746     014076      PRINTB      #SCHERL
1087     014036     012746     000001      MOV          #SCHERL,-(SP)
1088     014042     010600      MOV          #1,-(SP)
1089     014044     104414      MOV          SP,R0
1090     014046     062706     000004      TRAP        C$PNTB
1091     014052      PRINTX      #SCHERM
1092     014052     012746     014121      ADD         #4,SP
1093     014056     012746     000001      MOV          #SCHERM,-(SP)
1094     014062     010600      MOV          #1,-(SP)
1095     014064     104415      MOV          SP,R0
1096     014066     062706     000004      TRAP        C$PNTX
1097     014072      EXIT        MSG
1098     014072     000167      ADD         #4,SP
1099     014074     000042      .WORD       JSJMP
1100      014076     040445     041501     046040      .WORD       L10021-2-.
1101      014121      045      052101     051523      SCHERL: .ASCIZ  /%AAC LOW AT TSO4%N/
1102      014140      SCHERM: .ASCIZ  /%ATSSR FC=3%N/
1103      014140      .LIST      BEX
1104     014140     104423      .EVEN
1105      014140      ENDMSG
1106      014140      L10021:
1107     014142      TRAP        C$MSG
1108     014142      BGNMSG      ROMLER
1109      014142      ROMLER::
    
```

```

1109 014142          PRINTB #FMTCTR
1110 014142 012746 005062          MOV #FMTCTR,-(SP)
1111 014146 012746 000001          MOV #1,-(SP)
1112 014152 010600          MOV SP,R0
1113 014154 104414          TRAP C$PNTB
1114 014156 C52706 000004          ADD #4,SP
1115 014162          ENDMSG
1116 014162          L10022:
1117 014162 104423          TRAP C$MSG
1118
1119 014164          BGNMSG MICERR
1120 014164          MICERR::
1121 014164          PRINTB #MICERA
1122 014164 012746 014230          MOV #MICERA,-(SP)
1123 014170 012746 000001          MOV #1,-(SP)
1124 014174 010600          MOV SP,R0
1125 014176 104414          TRAP C$PNTB
1126 014200 062706 000004          ADD #4,SP
1127 014204          PRINTX #MICERB
1128 014204 012746 014304          MOV #MICERB,-(SP)
1129 014210 012746 000001          MOV #1,-(SP)
1130 014214 010600          MOV SP,R0
1131 014216 104415          TRAP C$PNTX
1132 014220 062706 000004          ADD #4,SP
1133 014224          EXIT MSG
1134 014224 000167          .WORD J$JMP
1135 014226 000136          .WORD L10023-2-.
1136
1136 014230 040445 047525 020113 MICERA: .NLIST BEX
1136 014304 040445 047503 042504 MICERB: .ASCIZ /%AUOK LIGHT OFF,XTAT3=OP PANEL=ERROR CODE%/
1136          .ASCIZ /%ACODE%S10%AERROR%S28%AMODULE(SLOT) SCOPE LOOP%N2/
1136          .LIST BEX
1136          .EVEN
1136          ENDMSG
1137
1138 014366          L10023:
1139 014366
1140 014366 104423          TRAP C$MSG
1141
1142
1143

```

```

1144
1145      .SBTTL  GLOBAL SUBROUTINES SECTION
1146
1147      :++
1148      : THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
1149      : THAT ARE USED IN MORE THAN ONE TEST.
1150      :--
1151
1152      ;      MODULES TO HANDLE TS04 INTERRUPTS.
1153
1154      014370      BGNSRV  TS4IN0      ;DEVICE 0.
1155      014370      TS4IN0::
1156      014370      112737 000001 002437      LET CTLFLG :B= #1      ;SET INTERRUPT OCCURRED FLAG.
1157      014370      112737 000001 002437      MOV#      #1,CTLFLG
1158      014376      ENDSRV
1159      014376      L10024:
1160      014376      000002      RTI
1161
1162      014400      BGNSRV  TS4IN1      ;DEVICE 1.
1163      014400      TS4IN1::
1164      014400      112737 000001 002437      LET CTLFLG :B= #1      ;SET INTERRUPT OCCURRED FLAG.
1165      014400      112737 000001 002437      MOV#      #1,CTLFLG
1166      014406      ENDSRV
1167      014406      L10025:
1168      014406      000002      RTI
1169
1170      014410      BGNSRV  TS4IN2      ;DEVICE 2.
1171      014410      TS4IN2::
1172      014410      112737 000001 002437      LET CTLFLG :B= #1      ;SET INTERRUPT OCCURRED FLAG.
1173      014410      112737 000001 002437      MOV#      #1,CTLFLG
1174      014416      ENDSRV
1175      014416      L10026:
1176      014416      000002      RTI
1177
1178      014420      BGNSRV  TS4IN3      ;DEVICE 3.
1179      014420      TS4IN3::
1180      014420      112737 000001 002437      LET CTLFLG :B= #1      ;SET INTERRUPT OCCURRED FLAG.
1181      014420      112737 000001 002437      MOV#      #1,CTLFLG
1182      014426      ENDSRV
1183      014426      L10027:
1184      014426      000002      RTI

```

```

1185      :      SUBROUTINE TO WAIT UP TO 1 MINUTE FOR THE SSR BIT
1186      :      TO SET - REPORT AN ERROR IF IT DOESN'T SET AND ATTEMPT TO DROP UNIT
1187
1188 014430 005037 002426      WT4SSR:: CLR      TIME      ;INIT TIMEOUT COUNTER.
1189 014434      1$:      BREAK      ;GO TO SUPERVISOR.
1190 014434 104422      TRAP      C$BRK
1191 014436 105777 165562      TSTB      @TSSR      ;READY?
1192 014442 100432      BMI      2$      ;BR IF SO
1193 014444 005337 002426      DEC      TIME      ;WAIT...
1194 014450 001371      BNE      1$      ;DONE WAITING? BR IF NOT.
1195 014452 017737 165546 002434      MOV      @TSSR,TEMP2 ;GET TSSR
1196 014460 012701 014532      MOV      #TERMTB,R1 ;
1197 014464      ERRDF      2,SSROFF,LONMSG ;IF SO, REPORT AN ERR IF SSR NOT SET YET.
1198 014464 104455      TRAP      C$ERDF
1199 014466 000002      .WORD      2
1200 014470 002566      .WORD      SSROFF
1201 014472 005774      .WORD      LONMSG
1202 014474      PRINTX      #TERMA,TEMP2,UNIT      ;PRINT TSSR CONTENTS
1203 014474 013746 002422      MOV      UNIT,-(SP)
1204 014500 013746 002434      MOV      TEMP2,-(SP)
1205 014504 012746 010142      MOV      #TERMA,-(SP)
1206 014510 012746 000003      MOV      #3,-(SP)
1207 014514 010600      MOV      SP,R0
1208 014516 104415      TRAP      C$PNTX
1209 014520 062706 000010      ADD      #10,SP
1210 014524      CALL      DROPU      ;ATTEMPT TO DROP UNIT
1211 014524 004737 014570      JSR      PC,DROPU
1212      ;IF DROPPED PROG ABORTS CURRENT PASS
1213      ;IF INHIBIT DROP FLAG SET BY OPERATOR (IDU)
1214      ;THEN PROGRAM CONTINUES
1215 014530 000207      2$:      RTS      PC      ;RETURN.
1216
1217 014532 003630 003651 003714      TERMTB: .WORD      TERM01,TERM02,TERM03
1218 014540 003775 004056      .WORD      TERM05,TERM06
1219 014544 004155 004231 004306      .WORD      TERM07,TERM10,TERM11
1220 014552 004411 004476 004547      .WORD      TERM12,TERM13,TERM14
1221 014560 004620 004705 004776      .WORD      TERM15,TERM16,TERM17
1222 014566 177777      .WORD      ENDTBL
1223
1224      ;      SUBR TO ATTEMPT TO DROP A UNIT FLAGGED WITH A DEVICE FATAL ERROR
1225
1226 014570      DROPU:: DODU LUNIT      ;EXECUTE DROP MACRO
1227 014570 013700 027574      MOV      LUNIT,R0
1228 014574 104451      TRAP      C$DODU
1229 014576      GPHARD      LUNIT,TIME      ;IS UNIT DROPPED
1230 014576 013700 027574      MOV      LUNIT,R0
1231 014602 104442      TRAP      C$GPHRD
1232 014604 010037 002426      MOV      R0,TIME
1233 014610      BCOMPLETE      1$      ;NO RETURN TO CALLER TO PROCEED
1234 014610 103401      BCS      1$
1235 014612      DOCLN      ;YES EXEC CEAN-UP CODE TO START NEW PASS
1236 014612 104444      TRAP      C$DCLN
1237 014614 000207      1$:      RTS      PC
1238
1239      ;
1240      SUBR TO WAIT AN EXTRA LONG TIME FOR THE SSR BIT TO SET

```



```
1241 :      NEEDED AFTER AN INIT IS SENT TO TS11 TO WAIT FOR LOAD SEQ
1242 :      TO COMPLETE.  REPORT ERROR AND DROP UNIT IF EXTRA
1243 :      TIME IS EXCEEDED.
1244
1245      WAITSR::BEGIN COUNTER1
1246      INCR TIME FROM #1 TO #13 BY #1
1247      014616 012737 000001 002426
1248      014616 000402
1249      014624 000402
1250      014626 005237 002426
1251      014626 005237 002426
1252      014632 023727 002426 000013
1253      014632 023727 002426 000013
1254      014640 003023
1255      014642 032777 000200 165354
1256      014642 032777 000200 165354
1257      014650 001401
1258      014652 000416
1259      014652 000416
1260      014654
1261      014654
1262      014654 012727 001750
1263      014660 000000
1264      014662 013727 002116
1265      014666 000000
1266      014670 005367 177772
1267      014674 001375
1268      014676 005367 177756
1269      014702 001367
1270      014704
1271      014704 104422
1272      014706
1273      014706
1274      014706
1275      014706 000747
1276      014710
1277      014710
1278      014710
1279      014710
1280      014710 023727 002426 000013
1281      014716 003511
1282      014720
1283      014720 012746 015144
1284      014724 012746 000001
1285      014730 010600
1286      014732 104417
1287      014734 062706 000004
1288      014740
1289      014740
1290      014740 012737 000001 002426
1291      014746 000402
1292      014750
1293      014750 005237 002426
1294      014754
1295      014754 023727 002426 000160
1296      014762 003023
```

IF #TS.SSR SETIN @TSSR THEN
LEAVE COUNTER1
ELSE
DELAY 1000.
BREAK
ENDIF
ENDINC
END COUNTER1
IF TIME GT #13 THEN
PRINTF #WAITMR
BEGIN COUNTER2
INCR TIME FROM #1 TO #160 BY #1

50006\$: MOV #1,TIME
BR 50005\$
50006\$: INC TIME
50005\$: CMP TIME,#13
BGT 50007\$
BIT #TS.SSR,@TSSR
BEQ 50010\$
BR 50004\$
50010\$: MOV #1000.,(PC)+
.WORD 0
MOV LSDLY,(PC)+
.WORD 0
DEC -6(PC)
BNE .-4
DEC -22(PC)
BNE .-20
TRAP C\$BRK
50011\$: BR 50006\$
50007\$: BR 50006\$
50004\$: CMP TIME,#13
BLE 50012\$
MOV #WAITMR,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTF
ADD #4,SP
MOV #1,TIME
BR 50014\$
50015\$: INC TIME
50014\$: CMP TIME,#160
BGT 50016\$

1297	014764				IF #TS.SSR SETIN @TSSR THEN		
1298	014764	032777	000200	165232		BIT	#TS.SSR,@TSSR
1299	014772	001401				BEQ	50017\$
1300	014774				LEAVE COUNTER2		
1301	014774	000416				BR	50013\$
1302	014776				ELSE		
1303	014776				DELAY 1000.	50017\$:	
1304	014776						
1305	014776	012727	001750			MOV	#1000.,(PC)+
1306	015002	000000				.WORD	0
1307	015004	013727	002116			MOV	L\$DLY,(PC)+
1308	015010	000000				.WORD	0
1309	015012	005367	177772			DEC	-6(PC)
1310	015016	001375				BNE	.-4
1311	015020	005367	177756			DEC	-22(PC)
1312	015024	001367				BNE	.-20
1313	015026				BREAK		
1314	015026	104422				TRAP	C\$BRK
1315	015030				ENDIF		
1316	015030				ENDINC	50020\$:	
1317	015030						
1318	015030	000747				BR	50015\$
1319	015032					50016\$:	
1320	015032				END COUNTER2	50013\$:	
1321	015032				IF TIME GT #160 THEN		
1322	015032						
1323	015032	023727	002426	000160		CMP	TIME,#160
1324	015040	003430				BLE	50021\$
1325	015042				LET TEMP2 := @TSSR		
1326	015042	017737	165156	002434		MOV	@TSSR,TEMP2
1327	015050				LET R1 := #TERMTB		
1328	015050	012701	014532			MOV	#TERMTB,R1
1329	015054				ERRDF 2,SSROFF,LONMSG		
1330	015054	104455				TRAP	C\$ERDF
1331	015056	000002				.WORD	2
1332	015060	002566				.WORD	SSROFF
1333	015062	005774				.WORD	LONMSG
1334	015064				PRINTX #TERMA,TEMP2,UNIT		
1335	015064	013746	002422			MOV	UNIT,-(SP)
1336	015070	013746	002434			MOV	TEMP2,-(SP)
1337	015074	012746	010142			MOV	#TERMA,-(SP)
1338	015100	012746	000003			MOV	#3,-(SP)
1339	015104	010600				MOV	SP,RO
1340	015106	104415				TRAP	C\$PNTX
1341	015110	062706	000010			ADD	#10,SP
1342	015114	004737	014570		JSR PC,DROPU		
1343	015120				ELSE		
1344	015120	000410				BR	50022\$
1345	015122					50021\$:	
1346	015122				PRINTF #CRMSG		
1347	015122	012746	005763			MOV	#CRMSG,-(SP)
1348	015126	012746	000001			MOV	#1,-(SP)
1349	015132	010600				MOV	SP,RO
1350	015134	104417				TRAP	C\$PNTF
1351	015136	062706	000004			ADD	#4,SP
1352	015142				ENDIF		

```
1353 015142  
1354 015142  
1355 015142  
1356 015142 000207  
1357  
1358 015144 047045 040445 040527 WAITMR: .ASCIZ /%N%AWAITING FOR RDY.../  
1359 015152 052111 047111 020107  
1360 015160 047506 020122 042122  
1361 015166 027131 027056 000  
1362 015174  
1363  
1364  
1365
```

50022\$:
50012\$:
.EVEN

```

1366
1367          :      SUBR TO WAIT FOR TAPE MOTION TO CEASE WHEN AN INIT TO THE
1368          :      ON-LINE TS11 INITIATED A LOAD SEQUENCE.
1369
1370 015174 000240      WAITMT:: NOP
1371 015176 000240      NOP
1372 015200 000240      NOP
1373 015202 000240      NOP
1374 015204          IF #MOT SETIN XSTATO THEN
1375 015204 032737 000200 002266
1376 015212 001501          PRINTF #WAITMS
1377 015214
1378 015214 012746 015420
1379 015220 012746 000001
1380 015224 010600
1381 015226 104417
1382 015230 062706 000004
1383 015234          BEGIN COUNTER
1384 015234          INCR TIME FROM #1 TO #160 BY #1
1385 015234 012737 000001 002426
1386 015242 000402
1387 015244          50026$:
1388 015244 005237 002426          INC      TIME
1389 015250          50025$:
1390 015250 023727 002426 000160          CMP      TIME,#160
1391 015256 003031          BGT      50027$
1392 015260
1393 015260 012737 100017 002230          MOV      #GES,CMDPKT
1394 015266          LET @TSDB := #CMDPKT
1395 015266 012777 002230 164720          MOV      #CMDPKT,@TSDB
1396 015274          DELAY 1000.
1397 015274 012727 001750          MOV      #1000.,(PC)+
1398 015300 000000          .WORD 0
1399 015302 013727 002116          MOV      L$DLY,(PC)+
1400 015306 000000          .WORD 0
1401 015310 005367 177772          DEC      -6(PC)
1402 015314 001375          BNE      .-4
1403 015316 005367 177756          DEC      -22(PC)
1404 015322 001367          BNE      .-20
1405 015324          BREAK
1406 015324 104422          TRAP    C$BRK
1407 015326          IF #MOT NOTSETIN XSTATO THEN
1408 015326 032737 000200 002266
1409 015334 001001
1410 015336          LEAVE COUNTER
1411 015336 000401          ENDIF
1412 015340          ENDINC
1413 015340          50030$:
1414 015340          BR      50024$
1415 015340 000741          50027$:
1416 015342          50024$:
1417 015342          END COUNTER
1418 015342          IF TIME GT #160 THEN
1419 015342
1420 015342 023727 002426 000160          CMP      TIME,#160
1421 015350 003412          BLE      50031$

```

```

1422 015352                                PRINTB #WAITEM
1423 015352 012746 015456                MOV     #WAITEM,-(SP)
1424 015356 012746 000001                MOV     #1,-(SP)
1425 015362 010600                        MOV     SP,R0
1426 015364 104414                        TRAP    C$PNTB
1427 015366 062706 000004                ADD     #4,SP
1428 015372                                DOCLN
1429 015372 104444                        TRAP    C$DCLN
1430 015374                                ELSE
1431 015374 000410                        BR      50032$
1432 015376                                50031$:
1433 015376                                PRINTF #CRMSG
1434 015376 012746 005763                MOV     #CRMSG,-(SP)
1435 015402 012746 000001                MOV     #1,-(SP)
1436 015406 010600                        MOV     SP,R0
1437 015410 104417                        TRAP    C$PNTF
1438 015412 062706 000004                ADD     #4,SP
1439 015416                                ENDIF
1440 015416                                50032$:
1441 015416                                ENDIF
1442 015416                                50023$:
1443 015416 000207                        RTS PC
1444
1445
1446 015420 040445 040527 052111  WAITMS: .ASCIZ  /%AWAITING FOR TAPE TO STOP.../
1447 015426 047111 020107 047506
1448 015434 020122 040524 042520
1449 015442 052040 020117 052123
1450 015450 050117 027056 000056
1451 015456 040445 040527 052111  WAITEM: .ASCII  /%AWAIT TOO LONG%/
1452 015464 052040 047517 046040
1453 015472 047117 022507 116
1454 015477 045 051501 044527  .ASCIZ  /%ASWITCH UNIT OFF-LINE, RESTART DIAG%/
1455 015504 041524 020110 047125
1456 015512 052111 047440 043106
1457 015520 046055 047111 026105
1458 015526 051040 051505 040524
1459 015534 052122 042040 040511
1460 015542 022507 000116
1461
1462                                .EVEN

```

```
1463 ;S/R TO GENERATE AN EVEN OR ODD PAR BIT. ENTER WITH R1 CONTAINING THE #
1464 ;FOR WHICH YOU WISH TO HAVE PAR GEN. EXIT WITH THE PARITY BIT IN THE
1465 ;LOW ORDER BIT OF TEMPO. CLOBBERREDS TEMPO
1466
1467 015546 012737 000001 002430 ODDPAR:: MOV #1,TEMPO ;INIT THIS REG FOR ADD PAR.
1468 015554 000137 015564 ;CONTINUE
1469 015560 005037 002430 EVNPAR: CLR TEMPO ;INIT THIS REG FOR EVEN PAR.
1470 015564 012737 000020 002434 PARGEN: MOV #20,TEMP2 ;INIT COUNTER
1471 015572 005337 002434 1$: DEC TEMP2 ;DONE ALL BITS?
1472 015576 100405 BMI 2$ ;BR IF SO
1473 015600 006001 ROR R1 ;ROTATE THE DATA.
1474 015602 103773 BCS 1$
1475 015604 005237 002430 INC TEMPO
1476 015610 000770 BR 1$ ;DO IT AGAIN
1477
1478 015612 006001 2$: ROR R1 ;ROTATE ONCE MORE TO GET ORIG CONTENTS
1479 015614 042737 177776 002430 BIC #177776,TEMPO ;CLR ALL BUT LO ORDER BIT IN TEMPO.
1480 ;LO ORDER BIT IS THE PAR BIT
1481 ;RETURNED TO THE CALLING ROUTINE.
1482 015622 000241 CLC ;CLR CARRY BIT
1483 015624 000207 RTS PC ;RETURN
1484
1485
1486
1487
1488 ;S/R TO LOAD THE DIABLK WITH R3 (DATA) BITS AND R4 (CONTROL) BITS WITH
1489 ;CORRECT PARITY.
1490 ;
1491 ; INPUTS: R3 DATA, R4 CONTROL, R5 DIABLK INDEX.
1492 ;
1493 ; OUTPUTS: R3 DATA (UNCHANGED), R4 CONTROL (WITH GOOD PARITY), R5
1494 ; UPDATED DIABLK INDEX.
1495 ;
1496 ; CALLS: ODDPAR SUBROUTINE
1497
1495 015626 110345 TALOAD:: MOVB R3,-(R5) ;LOAD DATA
1496 015630 005703 TST R3 ;ALL 0'S DATA?
1497 015632 001403 BEQ 1$ ;BR IF SO
1498 015634 052704 000020 BIS #1S.DAP,R4 ;IF NOT, SET THE PARITY DATA ALSO IN R4 (CNTRL W
1499 015640 000402 BR 3$ ;CONTINUE.
1500
1501 015642 042704 000020 1$: BIC #1S.DAP,R4 ;CLR THE PAR DATA BIT IN CNTRL WORD.
1502 015646 010401 3$: MOV R4,R1 ;SAVE IT.
1503 015650 004737 015546 JSR PC,ODDPAR ;CALC CONTROL WORD PARITY.
1504 015654 005737 002430 TST TEMPO ;IS PARITY BIT SET?
1505 015660 001402 BEQ 2$ ;BR IF NOT
1506 015662 052704 000200 BIS #1S.PAR,R4 ;IF SO, SET THE PAR BIT.
1507 015666 110445 2$: MOVB R4,-(R5) ;LOAD THE CONTROL WORD
1508 015670 000207 RTS PC ;RETURN.
```

```
1509 ;S/R TO COMPLETE FILLING THE DIABLK WITH ENTRIES WHOSE FORMAT THE TS04
1510 ;RECOGNIZES.  UNUSED DIABLK LOCATIONS TO CONTAIN A TS04 RTS COMMAND.
1511
1512 015672 020527 002317  FIXDIA:: CMP    R5,#DIABLK+1      ;ARE WE ON DIABLK ODD BOUNDARY?
1513 015676 001406          BEQ    1$              ;BR IF SO
1514 015700 100407          BMI    2$              ;ARE WE ON DIABLK EVEN BOUNDARY?
1515                                     ;BR IF SO-WE'RE DONE.
1516 015702 112745 000033          MOVB   #POPJLO,-(R5)    ;LOAD TS04
1517 015706 112745 000000          MOVB   #POPJHI,-(R5)    ;RTS COMMAND.
1518 015712 000767          BR     FIXDIA          ;REPEAT.
1519
1520
1521 015714 112745 177777  1$:   MOVB   #-1,-(R5)      ;MUST END ON AN EVEN BOUNDARY.
1522 015720 000207          2$:   RTS     PC          ;RETURN.
1523
1524
1525
1526 ;S/R TO SETUP THE TS04 COMMAND PACKET AND INITIATE EXECUTION OF THE DIA
1527 ;COMMAND.
1528 ;   CALLS:: WT4SSR SUBROUTINE.
1529
1530 015722 004737 014430  DIAEXE:: JSR   PC,WT4SSR      ;SEE IF SSR IS SET.  SHOULD BE
1531 015726 004737 015772          JSR   PC,INITM        ;INITIALIZE THE MESSAGE PACKET.
1532 015732 012777 002250 164254  MOV   #DIAPKT,@TSDB    ;INITIATE EXECUTION
1533 015740 004737 014430          JSR   PC,WT4SSR      ;WAIT FOR SSR TO SET
1534 015744 000207          RTS     PC
```

```
1535 ;S/R TO SETUP THE COMMAND PACKET TO DO AN SCH COMMAND AND ISSUE THE
1536 ;COMMAND.
1537
1538 ; CALLS:: WT4SSR SUBROUTINE.
1539
1540 015746 004737 014616 SCHEXE:: JSR PC, WAITSR ;WAIT FOR SSR BIT TO SET.
1541 015752 004737 015772 JSR PC, INITM ;INITIALIZE THE MESSAGE PACKET.
1542 015756 012777 002240 164230 MOV #SCHPKT,@TSDB ;INITIATE EXECUTION.
1543 015764 004737 014616 JSR PC, WAITSR ;WAIT FOR SSR.
1544 015770 000207 RTS PC ;RETURN.
1545
1546
1547
1548 ;S/R TO INITIALIZE THE MESSAGE PACKET WITH ALL ONES.
1549 ; REGISTERS USED: R1
1550
1551 015772 INITM: LET R1 := #0 ;CLEAR LOCATION COUNTER.
1552 015772 005001 ; CLR R1
1553 015774 WHILE R1 NE #MSGEXT DO ;WHILE THERE ARE MORE LOCATIONS:
1554 015774 ; 50033$:
1555 015774 020127 000016 CMP R1,#MSGEXT
1556 016000 001406 BEQ 50034$
1557 016002 LET MSGPKT(R1) := #-1 ;INIT ONE LOCATION IN MSG PACKET.
1558 016002 012761 177777 002260 MOV #-1,MSGPKT(R1)
1559 016010 LET R1 := R1 + #2 ;UPDATE COUNTER
1560 016010 062701 000002 ADD #2,R1
1561 016014 ENDDO ;END OF INIT LOOP.
1562 016014 000767 BR 50033$
1563 016016 ; 50034$:
1564 016016 000207 RTS PC ;RETURN.
```



```

1565 ;S/R TO SHIFT BYTE A 1 IN A FIELD OF 0'S. WHEN WE'RE THROUGH
1566 ;DOING THAT, SHIFT BYTE A 0 IN A FIELD OF 1'S.
1567
1568 : INPUTS: R4=DATA TO SHIFT.
1569 : OUTPUTS: R4=SHIFTED DATA.
1570
1571 016020 006204 SHWRAP:: ASR R4 ;SHIFT THE DATA.
1572 016022 105704 TSTB R4 ;SHIFTING A 1 OR A 0?
1573 016024 100404 BMI 1$ ;BR IF SHIFTING A 0.
1574 016026 001005 BNE 2$ ;DONE SHIFTING A 1? BR IF NOT.
1575 016030 012704 177577 MOV #177577,R4 ;IF SO, SETUP TO SHIFT 0 IN FIELD OF 1'S.
1576 016034 000402 BR 2$ ;CONTINUE.
1577
1578 016036 052704 000200 1$: BIS #200,R4 ;SET THE MSB.
1579 016042 000207 2$: RTS PC ;RETURN.
1580
1581
1582
1583 ;S/R TO CHECK FOR TRACK ACTIVE ERRORS.
1584 :
1585 :
1586 : OUTPUTS:ACTRAC CONTAINS THE ACTUAL TRACK ACTIVE DATA IN BITS 0 THRU 8
1587 :
1588 016044 113737 002267 002360 TKACER:: MOVB MSGPKT+C18$TA,ACTRAC ;GET ACTUAL STATE OF ACTIVE BITS FOR CHANS 1-8.
1589 016052 LET R2 :B= MSGPKT+PRCHST ;GET THE TRK ACT DATA FOR CHAN 9
1590 016052 113702 002274 MOVB MSGPKT+PRCHST,R2
1591 016056 032702 000004 BIT #CH9.TA,R2 ;IS IT SET?
1592 016062 001404 BEQ 1$ ;BR IF NOT
1593 016064 052737 000400 002360 BIS #BIT8,ACTRAC ;IF SO, SET THE APPROPRIATE BIT IN "ACTUAL" REG
1594 016072 000403 BR 3$ ;CONTINUE.
1595
1596 016074 042737 000400 002360 1$: BIC #BIT8,ACTRAC ;CLEAR THE BIT IN "ACTUAL" REG.
1597 016102 042737 177000 002360 3$: BIC #177000,ACTRAC ;CLR GARBAGE.
1598 016110 023737 002342 002360 CMP EXTRAC,ACTRAC ;EXPECTED & ACTUAL DATA IDENTICAL?
1599 016116 001402 BEQ 2$ ;BR IF SO.
1600 016120 105237 002436 INCB ERRFLG ;SET THE ERR FLAG.
1601 016124 000207 2$: RTS PC ;RETURN.
1602

```

```

1603 ;COMMON S/R USED BY THE PE DATA TESTS TO BUILD THE DIABLK
1604 ;IN CORE.
1605 :
1606 : INPUTS:
1607 : R2 = PREAMBLE DATA.
1608 : R3 = BYTE 1 DATA.
1609 :
1610 016126 004737 016154 DATBLD:: JSR PC,PEINIT ;GO SET THE WRAP TASK ADDR IN DIABLD
1611 ;AND INIT DIABLK INDEX.
1612 016132 112745 000012 MOVB #FC.DAT!FC.VCO,-(R5) ;SET DATA & VCO MODE IN FMT CNTRL.
1613 016136 010304 MOV R3,R4 ;SETUP SECOND BYTE OF DATA.
1614 016140 005104 COM R4 ;COMPLEMENT THE BYTE 2 DATA.
1615 016142 042704 177000 BIC #177000,R4 ;CLR GARBAGE BITS.
1616 016146 004737 016176 JSR PC,PEDATA ;LOAD THE PREAMBLE AND THE 2 DATA BYTES
1617 ;IN DIABLK
1618 016152 000207 RTS PC ;RETURN.
1619 :
1620 :
1621 :
1622 ;S/R TO LOAD THE DIABLK WITH THE TS04 WRAP TASK ADDRESS,
1623 ;AND SETUP THE READ CONTROL WORD (RDCTLO) FOR MAINTENANCE MODE.
1624 :
1625 ;IMPLICIT INPUTS: DIABLK, DIAEXT, WRPLO, WRPHI
1626 ;OUTPUTS: R5=DIABLK INDEX.
1627 :
1628 016154 012705 002336 PEINIT:: MOV #DIABLK+DIAEXT,R5 ;SAVE ADR OF LAST WORD IN DIABLK
1629 ;SO WE CAN FILL DIABLK IN REVERSE ORDER.
1630 016160 LET -(R5) :B= #WRPLO ;LOAD THE NEW WRAP TASK ADDR LO
1631 016160 112745 000005 MOVB #WRPHI,-(R5) ;LOAD WRAP TASK ADDR HI.
1632 016164 112745 000200 MOVB #RD.MAI,-(R5) ;SET MAINT MODE IN RDCTLO.
1633 016170 112745 000100 RTS PC
1634 016174 000207
1635 :
1636 :

```

```

1637 ;S/R TO LOAD THE DIABLK WITH THE ALL 1'S PREAMBLE CHARACTER AND 2 BYTES
1638 ;OF DATA AND CONTROL. IT ALSO LOADS THE EXPECTED UNDESKEWED DATA IN THE
1639 ;FORMATTER AND WRITE CONTROL REGISTERS.
1640
1641 :
1642 :       INPUTS: R2=PREAMBLE ALL 1'S CHARACTER.
1643 :               R3=1ST BYTE OF DATA
1644 :               R4=2ND BYTE OF DATA
1645 :       CALLS:  PELOAD SUBROUTINE
1646
1646 016176 010237 002432 PEDATA:: MOV R2,TEMP1 ;SETUP TO LOAD DIABLK PREAMBLE
1647 016202 004737 016334 JSR PC,PELOAD ;DO IT
1648 016206 010337 002432 MOV R3,TEMP1 ;SETUP TO LOAD DIABLK & 1ST DATA BYTE
1649 016212 004737 016334 JSR PC,PELOAD ;DO IT
1650 016216 010437 002432 MOV R4,TEMP1 ;SETUP TO LOAD DIABLK WITH 2ND DATA BYTE
1651 016222 004737 016334 JSR PC,PELOAD ;DO IT
1652
1653 ;LOAD THE EXPECTED RESULTS.
1654
1655 016226 013737 002416 002342 4$: MOV DTKIDN,EXTRAC ;IDENTIFY DEAD TRACKS.
1656 016234 005137 002342 COM EXTRAC ;ALL TRACKS NOT DEAD ARE ACTIVE.
1657 016240 042737 177000 002342 BIC #177000,EXTRAC ;CLR GARBAGE.
1658 016246 013737 002416 002346 MOV DTKIDN,EXTRDD ;TRACKS DEAD DEFINED IN DTKIDN REG.
1659 016254 012737 000777 002336 MOV #777,EXORDY ;OUTPUT READY HI ON ALL TRACKS.
1660 016262 010337 002344 MOV R3,EXDATA ;EXPECTED DATA INFO (NOT SO FOR SKEW TEST)
1661 016266 043737 002416 002344 BIC DTKIDN,EXDATA ;ZERO THE DATA CORRESPONDING TO THE DEAD TRACKS.
1662 016274 013737 002416 002340 MOV DTKIDN,EX1DTR ;EXPECTED ONES OR DEAD TRK INFO.
1663 016302 050337 002340 BIS R3,EX1DTR
1664 016306 005103 COM R3 ;COMPLEMENT DATA.
1665 016310 013737 002416 002350 MOV DTKIDN,EXODTR ;EXPECTED 0 OR DEAD TRK INFO.
1666 016316 050337 002350 BIS R3,EXODTR
1667 016322 005103 COM R3 ;RESTORE R3.
1668 016324 042737 177000 002350 BIC #177000,EXODTR ;CLR GARBAGE.
1669 016332 000207 RTS PC ;RETURN
1670

```

```

1671 ;S/R TO LOAD CONTENTS OF TEMP1 INTO THE DIABLK BY GENERATING THE TWO
1672 ;BIT CELLS OF DATA AND CONTROL. THE ROUTINE ALSO LOADS DATA TO SIMULATE
1673 ;DEAD TRACKS.
1674
1675
1676 :
1677 : INPUT: TEMP1=DATA TO LOAD.
1678 : R5=DIABLK INDEX.
1679 : DTKIDN=DEAD TRACK IDEN REG. 1=DEAD; 0=LIVE.
1680
1681 :CLOBBERS TEMPO,R1,TEMP1
1682
1683 : OUTPUT: UPDATED DIABLK DATA
1684 : UPDATED DIABLK INDEX
1685
1686 ;NOTE THAT A LOGICAL "1" IS SENT TO THE TS04 AS A 1 TO 0 TRANSITION.
1687 ;LIKewise, THE TS04 INTERPRETS A 0 TO 1 TRANSITION AS A LOGICAL 0.
1688 ;THEREFORE, A LOGICAL 1 HAS A 1 BIT CELL FOLLOWED BY A 0 BIT CELL, AND
1689 ;LOGICAL 0 CONSISTS OF A 0 BIT CELL FOLLOWED BY A 1 BIT CELL. THE FIRST
1690 ;PASS OF THIS SUBROUTINE LOADS THE FIRST BIT CELL, WHILE THE SECOND PASS
1691 ;LOADS BIT CELL 2.
1692 ;DEAD TRACK BIT CELLS ARE EITHER BOTH 0 OR BOTH 1. (NO TRANSITION.)
1693 016334 012701 000044 PELOAD:: MOV #IS.NRZ!IS.WRF,R1 ;INIT CONTROL WORD STORAGE REG
1694 016340 113745 002432 1$: MOVB TEMP1,-(R5) ;LOAD DATA WORD (BIT CELL)
1695 016344 000301 SWAB R1 ;LOAD THE
1696 016346 153701 002432 BISB TEMP1,R1 ;DATA WORD IN R1 ALSO TO
1697 016352 000301 SWAB R1 ;GENERATE PARITY.
1698 016354 032737 000400 002432 BIT #BIT8,TEMP1 ;WHAT ABOUT PARITY TRACK. SET?
1699 016362 001402 BEQ 2$ ;BR IF NOT
1700 016364 052701 000020 BIS #IS.DAP,R1 ;IF SO, SET THE BIT IN THE CNTRL WORD.
1701 016370 004737 015546 2$: JSR PC,ODDPAR ;CALCULATE ODD PARITY. RESULT IN TEMPO BIT 1
1702 016374 005737 002430 TST TEMPO ;PARITY BIT SET?
1703 016400 001402 BEQ 3$ ;BR IF NOT
1704 016402 052701 000200 BIS #IS.PAR,R1 ;IF SO SET PARITY BIT FOR BOTH THE
1705 ;DATA AND CONTROL WORD.
1706 016406 110145 3$: MOVB R1,-(R5) ;LOAD THE CNTRL WORD (BIT CELL)
1707 016410 005737 002432 TST TEMP1 ;DONE WITH PASS 2?
1708 016414 100432 BMI 4$ ;BR IF SO
1709 016416 005737 002416 TST DTKIDN ;LOOKING FOR ANY DEAD TRACKS?
1710 016422 001424 BEQ 5$ ;BR IF NOT.
1711
1712 ;XOR THE DATA WITH THE DEAD TRACKS SO NO TRANSITIONS OCCUR ON
1713 ;THE DEAD TRACK BIT CELLS.
1714 ;XOR = (A)(NOT B) + (NOT A)(B) = NOT(A + NOT B) + NOT(NOT A + B)
1715
1716 016424 013737 002432 002430 MOV TEMP1,TEMPO ;SAVE ARG A.
1717 016432 013701 002416 MOV DTKIDN,R1 ;SAVE ARG B.
1718 016436 005137 002430 COM TEMPO ;INVERT ARG A.
1719 016442 005101 COM R1 ;INVERT ARG B.
1720 016444 053701 002432 BIS TEMP1,R1 ;GENERATE (A) + (NOT B)
1721 016450 053737 002416 002430 BIS DTKIDN,TEMPO ;GENERATE (NOT A) + (B)
1722 016456 005137 002430 COM TEMPO ;INVERT
1723 016462 005101 COM R1 ;INVERT
1724 016464 053701 002430 BIS TEMPO,R1 ;OR THE TWO ARGUMENTS TO GIVE THE XOR.
1725 016470 010137 002432 MOV R1,TEMP1 ;TEMP1 = TEMP1 XOR DTKIDN
1726 016474 005137 002432 5$: COM TEMP1 ;SET UP TO LOAD BIT CELL 2.

```

GLOBAL AREAS
CZTSIB.P11

MACY11 30(1046)
19-AUG-79 20:26

19-AUG-79 20:27 PAGE 47

GLOBAL SUBROUTINES SECTION

L 6

SEQ 0076

1727 016500 000715
1728 016502 000207
1729

4\$: BR PELOAD :DO IT
RTS PC :RETURN

```

1730 ;S/R TO RETRIEVE THE TS04 FORMATTER AND WRITE CONTROL REGISTERS AND
1731 ;CHECK TO SEE IF THEY CONTAIN EXPECTED DATA.
1732
1733 INPUTS: EXPECTED DATA PREVIOUSLY SET UP IN PEDATA SUBROUTINE
1734 MSGPKT - TS04 MESSAGE PACKET.
1735
1736 OUTPUT: ERROR FLAG, ACTUAL DATA
1737
1738 ;CLOBBERS R5,R1
1739
1740 PEERCK:: INCR R1 FROM #0 TO #12 BY #2
1741
1742 CLR R1
1743 BR 50035$
1744
1745 50036$: ADD #2,R1
1746
1747 50035$: CMP R1,#12
1748 BGT 50037$
1749
1750 LET ACTTBL(R1) := #0 ;CLEAR 6 LOC OF ACTUAL WRAP REG TABLE
1751 CLR ACTTBL(R1)
1752 BR 50036$
1753
1754 50037$:
1755 MOV MSGPKT+C18$OR,ACORDY ;ACTUAL OUTPUT READY INFO
1756 MOV MSGPKT+C18$1D,AC1DTR ;ACTUAL 1'S OR DEAD TRACK INFO.
1757 MOV MSGPKT+C18$TA,ACTRAC ;ACTUAL TRACK ACTIVE INFO
1758 MOV MSGPKT+C18$DA,ACDATA ;ACTUAL DATA INFO
1759 MOV MSGPKT+C18$TD,ACTRDD ;ACTUAL TRACK DEAD INFO
1760 MOV MSGPKT+C18$OD,ACODTR ;ACTUAL 0 OR DEAD TRACK INFO.
1761 LET TEMP2 :B= MSGPKT+PRCHST ;GET THE TRK ACT DATA FOR CHAN 9
1762 MOV MSGPKT+PRCHST,TE
1763
1764 MOV #6,TEMP1 ;INIT COUNTER
1765 MOV #ACTTBL,R1 ;SET INDEX OF THE FORMATTER AND
1766 ;WRITE CONTROL REG., TABLE.
1767 1$: ASR TEMP2 ;SHIFT CHAN 9 DATA INTO CARRY BIT
1768 BCC 2$ ;BR IF DATA WAS CLR.
1769 BIS #BIT8,(R1)+ ;IF DATA WAS SET THEN SET THE CHAN 9
1770 ;BIT IN THE APPROPRIATE "ACTUAL" REG.
1771 BR 3$
1772 2$: BIC #BIT8,(R1)+ ;IF CLR, THEN CLR THE CHAN 9 BIT IN
1773 ;THE APPROPRIATE "ACTUAL" REG.
1774 3$: DEC TEMP1 ;DONE CHECKING ALL THE NEEDED BITS?
1775 BNE 1$ ;BR IF NOT
1776
1777 ;NOW COMPARE THE ACTUAL & EXPECTED DATA AND SET THE ERROR FLAG
1778 ;IF WE HAD AN INVALID COMPARISON.
1779 MOV #EXPTBL,R1 ;SAVE "ACTUAL" TABLE INDEX
1780 MOV #6,TEMP1 ;INIT COUNTER
1781 4$: LET R5 := (R1) XOR 16(R1)
1782 MOV (R1),R5
1783 MOV 16(R1),-(SP)
1784 BIC R5,(SP)
1785 BIC 16(R1),R5

```

1786	016670	052605							
1787	016672			IF R5 NE #0 THEN			BIS	(SP)+,R5	
1788	016672	005705					TST	R5	
1789	016674	001402					BEQ	50040\$	
1790	016676	005237	002436	INC ERRFLG					
1791	016702			ENDIF					
1792	016702					50040\$:			
1793	016702			LET 34(R1) := 34(R1) SET.BY R5					
1794	016702	050561	000034				BIS	R5,34(R1)	
1795	016706			LET R1 := R1 + #2					
1796	016706	062701	000002				ADD	#2,R1	
1797									
1798	016712	005337	002432	DEC	TEMP1				;DONE CHECKNING ALL DATA?
1799	016716	001356		BNE	4\$;BR IF NOT
1800	016720	000207		RTS	PC				

```

1801 ;S/R TO SETUP THE DIABLK IN SUCH A WAY THAT WILL ENABLE THE ROM
1802 ;LOOKUP TABLE LOCATION, SPECIFIED IN ROMLKI, TO BE EXAMINED.
1803
1804 ;10.0 ADDRESS BITS ARE NEEDED TO ADDRESS THE ROM FROM 0-1777. THE 9 LOW
1805 ;ORDER ROM ADDRESS BITS ARE GATED FROM THE TRACK ACTIVE DATA WHILE THE
1806 ;MSB OF THE ROM ADDRESS IS GATED FROM THE VERTICAL PARITY ERROR (VPE).
1807 ;
1808 ;THIS S/R LOADS THE DIABLK WITH AN ALL 1'S PREAMBLE; CAUSES A VPE TO SET
1809 ;OR CLEAR IN BYTE 1 DEPENDING ON THE STATE OF ROMLKI (BIT 9); AND CAUSES
1810 ;DEAD TRACKS IN BYTE 2 CORRESPONDING TO BITS 0-8 OF ROMLKI.
1811 ;
1812
1813 016722 005037 002416 ROMLOK:: CLR DTKIDN ;CLR DEAD TRACK REG.
1814 016726 012737 000777 002432 MOV #777,TEMP1 ;SET UP TO LOAD ALL 1'S PREAMBLE
1815 016734 004737 016334 JSR PC,PELOAD ;LOAD PREAMBLE & CONTROL.
1816 016740 032737 001000 002420 BIT #BIT9,ROMLKI ;MSB OF ROM ADDR SET?
1817 016746 001403 BEQ 1$ ;BR IF NOT
1818 016750 005037 002432 CLR TEMP1 ;CAUSE VPE=1.
1819 016754 000403 BR 2$ ;CONTINUE
1820
1821 016756 012737 000777 002432 1$: MOV #777,TEMP1 ;CAUSE VPE=0.
1822 016764 004737 016334 2$: JSR PC,PELOAD ;LOAD BYTE 1 + CNTRL.
1823 016770 005037 002432 CLR TEMP1 ;INIT TEMP1.
1824 016774 013737 002420 002416 MOV ROMLKI,DTKIDN ;LOAD ROM ADDR IN DEAD TRACK REG.
1825 017002 005137 002416 COM DTKIDN ;INVERT IT.
1826 017006 042737 177000 002416 BIC #177000,DTKIDN ;CLR GARBAGE. MSB ADDRESS
1827 ;WAS TAKEN CARE OF IN BYTE 1.
1828 017014 004737 016334 JSR PC,PELOAD ;LOAD BYTE 2 DATA AND CONTROL.
1829 017020 000207 RTS PC ;RETURN.

```


1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884

:S/R TO GENERATE THE EXPECTED ROM LOOKUP TABLE OUTPUT FROM THE ADDRESS
:IN ROMLKI.

INPUTS: ROMLKI = ROM ADDRESS
OUTPUTS: EXPOML = EXPECTED ROM LOOKUP TABLE OUTPUT.
CLOBBERS: TEMPO, R1, R2

:THE ROM CONTENTS ARE SHOWN IN THE FOLLOWING TABLE:

OUTPUT	ADDRESSES WHICH ASSERT*	COMMENTS
	AAAAAAAAAA 9876543210	

MULT (MSB)	X-----	X = DON'T CARE. MULT = 1 IF ANY 2 ADDRESS LINES (OTHER THAN A9) ARE 1.
RDTMK	xxx0100x11 x110x001xx	X = DON'T CARE.
PREAM	XAACBCCABB	X = DON'T CARE. MUST HAVE AT LEAST ONE OF A & B & C. MUST HAVE ALL OF A OR B OR C.
9 OF 9	x111111111	X = DON'T CARE.
0 OF 9	x000000000	X = DON'T CARE.
COR DATA	1000000001 1000000010 1000000100 1000001000 1000010000 1000100000 1001000000 1010000000	
INCOR DATA	1000000000	
8 OF 9 (LSB)	x111111111 x111111110 x111111101 x111111011 x111110111 x111101111 x111011111 x110111111 x101111111 x011111111	X = DON'T CARE.

* = A0-A7 = MUX 2(0)-2(7)
A8 = MUX PARITY
A9 = LATCHED PARITY ERROR

```
1885 017022 005037 002352      ROMEX:: CLR      EXROML      ;INIT. EXPECTED ROM REG.
1886
1887                          ;CHECK FOR 8 OF 9.
1888
1889 017026 012702 000001      X8OF9: MOV      #1,R2      ;INIT 0'S CNTR.
1890 017032 012737 000011 002430  MOV      #11,TEMPO      ;LOAD BIT CNTR. NOT INTERESTED IN BIT10.
1891 017040 013701 002420      MOV      ROMLKI,R1      ;SAVE ROM TABLE ADDRESS
1892 017044 000241      1$:  CLC          ;CLR CARRY TO PREPARE FOR SHIFT
1893 017046 006201      ASR      R1          ;SHIFT IT. WAS IS A 1?
1894 017050 103401      BCS      2$          ;BR IF SO
1895 017052 005302      DEC      R2          ;DECR 0 CNTR.
1896 017054 005337 002430      2$:  DEC      TEMPO      ;DECR BIT CNTR DONE?
1897 017060 001371      BNE      1$          ;BR IF NOT
1898 017062 005702      TST      R2          ;8 OF 9?
1899 017064 100403      BMI      XCORDA      ;BR IF NOT
1900 017066 052737 000001 002352  BIS      #.8OF9,EXROML ;IF SO, SET THE BIT
1901
1902                          ;CHECK FOR CORRECTABLE DATA
1903
1904 017074 032737 001000 002420  XCORDA: BIT      #A9,ROMLKI ;COULD THIS BE CORRECTABLE DATA
1905 017102 001427      BEQ      XINCOR      ;BR IF NOT
1906 017104 032737 000400 002420  BIT      #A8,ROMLKI ;COULD THIS BE CORRECTABLE DATA?
1907 017112 001023      BNE      XINCOR      ;BR IF NOT.
1908 017114 012702 000002      MOV      #2,R2      ;INIT 1'S CNTR.
1909 017120 012737 000012 002430  MOV      #12,TEMPO    ;LOAD BIT CNTR. NOT INTERESTED IN MSB.
1910 017126 013701 002420      MOV      ROMLKI,R1    ;SAVE ROM TABLE ADDR.
1911 017132 000241      1$:  CLC          ;INIT CARRY TO PREPARE FOR SHIFT.
1912 017134 006201      ASR      R1          ;SHIFT IT. WAS IT A 1?
1913 017136 103001      BCC      2$          ;BR IF NOT
1914 017140 005302      DEC      R2          ;DECR. 1'S CNTR.
1915 017142 005337 002430      2$:  DEC      TEMPO      ;DECR BIT CNTR. DONE?
1916 017146 001371      BNE      1$          ;BR IF NOT
1917 017150 005702      TST      R2          ;CORRECTABLE DATA?
1918 017152 001003      BNE      XINCOR      ;BR IF NOT
1919 017154 052737 000004 002352  BIS      #.CORD,EXROML ;IF SO, SET THE BIT.
1920
1921                          ;CHECK FOR INCORRECTABLE DATA
1922
1923 017162 022737 001000 002420  XINCOR: CMP      #1000,ROMLKI ;INCORRECTABLE DATA?
1924 017170 001003      BNE      X0OF9      ;BR IF NOT
1925 017172 052737 000002 002352  BIS      #.INCOR,EXROML ;IF SO, SET THE "INCOR" BIT.
1926
1927                          ;CHECK FOR 0 OF 9
1928
1929 017200 032737 000777 002420  X0OF9: BIT      #777,ROMLKI ;0 OF 9?
1930 017206 001003      BNE      X9OF9      ;BR IF NOT
1931 017210 052737 000010 002352  BIS      #.0OF9,EXROML ;IF SO, SET THE BIT.
1932
1933                          ;CHECK FOR 9 OF 9
1934
1935 017216 022737 000777 002420  X9OF9: CMP      #777,ROMLKI ;9 OF 9?
1936 017224 001404      BEQ      1$          ;BR IF SO
1937 017226 022737 001777 002420  CMP      #1777,ROMLKI ;9 OF 9?
1938 017234 001003      BNE      XMULT      ;BR IF NOT
1939 017236 052737 000020 002352  1$:  BIS      #.9OF9,EXROML ;SET THE BIT.
1940
```

```

1941                                     ;CHECK FOR MULTIPLE TRACKS
1942
1943 017244 012702 000001                XMULT: MOV #1,R2                ;INIT 1'S CNTR.
1944 017250 012737 000011 002430        MOV #11,TEMPO                ;LOAD BIT CNTR.
1945 017256 013701 002420                MOV ROMLKI,R1                ;SAVE ROM TABLE ADDR
1946 017262 000241                        1$: CLC                        ;INIT CARRY TO PREPARE FOR SHIFT.
1947 017264 006201                        ASR R1                        ;SHIFT IT WAS IT A 1?
1948 017266 103001                        BCC 2$                        ;BR IF NOT
1949 017270 005302                        DEC R2                        ;DECR 1'S CNTR.
1950 017272 005337 002430                2$: DEC TEMPO                ;DECR BIT CNTR. DONE?
1951 017276 001371                        BNE 1$                        ;BR IF NOT
1952 017300 005702                        TST R2                        ;MULTIPLE TRACKS?
1953 017302 100003                        BPL XPREAM                    ;BR IF NOT
1954 017304 052737 000200 002352        BIS #.MULT,EXROML            ;IF SO, SET BIT BIT
1955
1956                                     ;CHECK FOR PREAMBLE
1957                                     ;ADDR LINES 9876543210
1958                                     ; XAACBCCABB
1959                                     ;PREAM=1 IF HAVE ONE OF A&B&C AND
1960                                     ; ALL OF A OR B OR C.                X = 0 OR 1.
1961
1962 017312 005002                        XPREAM: CLR R2
1963 017314 013701 002420                MOV ROMLKI,R1                ;SAVE ROM TABLE ADDRESS.
1964 017320 010137 002430                MOV R1,TEMPO                ;GENERATE THE
1965 017324 005137 002430                COM TEMPO                    ;COMPLIMENT OF THE ADDR.
1966 017330 032737 000604 002430        BIT #A8!A7!A2,TEMPO         ;ALL OF GROUP A?
1967 017336 001003                        BNE 1$                        ;BR IF NOT
1968 017340 052702 000011                BIS #BIT0!BIT3,R2           ;SET "ALL OF A" AND "1 OF A" INDICATION
1969 017344 000405                        BR 2$
1970
1971 017346 032701 000604                1$: BIT #A8!A7!A2,R1         ;1 OF GROUP A?
1972 017352 001440                        BEQ 7$                        ;BR IF NOT - NOT A PREAM
1973 017354 052702 000001                BIS #BIT0,R2                ;SET "1 OF A" INDICATION.
1974 017360 032737 000043 002430        2$: BIT #A5!A1!A0,TEMPO     ;ALL OF GROUP B ACTIVE?
1975 017366 001003                        BNE 3$                        ;BR IF NOT
1976 017370 052702 000012                BIS #BIT1!BIT3,R2           ;SET "ALL OF B" + "1 OF B".
1977 017374 000405                        BR 4$                        ;CONTINUE
1978 017376 032701 000043                3$: BIT #A5!A1!A0,R1         ;"1 OF B" ACTIVE.
1979 017402 001424                        BEQ 7$                        ;BR IF NOT - NOT A PREAM.
1980 017404 052702 000002                BIS #BIT1,R2                ;SET "1 OF B" INDICATION.
1981 017410 032737 000130 002430        4$: BIT #A6!A4!A3,TEMPO     ;ALL OF GROUP C ACTIVE?
1982 017416 001003                        BNE 5$                        ;BR IF NOT
1983 017420 052702 000014                BIS #BIT2!BIT3,R2           ;SET ALL OF C AND 1 OF C
1984 017424 000405                        BR 6$                        ;INDICATION.
1985 017426 032701 000130                5$: BIT #A6!A4!A3,R1         ;1 OF C?
1986 017432 001410                        BEQ 7$                        ;BR IF NOT-NO PREAMBLE
1987 017434 052702 000004                BIS #BIT2,R2                ;SET 1 OF C INDICATION.
1988
1989                                     ;IN ORDER FOR THE PREAMBLE CONDITION TO BE TRUE, R2 MUST CONTAIN
1990                                     ;17 AT THIS POINT.
1991
1992 017440 022702 000017                6$: CMP #17,R2                ;PREAMBLE?
1993 017444 001003                        BNE 7$                        ;BR IF NOT
1994 017446 052737 000040 002352        BIS #.PREAM,EXROML          ;IF SO, SET THE PREAMBLE BIT IN
1995                                     ;EXPECTED LOOKUP TABLE REG.
1996 017454 000240                7$: NOP

```

```

1997
1998
1999
2000
2001
2002
2003
2004
2005 017456 013701 002420          XRDFMK: MOV    ROMLK1,R1      ;SAVE ROM TABLE ADDR.
2006 017462 010137 002430          MOV    R1,TEMPO      ;GENERATE THE
2007 017466 005137 002430          COM    TEMPO         ;COMPLIMENT OF THE ADDR.
2008 017472 032701 000130          BIT    #A6:A4:A3,R1  ;CHECK THE 0'S IN COND 1 AND COND 2.
2009 017476 001013                 BNE    3$            ;BR IF COND 1 AND COND 2 NOT TRUE
2010 017500 032737 000043 002430   BIT    #A5:A1:A0,TEMPO ;CHECK 1'S IN COND 1.
2011 017506 001404                 BEQ    2$            ;BR IF COND 1 IS TRUE
2012 017510 032737 000604 002430   BIT    #A8:A7:A2,TEMPO ;CHECK 1'S IN CONDITION 2
2013 017516 001003                 BNE    3$            ;BR IF COND 2 IS FALSE
2014 017520 052737 000100 002352 2$: BIS    #.RDFMK,EXROML ;WE ARE READING A FILE MARK.
2015                                     ;SET APPROPRIATE BIT.
2016 017526 000207          3$:   RTS    PC      ;RETURN

```

```

2017 ;S/R TO RETRIEVE THE ACTUAL ROM LOOKUP TABLE OUTPUT AND COMPARE IT
2018 ;WITH THE EXPECTED OUTPUT.
2019 :
2020 : INPUT: EXROML - EXPECTED ROM OUTPUT.
2021 : MSGPKT - TS04 MESSAGE PACKET.
2022 :
2023 : OUTPUT: ERRFLG - ERROR FLAG
2024 : ACROML - ACTUAL ROM OUTPUT.
2025 :
2026 :
2027 :
2028 017530 ROMCK:: LET ACROML :B= MSGPKT+ROM$LK ;GET THE ACTUAL ROM OUTPUT
2029 017530 113737 002273 002370 ;MOV B,MSGPKT+ROM$LK
2030 017536 123737 002370 002352 ;MOV B,ACROML
2031 017544 001402 ;CMPB ACROML,EXROML ;ARE THEY THE SAME.
2032 017546 005237 002436 ;BEQ 1$ ;BR IF SO
2033 017552 000207 ;INC ERRFLG ;IF NOT SET THE ERROR FLAG
1$: RTS PC

```

2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089

017554 005037 002432
017560
017560 010102
017562 010246
017564 042716 000007
017570 042602
017572
017572 010103
017574 010346
017576 042716 000070
017602 042603
017604
017604 010104
017606 010446
017610 042716 000700
017614 042604
017616
017616 005005
017620 004737 020372
017624
017624 005705
017626 001011
017630
017630 012746 005062
017634 012746 000001
017640 010600
017642 104414
017644 062706 000004
017650
017650 000561
017652

: SUBROUTINES MSORT 1 AND MSORT2 SELECT BAD BOARDS FOR TESTS 4 - 7 PER
: TABLE BELOW WHERE:

22 = M8922 57 = G157
24 = M8924 23 = M8923

BAD BOARD SELECTION TABLE			
R5	BD/SLOT/BITS	MASK	PRINTB MSG
0	22/7/0-8		FMTCTR
1	24/6/0,1,2	R2=7	FMTCH6
2	24/5/3,4,5	R3=70	FMTCH5
3	24/4/6,7,8	R4=700	FMTCH4
400	57/2/0,1,5,6	R2=143	RDCH2
1000	57/1/2,3,4,7	R3=234	RDCH1
1400	23/3/8	R4=400	RDCHP3

: TEST 4 -- MODULE SORT

```

MSORT1:: CLR    TEMP1                ;TEPM1=BOARDS 24 SELECTION FLAG
          LET R2 := R1 AND #7         ;R2=BOARD 24/6 MASK
                                          MOV    R1,R2
                                          MOV    R2,-(SP)
                                          BIC    #7,(SP)
                                          BIC    (SP)+,R2
          LET R3 := R1 AND #70        ;R3=BOARD 24/5 MASK
                                          MOV    R1,R3
                                          MOV    R3,-(SP)
                                          BIC    #70,(SP)
                                          BIC    (SP)+,R3
          LET R4 := R1 AND #700       ;R4=BOARD 24/4 MASK
                                          MOV    R1,R4
                                          MOV    R4,-(SP)
                                          BIC    #700,(SP)
                                          BIC    (SP)+,R4
          LET R5 := #0                ;R5=BOARD 22/7 FLAG
          JSR    PC,PAIRST             ;SORT 22 AND 24 BOARDS, RESULT IN R5 LO BYTE
          IF R5 EQ #0 THEN
          PRINTB #FMTCTR              ;22 BOARD SELECTED, CALL BAD 22/7
                                          MOV    #FMTCTR,-(SP)
                                          MOV    #1,-(SP)
                                          MOV    SP,R0
                                          TRAP  C$PNTB
                                          ADD    #4,SP
          ELSE                          ;24 BOARD SELECTED, CONTINUE SORTING
          BR     50041$
          BR     50042$
          BR     50041$:

```

2090	017652		LET R2 := R1 AND #143	;R2=BOARD 57/2 MASK	
2091	017652	010102			MOV R1,R2
2092	017654	010246			MOV R2,-(SP)
2093	017656	042716	000143		BIC #143,(SP)
2094	017662	042602			BIC (SP)+,R2
2095	017664		LET R3 := R1 AND #234	;R3=BOARD 57/1 MASK	
2096	017664	010103			MOV R1,R3
2097	017666	010346			MOV R3,-(SP)
2098	017670	042716	000234		BIC #234,(SP)
2099	017674	042603			BIC (SP)+,R3
2100	017676		LET R4 := R1 AND #400	;R4=BOARD 23/3 MASK	
2101	017676	010104			MOV R1,R4
2102	017700	010446			MOV R4,-(SP)
2103	017702	042716	000400		BIC #400,(SP)
2104	017706	042604			BIC (SP)+,R4
2105	017710		LET R5 := SWAP R5		
2106	017710	000305			SWAB R5
2107	017712	004737	JSR PC,PAIRST	;SORT 57 AND 23 BOARDS,	RESULT IN R5 HI BYTE
2108	017716		LET R5 := SWAP R5	;ANALYZE R5 LO BYTE	
2109	017716	000305			SWAB R5
2110	017720		IF R5 EQ #1 THEN		
2111	017720	020527	000001		CMP R5,#1
2112	017724	001011			BNE 50043\$
2113	017726		PRINTB #FMTCH6	;CALL BAD 24/6	
2114	017726	012746	005174		MOV #FMTCH6,-(SP)
2115	017732	012746	000001		MOV #1,-(SP)
2116	017736	010600			MOV SP,R0
2117	017740	104414			TRAP C\$PNTB
2118	017742	062706	000004		ADD #4,SP
2119	017746		ELSE		
2120	017746	000432			BR 50044\$
2121	017750				50043\$:
2122	017750		IF R5 EQ #2 THEN		
2123	017750	020527	000002		CMP R5,#2
2124	017754	001011			BNE 50045\$
2125	017756		PRINTB #FMTCH5	;CALL BAD 24/5	
2126	017756	012746	005251		MOV #FMTCH5,-(SP)
2127	017762	012746	000001		MOV #1,-(SP)
2128	017766	010600			MOV SP,R0
2129	017770	104414			TRAP C\$PNTB
2130	017772	062706	000004		ADD #4,SP
2131	017776		ELSE		
2132	017776	000416			BR 50046\$
2133	020000				50045\$:
2134	020000		IF R5 EQ #3 THEN		
2135	020000	020527	000003		CMP R5,#3
2136	020004	001011			BNE 50047\$
2137	020006		PRINTB #FMTCH4	;CALL BAD 24/4	
2138	020006	012746	005326		MOV #FMTCH4,-(SP)
2139	020012	012746	000001		MOV #1,-(SP)
2140	020016	010600			MOV SP,R0
2141	020020	104414			TRAP C\$PNTB
2142	020022	062706	000004		ADD #4,SP
2143	020026		ELSE		
2144	020026	000402			BR 50050\$
2145	020030				50047\$:

```
2146 020030 005237 002432          INC TEMP1          ;TEPM1 NOT=0 WHEN NO BAD 24 WERE CALLED
2147 020034          ENDIF
2148 020034          50050$:
2149 020034          ENDIF
2150 020034          50046$:
2151 020034          ENDIF
2152 020034          50044$:
2153 020034          IF TEMP1 NE #0 THEN          ;IF NO BAD 24 BOARDS CALLED, CONTINUE SORT
2154 020034 005737 002432          TST          TEMP1
2155 020040 001465          BEQ          50051$
2156 020042          LET R1 := ACTRK1 OR ACTRK3          ;ANALYZE TRACK INACTIVE SUBTSTS RESULTS
2157 020042 013701 002406          MOV          ACTRK1,R1
2158 020046 053701 002412          BIS          ACTRK3,R1
2159 020052          IF R1 NE #0 THEN          ;R1 NOT=0, TRACK INACTIVE FAILED
2160 020052 005701          TST          R1
2161 020054 001430          BEQ          50052$
2162 020056          LET R1 := R5          ;SAVE R5
2163 020056 010501          MOV          R5,R1
2164 020060          LET R5 := R5 AND #3          ;KEEP 2 LSB'S
2165 020060 010546          MOV          R5,-(SP)
2166 020062 042716 000003          BIC          #3,(SP)
2167 020066 042605          BIC          (SP)+,R5
2168 020070 004737 020440          JSR PC,FMTSEL          ;CALL APPROPRIATE BAD 24
2169 020074          PRINTB #FMTCTR          ;CALL BAD 22
2170 020074 012746 005062          MOV          #FMTCTR,-(SP)
2171 020100 012746 000001          MOV          #1,-(SP)
2172 020104 010600          MOV          SP,R0
2173 020106 104414          TRAP          C$PNTB
2174 020110 062706 000004          ADD          #4,SP
2175 020114          LET R5 := SWAP R1          ;FETCH R5 HI BYTE
2176 020114 010105          MOV          R1,R5
2177 020116 000305          SWAB          R5
2178 020120          LET R5 := R5 AND #3          ;KEEP 2 LSB'S
2179 020120 010546          MOV          R5,-(SP)
2180 020122 042716 000003          BIC          #3,(SP)
2181 020126 042605          BIC          (SP)+,R5
2182 020130 004737 020562          JSR PC,RDCSEL          ;CALL APPROPRIATE BAD 57/23
2183 020134          ELSE          ;R1=0,TRACK INACTIVE DID NOT FAIL
2184 020134 000427          BR          50053$
2185 020136          50052$:
2186 020136          LET R1 := R5          ;SAVE R5
2187 020136 010501          MOV          R5,R1
2188 020140          LET R5 := SWAP R5
2189 020140 000305          SWAB          R5
2190 020142          LET R5 := R5 AND #3          ;KEEP 2 LSB'S OF HI BYTE
2191 020142 010546          MOV          R5,-(SP)
2192 020144 042716 000003          BIC          #3,(SP)
2193 020150 042605          BIC          (SP)+,R5
2194 020152 004737 020562          JSR PC,RDCSEL          ;CALL APPROPRIATE BAD 57/23
2195 020156          LET R5 := R1 AND #3          ;FETCH 2 LSB'S OF R5 LO BYTE
2196 020156 010105          MOV          R1,R5
2197 020160 010546          MOV          R5,-(SP)
2198 020162 042716 000003          BIC          #3,(SP)
2199 020166 042605          BIC          (SP)+,R5
2200 020170 004737 020440          JSR PC,FMTSEL          ;CALL APPROPRIATE BAD 24
2201 020174          PRINTB #FMTCTR          ;CALL BAD 22
```


2202 020174 012746 005062
2203 020200 012746 000001
2204 020204 010600
2205 020206 104414
2206 020210 062706 000004
2207 020214
2208 020214
2209 020214
2210 020214
2211 020214
2212 020214
2213 020214 000207

ENDIF
ENDIF
ENDIF
RTS PC

50053\$:
50051\$:
50042\$:

MOV #FMTCTR, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP C\$PNTB
ADD #4, SP

```
2214
2215 ; TEST 5 - 7 -- MODULE SORT
2216
2217 020216 MSORT2:: IF R1 NE #0 THEN ;ERROR FROM 24 BOARDS, WHICH ONE?
2218 020216 005701 TST R1
2219 020220 001442 BEQ 50054$
2220 020222 LET R2 := R1 AND #7 ;R2=24/6 MASK
2221 020222 010102 MOV R1,R2
2222 020224 010246 MOV R2,-(SP)
2223 020226 042716 000007 BIC #7,(SP)
2224 020232 042602 BIC (SP)+,R2
2225 020234 LET R3 := R1 AND #70 ;R3=24/5 MASK
2226 020234 010103 MOV R1,R3
2227 020236 010346 MOV R3,-(SP)
2228 020240 042716 000070 BIC #70,(SP)
2229 020244 042603 BIC (SP)+,R3
2230 020246 LET R5 := R1 AND #700 ;R4=24/4 MASK
2231 020246 010105 MOV R1,R5
2232 020250 010546 MOV R5,-(SP)
2233 020252 042716 000700 BIC #700,(SP)
2234 020256 042605 BIC (SP)+,R5
2235 020260 LET R5 := #0 ;R5=22/7 FLAG
2236 020260 005005 CLR R5
2237 020262 004737 020372 JSR PC,PAIRST ;SORT 22 AND 24 BOARDS
2238 020266 IF R5 EQ #0 THEN ;
2239 020266 005705 TST R5
2240 020270 001011 BNE 50055$
2241 020272 PRINTB #FMTCTR ;CALL 22/7
2242 020272 012746 005062 MOV #FMTCTR,-(SP)
2243 020276 012746 000001 MOV #1,-(SP)
2244 020302 010600 MOV SP,R0
2245 020304 104414 TRAP C$PNTB
2246 020306 062706 000004 ADD #4,SP
2247 020312 ELSE ;24 SELECTED, WHICH ONE?
2248 020312 000404 BR 50056$
2249 020314 50055$:
2250 020314 004737 020440 JSR PC,FMTSEL ;CALL APPROPRIATE 24
2251 020320 004737 020334 JSR PC,DTRCHK ;CHECK OR FUNCTION OF 22
2252 020324 ENDIF
2253 020324 50056$:
2254 020324 ELSE
2255 020324 000402 BR 50057$
2256 020326 50054$:
2257 020326 004737 020334 JSR PC,DTRCHK ;NO ERROR FROM 24, CHECK OR FUNCTION OF 22
2258 020332 ENDIF
2259 020332 50057$:
2260 020332 000207 RTS PC
2261
2262 ;CHECK DEAD TRACK ORING ON 22 BOARD
2263
2264 020334 DTRCHK: LET R1 := OR1DTR OR ORODTR ;GET OR DATA
2265 020334 013701 002374 MOV OR1DTR,R1
2266 020340 053701 002404 BIS ORODTR,R1
2267 020344 IF R1 NE #0 THEN
2268 020344 005701 TST R1
2269 020346 001410 BEQ 50060$
```

GLOBAL AREAS MACY11 30(1046) 19-AUG-79 20:27
CZTSIB.P11 19-AUG-79 20:26

PAGE 61
GLOBAL SUBROUTINES SECTION

SEQ 0090

2270	020350		
2271	020350	012746	005062
2272	020354	012746	000001
2273	020360	010600	
2274	020362	104414	
2275	020364	062706	000004
2276	020370		
2277	020370		
2278	020370	000207	

PRINTB #FMTCTR

;OR DATA BAD, CALL 22 BOARD

MOV	#FMTCTR,-(SP)
MOV	#1,-(SP)
MOV	SP,R0
TRAP	(SPNTB
ADD	#4,SP

ENDIF

50060\$:

RTS PC

```

2279 ;PAIR OF BOARDS SORTING SUBR
2280
2281 ;(R2)(R3) + (R2)(R4) + (R3)(R4) NEQ 0 ----> R5=0
2282
2283 ; R2 NEQ 0 ----> R5=1
2284 ; R3 NEQ 0 ----> R5=2
2285 ; R4 NEQ 0 ----> R5=3
2286
2287 020372 PAIRST: IF R2 NE #0 THEN
2288 020372 005702
2289 020374 001407
2290 020376 IF R3 EQ #0 AND R4 EQ #0 THEN
2291 020376 005703
2292 020400
2293 020402 005704
2294 020404 001002
2295 020406 LET R5 := R5 SET.BY #1
2296 020406 052705 000001
2297 020412 ENDIF
2298 020412
2299 020412 50062$:
2300 020412 000411 ELSE
2301 020414
2302 020414 IF R3 NE #0 THEN
2303 020414 005703
2304 020416 001405
2305 020420 IF R4 EQ #0 THEN
2306 020420 005704
2307 020422 001002
2308 020424 LET R5 := R5 SET.BY #2
2309 020424 052705 000002
2310 020430 ENDIF
2311 020430
2312 020430 ELSE
2313 020430 000402
2314 020432
2315 020432 LET R5 := R5 SET.BY #3
2316 020432 052705 000003
2317 020436 ENDIF
2318 020436
2319 020436 ENDIF
2320 020436 50066$:
2321 020436 000207 RTS PC
2322
2323 ;FORMATTERS 24 SELECT
2324
2325 020440 FMTSEL: SELECT R5 OF 3 VERIFY
2326 020440 010546
2327 020442 003403
2328 020444 021627 000003
2329 020450 003402
2330 020452 50067$:
2331 020452 012716 000004
2332 020456 50070$:
2333 020456 006316
2334 020460 060716

```

```

TST R2
BEQ 50061$
TST R3
BNE 50062$
TST R4
BNE 50062$
BIS #1,R5
50062$:
BR 50063$
50061$:
TST R3
BEQ 50064$
TST R4
BNE 50065$
BIS #2,R5
50065$:
BR 50066$
50064$:
BIS #3,R5
50066$:
50063$:
MOV R5, -(SP)
BLE 50067$
CMP (SP), #3
BLE 50070$
MOV #4, (SP)
50070$:
ASL (SP)
ADD PC, (SP)

```

2335 020462 063607
 2336 020464
 2337 020464 000010
 2338 020466 000032
 2339 020470 000054
 2340 020472 000074
 2341 020474
 2342 020474
 2343 020474
 2344 020474 012746 005174
 2345 020500 012746 000001
 2346 020504 010600
 2347 020506 104414
 2348 020510 062706 000004
 2349 020514
 2350 020514 000421
 2351 020516
 2352 020516
 2353 020516 012746 005251
 2354 020522 012746 000001
 2355 020526 010600
 2356 020530 104414
 2357 020532 062706 000004
 2358 020536
 2359 020536 000410
 2360 020540
 2361 020540
 2362 020540 012746 005326
 2363 020544 012746 000001
 2364 020550 010600
 2365 020552 104414
 2366 020554 062706 000004
 2367 020560
 2368 020560
 2369 020560 000207
 2370
 2371
 2372
 2373 020562
 2374 020562 010546
 2375 020564 003403
 2376 020566 021627 000003
 2377 020572 003402
 2378 020574
 2379 020574 012716 000004
 2380 020600
 2381 020600 006316
 2382 020602 060716
 2383 020604 063607
 2384 020606
 2385 020606 000010
 2386 020610 000032
 2387 020612 000054
 2388 020614 000074
 2389 020616
 2390 020616

CASE 1

PRINTB #FMTCH6

CASE 2

PRINTB #FMTCH5

CASE 3

PRINTB #FMTCH4

ENDSELECT

RTS PC

;READ CHANNEL 57/23 SELECT

RDCSEL: SELECT R5 OF 3 VERIFY

CASE 1

50071\$: ADD @ (SP)+,PC
 .WORD 50075\$-50071\$
 .WORD 50074\$-50071\$
 .WORD 50073\$-50071\$
 .WORD 50072\$-50071\$
 50075\$:
 MOV #FMTCH6,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #4,SP
 BR 50072\$
 50074\$:
 MOV #FMTCH5,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #4,SP
 BR 50072\$
 50073\$:
 MOV #FMTCH4,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #4,SP
 50072\$:
 MOV R5,-(SP)
 BLE 50076\$
 CMP (SP),#3
 BLE 50077\$
 50076\$:
 MOV #4,(SP)
 50077\$:
 ASL (SP)
 ADD PC,(SP)
 ADD @ (SP)+,PC
 50100\$:
 .WORD 50104\$-50100\$
 .WORD 50103\$-50100\$
 .WORD 50102\$-50100\$
 .WORD 50101\$-50100\$
 50104\$:

2391 020616
2392 020616 012746 005403
2393 020622 012746 000001
2394 020626 010600
2395 020630 104414
2396 020632 062706 000004
2397 020636
2398 020636 000421
2399 020640
2400 020640
2401 020640 012746 005452
2402 020644 012746 000001
2403 020650 010600
2404 020652 104414
2405 020654 062706 000004
2406 020660
2407 020660 000410
2408 020662
2409 020662
2410 020662 012746 005521
2411 020666 012746 000001
2412 020672 010600
2413 020674 104414
2414 020676 062706 000004
2415 020702
2416 020702
2417 020702 000207

PRINTB #RDCH2

CASE 2

PRINTB #RDCH1

CASE 3

PRINTB #RDCHP3

ENDSELECT
RTS PC

MOV #RDCH2,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

BR 50101\$
50103\$:

MOV #RDCH1,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

BR 50101\$
50102\$:

MOV #RDCHP3,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

50101\$:

```
2418
2419 ; TEST 9, 10 -- MODULE SORT
2420
2421 020704 MSORT3:: LET R1 := @TSSR CLR.BY #TCFCMK ;GET TERMINATION AND FATAL CLASS CODES
2422 020704 017701 161314 MOV @TSSR,R1
2423 020710 042701 177701 BIC #TCFCMK,R1
2424 020714 IF R1 EQ #16 THEN ;FC=0, TC=7
2425 020714 020127 000016 CMP R1,#16
2426 020720 001075 BNE 50105$
2427 020722 ERRHRD 19,MICROE,MICERR TRAP C$ERHRD
2428 020722 104456 .WORD 19
2429 020724 000023 .WORD MICROE
2430 020726 003561 .WORD MICERR
2431 020730 014164
2432 020732 LET R1 := SWAP XSTAT3 ;INPUT MICRO CODE ERROR
2433 020732 013701 002274 MOV XSTAT3,R1
2434 020736 000301 SWAB R1
2435 020740 LET R1 := R1 CLR.BY #177400 ;KEEP 8 LSB'S
2436 020740 042701 177400 BIC #177400,R1
2437 020744 IF R1 EQ #337 THEN
2438 020744 020127 000337 CMP R1,#337
2439 020750 001011 BNE 50106$
2440 020752 PRINTB #COD337
2441 020752 012746 027256 MOV #COD337,-(SP)
2442 020756 012746 000001 MOV #1,-(SP)
2443 020762 010600 MOV SP,R0
2444 020764 104414 TRAP C$PNTB
2445 020766 062706 000004 ADD #4,SP
2446 020772 ELSE
2447 020772 000450 BR 50107$
2448 020774 50106$:
2449 020774 IF R1 LOS #127 THEN
2450 020774 020127 000127 CMP R1,#127
2451 021000 101034 BHI 50110$
2452 021002 LET R1 := R1 CLR.BY #177740
2453 021002 042701 177740 BIC #177740,R1
2454 021006 LET R1 := R1 SHIFT 1 ;R1X2 FOR WORD ADDRESSING
2455 021006 006301 ASL R1
2456 021010 LET R1 := R1 + #LOOKTB ;INDEX INTO LOOK-UP TBL
2457 021010 062701 021116 ADD #LOOKTB,R1
2458 021014 LET R2 := (R1) SHIFT 1 ;GET WORD OFFSET INTO CODTBL FROM LOOKTB
2459 021014 111102 MOVB (R1),R2
2460 021016 106302 ASLB R2
2461 021020 LET R2 := R2 + #CODTBL ;INDEX INTO CODTBL
2462 021020 062702 021176 ADD #CODTBL,R2
2463 021024 LET R3 := SWAP (R1) ;GET NO OF LINE TO PRINT
2464 021024 011103 MOV (R1),R3
2465 021026 000303 SWAB R3
2466 021030 LET R3 := R3 AND #377 ;FROM LOOKTB
2467 021030 010346 MOV R3,-(SP)
2468 021032 042716 000377 BIC #377,(SP)
2469 021036 042603 BIC (SP)+,R3
2470 021040 REPEAT
2471 021040 50111$:
2472 021040 PRINTB (R2) ;PRINT CURRENT LINE
2473 021040 011246 MOV (R2),-(SP)
```

2474	021042	012746	000001			MOV	#1,-(SP)
2475	021046	010600				MOV	SP,R0
2476	021050	104414				TRAP	C\$PNTB
2477	021052	062706	000004			ADD	#4,SP
2478	021056			LET R2 := R2 + #2	; THEN THE NEXT,		
2479	021056	062702	000002			ADD	#2,R2
2480	021062			LET R3 := R3 - #1	; IF ANY		
2481	021062	005303				DEC	R3
2482	021064			UNTIL R3 EQ #0			
2483	021064	005703				TST	R3
2484	021066	001364				BNE	50111\$
2485	021070			ELSE			
2486	021070	000411				BR	50112\$
2487	021072					50110\$:	
2488	021072			PRINTB #CODXXX,R1			
2489	021072	010146				MOV	R1,-(SP)
2490	021074	012746	027220			MOV	#CODXXX,-(SP)
2491	021100	012746	000002			MOV	#2,-(SP)
2492	021104	010600				MOV	SP,R0
2493	021106	104414				TRAP	C\$PNTB
2494	021110	062706	000006			ADD	#6,SP
2495	021114			ENDIF		50112\$:	
2496	021114						
2497	021114			ENDIF		50107\$:	
2498	021114						
2499	021114			ENDIF		50105\$:	
2500	021114						
2501	021114	000207		RTS PC			
2502							


```

2503
2504
2505
2506
2507 021116      000      003
2508 021120      003      002
2509 021122      005      001
2510 021124      006      002
2511 021126      010      002
2512 021130      012      001
2513 021132      013      001
2514 021134      014      001
2515 021136      015      003
2516 021140      020      003
2517 021142      023      002
2518 021144      025      003
2519 021146      030      004
2520 021150      034      004
2521 021152      040      004
2522 021154      044      003
2523 021156      047      002
2524 021160      051      002
2525 021162      053      001
2526 021164      054      001
2527 021166      055      002
2528 021170      057      003
2529 021172      062      003
2530 021174      065      003
2531
2532
2533
2534

```

```

;MICRO DIAG ERROCR CODE LOOKUP TABLE
;COD00 THRU COD27
;COD22, COD23, COD30 AND ABOVE, ARE UNDEFINED

```

```

LOOKTB: .BYTE 0,3
        .BYTE 3,2
        .BYTE 5,1
        .BYTE 6,2
        .BYTE 10,2
        .BYTE 12,1
        .BYTE 13,1
        .BYTE 14,1
        .BYTE 15,3
        .BYTE 20,3
        .BYTE 23,2
        .BYTE 25,3
        .BYTE 30,4
        .BYTE 34,4
        .BYTE 40,4
        .BYTE 44,3
        .BYTE 47,2
        .BYTE 51,2
        .BYTE 53,1
        .BYTE 54,1
        .BYTE 55,2
        .BYTE 57,3
        .BYTE 62,3
        .BYTE 65,3

```

```

:
:
:
:

```

```

!
! -----> NO OF LINES TO PRINT, IN HI BYTE (ODD)
! -----> OFFSET IN CODTBL, IN LO BYTE (EVEN)

```

```

2535 ;MICRO CODE ERROR AD TABLE
2536
2537 021176 021356 021464 021551 CODTBL: .WORD COD00,COD00A,COD00B ;0 3
2538 021204 021633 021732 .WORD COD01,COD01A ;3 2
2539 021210 021754 .WORD COD02 ;5 1
2540 021212 022047 022155 .WORD COD03,COD03A ;6 2
2541 021216 022233 022341 .WORD COD04,COD04A ;10 2
2542 021222 022370 .WORD COD05 ;12 1
2543 021224 022467 .WORD COD06 ;13 1
2544 021226 022564 .WORD COD07 ;14 1
2545 021230 022655 022763 023065 .WORD COD10,COD10A,COD10B ;15 3
2546 021236 023116 023201 023265 .WORD COD11,COD11A,COD11B ;20 3
2547 021244 023365 023470 .WORD COD12,COD12A ;23 2
2548 021250 023550 023647 023721 .WORD COD13,COD13A,COD13B ;25 3
2549 021256 024001 024103 024163 .WORD COD14,COD14A,COD14B,COD14C ;30 4
2550 021264 024236
2551 021266 024316 024417 024503 .WORD COD15,COD15A,COD15B,COD15C ;34 4
2552 021274 024557
2553 021276 024637 024741 025021 .WORD COD16,COD16A,COD16B,COD16C ;40 4
2554 021304 025103
2555 021306 025163 025267 025346 .WORD COD17,COD17A,COD17B ;44 3
2556 021314 025426 025534 .WORD COD20,COD20A ;47 2
2557 021320 025634 025742 .WORD COD21,COD21A ;51 2
2558 021324 025770 .WORD COD22 ;53 1
2559 021326 026025 .WORD COD23 ;54 1
2560 021330 026062 026170 .WORD COD24,COD24A ;55 2
2561 021334 026226 026334 026410 .WORD COD25,COD25A,COD25B ;57 3
2562 021342 026505 026570 026644 .WORD COD26,COD26A,COD26B ;62 3
2563 021350 026741 027047 027123 .WORD COD27,COD27A,COD27B ;65 3
2564 : ;
2565 : ;
2566 : ;
2567 : ;

```

:OFFSET INTO THIS TABLE <-----:
:NO OF .WORDS=NO OF LINES TO PRINT <-----:

2568
2569
2570

:MICRO CODE ERROR MSGS

```

.NLIST BEX
021356 040445 030061 020060 COD00: .ASCIZ /%A100 BASIC IO MICRO FAIL:PAR ERR, IOATN,HANDSHAKE, M8967(12) 14%N/
021464 040445 020040 020040 COD00A: .ASCIZ /%A DATA WINDOW TEST BETWEEN IO AND MAIN MICROS,%N/
021551 045 020101 020040 COD00B: .ASCIZ /%A SERIAL BUS.SHIN (SHIFT IN) STUCK TRUE,...%N/
021633 045 030501 030460 COD01: .ASCIZ /%A101 ERROR IN IO CONTROL REGISTER TEST%S13%AM8966(14) 15%N/
021732 051445 030065 040445 COD01A: .ASCIZ /%S50%AM8967(12)%N/
021754 040445 030061 020062 COD02: .ASCIZ /%A102 FAILURE OF FRAME COUNTER TEST%S17%AM8966(14) 15%N/
022047 045 030501 031460 COD03: .ASCIZ /%A103 FAILURE OF IO SILO NON-PARITY ERROR DATA M8966(14) 16%N/
022155 045 020101 020040 COD03A: .ASCIZ /%A TEST OR THE WRITE FLAG%S24%AM8963(11)%N/
022233 045 030501 032060 COD04: .ASCIZ /%A104 FAILURE OF IO SILO PARITY ERROR TEST OR M8966(14) 17%N/
022341 045 020101 020040 COD04A: .ASCIZ /%A DATA LATE TEST%N/
022370 040445 030061 020065 COD05: .ASCIZ /%A105 FAILURE OF SHIFT LOOP WITH ZEROES%S13%AM8965(15) 20%N/
022467 045 030501 033060 COD06: .ASCIZ /%A106 FAILURE OF SHIFT LOOP WITH ONES%S15%AM8965(15) 21%N/
022564 040445 030061 020067 COD07: .ASCIZ /%A107 FAILURE OF SHIFT LENGTH MUX%S19%AM8965(15) 22%N/
022655 045 030501 030061 COD10: .ASCIZ /%A110 FAILURE TO RECEIVE CORRECT OP-CODE FROM TS1! M8965(15) 47%N/
022763 045 020101 020040 COD10A: .ASCIZ /%A WHEN DATA SENT OVER THE SERIAL BUS%S12%ATS11,MOTHER BOARD%N/
023065 045 032523 022460 COD10B: .ASCIZ /%S50%ASERIAL BUS CABLE%N/
023116 040445 030461 020061 COD11: .ASCIZ /%A111 FAILURE OF 1KHZ CLOCK TEST%S20%AG159%S9%A2%N/
023201 045 020101 020040 COD11A: .ASCIZ /%A TSTS TAC SYNC FLOP AND ATTN%S19%ACBUS CABLE%N/
023265 045 041501 042510 COD11B: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159 M8963(11)%N/
023365 045 030501 031061 COD12: .ASCIZ /%A112 LIGHT REG CHANGED WHEN MOTION REG WAS CLEARED G159%S9%A3,4%N/
023470 040445 044103 041505 COD12A: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159%N/
023550 040445 030461 020063 COD13: .ASCIZ /%A113 FWD OR MVG BITS WRONG AFTER 1 TICK OF%S9%AG159%S9%A3,4%N/
023647 045 020101 020040 COD13A: .ASCIZ /%A SIMULATED COMMAND AND TACH PULSES%N/
023721 045 041501 042510 COD13B: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159%N/
024001 045 030501 032061 COD14: .ASCIZ /%A114 FAILURE OF SIMULATED CAPSTAN SPEED TEST:%S6%AG159%S9%A3,4%N/
024103 045 020101 020040 COD14A: .ASCIZ /%A CAPSTAN SPEED COUNTER OUT OF RANGE WHEN%N/
024163 045 020101 020040 COD14B: .ASCIZ /%A TAPE MOTION AT SPEED WAS SIMULATED%N/
024236 040445 044103 041505 COD14C: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159%N/
024316 040445 030461 020065 COD15: .ASCIZ /%A115 FAILURE OF SIMULATED SLOW CAPSTAN TEST:%S7%AG159%S9%A3,4%N/
024417 045 020101 020040 COD15A: .ASCIZ /%A SPEED COUNTER NOT LATCHED UP WITH MAX COUNT%N/
024503 045 020101 020040 COD15B: .ASCIZ /%A WHEN SLOW TACK TICKS WERE SIMULATED%N/
024557 045 041501 042510 COD15C: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159%N/
024637 045 030501 033061 COD16: .ASCIZ /%A116 FAILURE OF SIMULATED CAPSTAN DECEL TEST:%S6%AG159%S9%A3,4%N/
024741 045 020101 020040 COD16A: .ASCIZ /%A COUNTER NOT ZERO FOR FORWARD OR 377 FOR%N/
025021 045 020101 020040 COD16B: .ASCIZ /%A REVERSE WHILE DECELERATING OR MVG BIT=0%N/
025103 045 041501 042510 COD16C: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159%N/
025163 045 030501 033461 COD17: .ASCIZ /%A117 FAILURE OF MOVING FLOP TO RESET AFTER STOP%S4%AG159%S9%A3,4%N/
025267 045 020101 020040 COD17A: .ASCIZ /%A (DIRECTION REVERSAL FOR ONE TACH TICK)%N/
025346 040445 044103 041505 COD17B: .ASCIZ /%ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159%N/
025426 040445 031061 020060 COD20: .ASCIZ /%A120 FAILURE OF WRITE BOARD TO TURN ON AND EMPTY M8929(13) 23%N/
025534 040445 020040 020040 COD20A: .ASCIZ /%A THE SILO OR DATA LATE BIT NOT WORKING M8966(14)%N/
025634 040445 031061 020061 COD21: .ASCIZ /%A121 FAILURE OF WRITE BOARD TO EMPTY SILO AT M8929(13) 23%N/
025742 040445 020040 020040 COD21A: .ASCIZ /%A CORRECT SPEED%N/
025770 040445 031061 020062 COD22: .ASCIZ /%A122 UNDEFINED ERROR CODE%N/
026025 045 030501 031462 COD23: .ASCIZ /%A123 UNDEFINED ERROR CODE%N/
026062 040445 031061 020064 COD24: .ASCIZ /%A124 FORMATTER CTRL BOARD: FORMATTER FLAG FAILED M8922(7) 24%N/
026170 040445 044103 041505 COD24A: .ASCIZ /%ACHECK VCO ADJUST ON M8922%N/
026226 040445 031061 020065 COD25: .ASCIZ /%A125 FORMATTER SILO FILLING AND DATA ERROR M8922(7) 24%N/
026334 040445 044103 041505 COD25A: .ASCIZ /%ACHECK VCO ADJUST ON M8922%S25%AM8923(3)%N/
026410 040445 044103 041505 COD25B: .ASCIZ /%ACHECK SKEW, THRESHOLD ADJUSTS ON M8923%S12%AM8924(4,5,6)%N/
026505 045 030501 033062 COD26: .ASCIZ /%A126 PEAK SHIFT TEST ERROR%S25%AM8922(7)%S5%A25%N/
026570 040445 044103 041505 COD26A: .ASCIZ /%ACHECK VCO ADJUST ON M8922%S25%AM8923(3)%N/
026644 040445 044103 041505 COD26B: .ASCIZ /%ACHECK SKEW, THRESHOLD ADJUSTS ON M8923%S12%AM8924(4,5,6)%N/

```

026741 045 030501 033462 COD27: .ASCIZ /%A127 FORMATTER TABLE LOOKUP ROM CHECKSUM TEST ERR M8922(7) 26%N/
027047 045 041501 042510 COD27A: .ASCIZ /%ACHECK VCO ADJUST ON M8922%S25%AM8923(3)%N/
027123 045 041501 042510 COD27B: .ASCIZ /%ACHECK SKEW, THRESHOLD ADJUSTS ON M8923%S12%AM8924(4,5,6)%N/

027220 047445 022463 020101 CODXXX: .ASCIZ /%O3%A UNDEFINED ERROR CODE%N/

027256 040445 031463 020067 COD337: .ASCIZ /%A337 CAPSTAN RUNAWAY%N/
.LIST BEX
.EVEN

2571
2572
2573 027310

ENDMOD

.SBTTL LOAD DEV PROTECTION TABLE

2574
2575
2576
2577
2578
2579
2580

:++
:TABLE FOR SUPERVISOR TO IDENTIFY P-TBL FOR LOAD DEV
:AND TO WARN OPERATOR WHEN HE TRIES TO TEST THE LOAD DEVICE
:--

2581
2582 027310
2583 027310
2584 027310 000000
2585 027312 177777
2586 027314 177777
2587 027316

BGNPRCT
L\$PROT::
.WORD 0
.WORD -1
.WORD -1
ENDPROT

:P-TBL OFFSET OF TSSR
:P-TBL OFFSET OF MASS BUS UNIT#: -1 = NOT A MASS BUS DEV
:P-TBL OFFSET OF DRIVE #: -1 = NONE, ONE DRIVE PER UNIBU

```

2588
2589
2590          .TITLE MISCELLANEOUS SECTIONS
2591 027316    .SBTTL INITIALIZE SECTION
2592          BGNMOD
2593          :++
2594          : THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
2595          : AT THE BEGINNING OF EACH PASS.
2596          :--
2597 027316    BGNINIT
2598 027316    L$INIT::
2600
2601 027316 032727 000003 002230          BIT      #BIT0:BIT11,#CMDPKT          ;;IS CMD PKT ON A MODULO 4 BOUNDARY?
2602 027324 001421          BEQ      2$          ;;BR IF SO. OK.
2603 027326          1$:      ERRSF  1,MODUER          ;;IF NOT, TELL HIM THE PROGRAM IS SCREWED UP.
2604 027326 104454          TRAP      C$ERSF
2605 027330 000001          .WORD    1
2606 027332 002500          .WORD    MODUER
2607 027334 000000          .WORD    0
2608 027336          DELAY 2000.          ;GO TO THE SUPERVISOR, WAIT 1 SECOND.
2609 027336 012727 003720          MOV      #2000.,(PC)+
2610 027342 000000          .WORD    0
2611 027344 013727 002116          MOV      L$DLY,(PC)+
2612 027350 000000          .WORD    0
2613 027352 005367 177772          DEC      -6(PC)
2614 027356 001375          BNE      -4
2615 027360 005367 177756          DEC      -22(PC)
2616 027364 001367          BNE      -20
2617 027366 000757          BR       1$          ;;TELL HIM AGAIN IF HE INSISTS ON CONTINUING.
2618
2619 027370          2$:      READEF #EF.NEW          ;IS THIS A NEW PASS?
2620 027370 012700 000035          MOV      #EF.NEW,RO
2621 027374 104447          TRAP      C$REFG
2622 027376          BNCOMPLETE 3$          ;BR IF NOT.
2623 027376 103003          BCC      3$
2624 027400          LET LUNIT := #-1          ;INIT THE LOGICAL UNIT #.
2625 027400 012737 177777 027574          MOV      #-1,LUNIT
2626 027406          3$:      READEF #EF.PWR          ;HAS THERE BEEN A POWER FAILURE?
2627 027406 012700 000034          MOV      #EF.PWR,RO
2628 027412 104447          TRAP      C$REFG
2629 027414          BCOMPLETE 4$          ;BRANCH IF SO - KEEP CURRENT UNIT #.
2630 027414 103402          BCS      4$
2631 027416          LET LUNIT := LUNIT + #1          ;UPDATE UNIT #.
2632 027416 005237 027574          INC      LUNIT
2633 027422          4$:      LET RO := LUNIT          ;PREPARE TO PASS # TO SUPER.
2634 027422 013700 027574          MOV      LUNIT,RO
2635 027426          IF RO GT #3 THEN          ;IF # IS PASS THE LIMIT THEN:
2636 027426 020027 000003          CMP      RO,#3
2637 027432 003401          BLE      50113$
2638 027434          DOCLN          ;DO CLEANUP AND TERMINATE PASS.
2639 027434 104444          TRAP      C$DCLN
2640 027436          ENDIF
2641 027436          50113$:
2642
2643 027436          GPHARD RO,RO          ;SETUP TO RETRIEVE HRD P-TABLE DATA.

```

```

2644 027436 104442
2645 027440
2646 027440 103362
2647 027442 011037 002224
2648 027446 013737 002224 002214
2649 027454 162737 000002 002214
2650 027462 013737 002214 002220
2651 027470 013737 002220 002222
2652 027476 062737 000001 002222
2653 027504 013737 002214 002216
2654 027512 062737 000001 002216
2655 027520 016037 000002 002226
2656 027526
2657 027526 013705 027574
2658 027532 006305
2659 027534
2660 027534 012746 000340
2661 027540 016546 002306
2662 027544 013746 002226
2663 027550 012746 000003
2664 027554 104437
2665 027556 062706 000010
2666 027562
2667 027562 013737 027574 002422
2668 027570
2669 027570 104432
2670 027572 000004
2671
2672
2673
2674 027574 000000
2675
2676
2677
2678 027576
2679 027576
2680 027576 104411
2681
2682
2683
2684
2685
2686
2687
2688
2689 027600
2690 027600
2691
2692 027600
2693 027600 105037 030050
2694 027604
2695 027604 012746 000340
2696 027610 012746 030042
2697 027614 012746 000004
2698 027620 012746 000003
2699 027624 104437

; LOCAL STORAGE THAT IS USED ONLY DURING THE INITIALIZE SECTION.
LUNIT: .WORD 0 ;CURRENT LOGICAL UNIT #.
.EVEN
ENDINIT
L10031:
.SBTTL AUTO DROP SECTION
; ++
; SECTION EXECUTED AFTER THE INIT CODE WHEN "ADR" FLAG IS SET BY OPERATOR
; SECTION CHECKS FOR A VALID INTERFACE LOCATION. DROPS UNIT IF NO RESPONSE
; FROM INTERFACE
; --
BGNAUTO
L$AUTO::
LET TRAPD4 :B= #0
SETVEC #4,#TRAP4,#PRI07
CLRB TRAPD4
MOV #PRI07,-(SP)
MOV #TRAP4,-(SP)
MOV #4,-(SP)
MOV #3,-(SP)
TRAP C$SVEC

BNCOMPLETE 3$ ;BR IF THIS UNIT HAS BEEN DROPPED.
TRAP C$GPHRD
BCC 3$

MOV (R0),TSSR ;GET THE TSSR ADDR.
MOV TSSR,TSDB ;CALCULATE THE
SUB #2,TSDB ;TSDB ADDRESS.
MOV TSDB,TSBA ;LOAD TSBA ADDRESS.
MOV TSBA,TSBAHI ;CALCULATE THE TSBA
ADD #1,TSBAHI ;HI BYTE ADDRESS.
MOV TSDB,TSDBHI ;CALCULATE THE
ADD #1,TSDBHI ;TSDB HI BYTE ADDR.
MOV 2(R0),TSVCT ;GET THE VECTOR ADDRESS.
LET R5 := LUNIT SHIFT 1 ;GET LOGICAL UNIT # X 2.

MOV LUNIT,R5
ASL R5
SETVEC TSVCT,TS4INT(R5),#INTPRI ;SET UP INTERRUPT PROCESSING CONDITIONS.
MOV #INTPRI,-(SP)
MOV TS4INT(R5),-(SP)
MOV TSVCT,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP

LET UNIT := LUNIT ;SET UP UNIT # FOR PRINTOUTS.
MOV LUNIT,UNIT

TRAP C$EXIT
.WORD L10031-.
    
```

2700	027626	062706	000010			ADD	#10,SP
2701	027632			LET R1 := @TSSR		MOV	@TSSR,R1
2702	027632	017701	152366				
2703	027636			CLRVEC #4		MOV	#4,R0
2704	027636	012700	000004			TRAP	C\$CVEC
2705	027642	104436					
2706	027644			IFB TRAPD4 NE #0 THEN		TSTB	TRAPD4
2707	027644	105737	030050			BEQ	50114\$
2708	027650	001415					
2709	027652			PRINTF #AUTODM,TSSR		MOV	TSSR,-(SP)
2710	027652	013746	002224			MOV	#AUTODM,-(SP)
2711	027656	012746	027746			MOV	#2,-(SP)
2712	027662	012746	000002			MOV	SP,R0
2713	027666	010600				TRAP	C\$PNTF
2714	027670	104417				ADD	#6,SP
2715	027672	062706	000006				
2716	027676	004737	014570	JSR PC,DROPU			
2717	027702			ELSE			
2718	027702	000420				BR	50115\$
2719	027704					50111\$:	
2720	027704			IF #TS.SSR NOTSETIN @TSSR THEN			
2721	027704	032777	000200	152312		BIT	#TS.SSR,@TSSR
2722	027712	001014				BNE	50116\$
2723	027714			PRINTF #SSROFF,UNIT			
2724	027714	013746	002422			MOV	UNIT,-(SP)
2725	027720	012746	002566			MOV	#SSROFF,-(SP)
2726	027724	012746	000002			MOV	#2,-(SP)
2727	027730	010600				MOV	SP,R0
2728	027732	104417				TRAP	C\$PNTF
2729	027734	062706	000006			ADD	#6,SP
2730	027740	004737	014570	JSR PC,DROPU			
2731	027744			ENDIF			
2732	027744					50116\$:	
2733	027744			ENDIF		50115\$:	
2734	027744						
2735							
2736	027744			ENDAUTO			
2737	027744			L10032:			
2738	027744	104461				TRAP	C\$AUTO
2739							
2740	027746	040445	052502	020123	AUTODM: .ASCII /%ABUS TRAP AT %06%N/		
2741	027754	051124	050101	040440			
2742	027762	020124	047445	022466			
2743	027770	116					
2744	027771	045	044501	052116	.ASCIZ /%AINTERFACE BAD OR NOT SET TO ABOVE AD%N/		
2745	027776	051105	040506	042503			
2746	030004	041040	042101	047440			
2747	030012	020122	047516	020124			
2748	030020	042523	020124	047524			
2749	030026	040440	047502	042526			
2750	030034	040440	022504	000116			
2751					.EVEN		
2752							
2753					: DEVICE BUS TRAP HANDLER		
2754					: OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4		
2755					: 0: NO TRAP		

2756						
2757	030042			TRAP4: LET TRAPD4 :B= TRAPD4 + #1		
2758	030042	105237	030050			INCB TRAPD4
2759	030046	000002		RTI		
2760						
2761	030050	000		TRAPD4: .BYTE 0		;TRAPED AT 4 FLAG
2762		030052		.EVEN		
2763						


```
2764 .SBTTL CLEANUP CODING SECTION
2765
2766 ;++
2767 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2768 ; AT THE END OF EACH PASS.
2769 ;--
2770
2771 030052 BGNCLN
2772 030052 L$CLEAN::
2773
2774 030052 CLRVEC TSVCT ;RELEASE THE INTERRUPT VECTOR.
2775 030052 013700 002226 MOV TSVCT,RO
2776 030056 104436 TRAP C$CVEC
2777 030060 SETPRI PRI07
2778 030060 013700 000340 MOV PRI07,RO
2779 030064 104441 TRAP C$SPRI
2780
2781 030066 EXIT CLN
2782 030066 104432
2783 030070 000002 TRAP C$EXIT
2784 .WORD L10033-.
2785
2786 .EVEN
2787
2788 030072 ENDCLN
2789 030072 L10033:
2790 030072 104412 TRAP C$CLEAN
```

```
2791 .SBTTL DROP UNIT SECTION
2792
2793 :++
2794 : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2795 : TO NO LONGER BE TESTED.
2796 :--
2797
2798 030074 BGNDU
2799 030074 L$DU::
2800
2801
2802 030074 000240 NOP
2803 030076 000240 NOP
2804 030100 000240 NOP
2805 030102 000240 NOP
2806
2807 030104 EXIT DU
2808 030104 000167 .WORD JSJMP
2809 030106 000000 .WORD L10034-2-
2810
2811
2812 .EVEN
2813
2814 030110 ENDDU
2815 030110 L10034:
2816 030110 104453 TRAP C$DU
```

```
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825 030112  
2826 030112  
2827  
2828  
2829 030112 000240  
2830 030114 000240  
2831 030116 000240  
2832 030120 000240  
2833  
2834 030122  
2835 030122 000167  
2836 030124 000000  
2837  
2838  
2839  
2840  
2841 030126  
2842 030126  
2843 030126 104452  
2844  
2845 030130  
2846
```

```
.SBTTL ADD UNIT SECTION  
:++  
: THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
: TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF  
: 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.  
:--  
EGNAU  
L$AU::  
  
NOP  
NOP  
NOP  
NOP  
  
EXIT AU  
  
L10035:  
  
ENDAU  
  
ENDMOD  
  
TRAP CSAU  
  
.WORD JSJMP  
.WORD L10035-2-
```

```

2847 .TITLE HARDWARE TESTS
2848
2849 .SBTTL TEST 1: PDP11/TS11 WRAP TEST.
2850 030130 BGNMOD
2851
2852 :++
2853 : TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND THE TS11 BY
2854 : WRAPPING THE FOLLOWING PATTERNS:
2855 : A 1 IN A FIELD OF 0'S; A 0 IN A FIELD OF 1'S.
2856
2857 : WHEN DATA IS WRITTEN TO THE TSDB HI BYTE, THE DATA IS WRAPPED
2858 : AROUND WIHTIN THE TS11 AND APPEARS IN THE TSBA LO
2859 : AND HI BYTES. THE 2 LOW ORDER BITS OF THE DATA WILL BE
2860 : REFLECTED IN THE TSSR EXTENDED ADDRESS BITS.
2861 : R4 CONTAINS A COPY OF THE DATA SENT.
2862 : R3 CONTAINS THE EXPECTED TSBA RESULTS
2863 : R2 CONTAINS THE EXPECTED STATE OF THE TWO EXTENDED ADDRESS
2864 : BITS IN THE TSSR.
2865 :--
2866
2867
2868
2869 030130 BGNTST
2870 030130 T1::
2871
2872 030130 BGNSUB
2873 030130 T1.1:
2874 030130 104402 TRAP C$BSUB
2875 030132 BGNSEG TRAP C$BSEG
2876 030132 104404 ;RESET THE BUS. TRAP C$RESET
2877 030134 BRESET
2878 030134 104433 IF #TS.SSR SETIN @TSSR THEN
2879 030136 BIT #TS.SSR,@TSSR
2880 030136 032777 000200 152060 BEQ 50117$
2881 030144 001416
2882 030146 ERRDF 29,SSRON TRAP C$ERDF
2883 030146 104455 .WORD 29
2884 030150 000035 .WORD SSRON
2885 030152 002623 .WORD 0
2886 030154 000000
2887 030156 PRINTB #TS11BD MOV #TS11BD,-(SP)
2888 030156 012746 005037 MOV #1,-(SP)
2889 030162 012746 000001 MOV SP,R0
2890 030166 010600 TRAP C$PNTB
2891 030170 104414 ADD #4,SP
2892 030172 062706 000004 CALL DROPU JSR PC,DROPU
2893 030176 CALL DROPU
2894 030176 004737 014570
2895 030202 ENDIF
2896 030202 50117$:
2897 030202 ENDSEG
2898 030202 10000$:
2899 030202 104405 TRAP C$ESEG
2900 030204 CALL WAITSR JSR PC,WAITSR
2901 030204 004737 014616
2902 030210 ENDSUB
  
```

```

2903 030210 L10037:
2904 030210 104403 TRAP C$ESUB
2905 030212 012704 000200 MOV #200,R4 ;INIT THE DATA.
2906 ;REPEAT FOLLOWING UNTIL DATA ALL SHIFTED
2907 030216 005002 T1DATA: CLR R2 ;R2 WILL HOLD X ADDR BITS.
2908 030220 032704 000001 BIT #BIT0,R4 ;BIT 0 SET IN DATA?
2909 030224 001402 BEQ 1$ ;BR IF NOT.
2910 030226 052702 000400 BIS #TS.XA0,R2 ;IF SO SET THE BIT.
2911 030232 032704 000002 1$: BIT #BIT1,R4 ;BIT 1 SET IN DATA?
2912 030236 001402 BEQ 2$ ;BR IF NOT.
2913 030240 052702 001000 BIS #TS.XA1,R2 ;IF SO SET THE BIT.
2914 ;R2 NOW CONTAINS THE EX ADDR BITS
2915 ;EXPECTED IN THE TSSR.
2916 030244 005003 2$: CLR R3 ;CLR TSBA EXP RESULTS REG.
2917 030246 150403 BISB R4,R3 ;SETUP R3 TO LOOK LIKE
2918 030250 000303 SWAB R3 ;THE TSBA (EXPECTED RESULTS).
2919 030252 150403 BISB R4,R3 ;R3 NOW CONTAINS THE TSBA EXPECTED RESULTS.
2920 030254 BGNSUB
2921 030254 T1.2:
2922 030254 104402 TRAP C$BSUB
2923 030256 BGNSEG
2924 030256 104404 TRAP C$BSEG
2925 030260 004737 014616 JSR PC,WAITSR ;GO WAIT FOR SSR BIT TO SET.
2926 030264 IF #TS.SSR SETIN @TSSR THEN
2927 030264 032777 000200 151732 BIT #TS.SSR,@TSSR
2928 030272 001402 BEQ 50120$
2929 030274 004737 030332 JSR PC,T1WRAP
2930 030300 ENDIF
2931 030300 50120$:
2932 030300 ENDSEG
2933 030300 10000$:
2934 030300 104405 TRAP C$ESEG
2935 030302 005077 151716 CLR @TSSR ;SUBSYSTEM INIT.
2936 030306 CALL WAITSR
2937 030306 004737 014616 JSR PC,WAITSR
2938 030312 CALL SHWRAP JSR PC,SHWRAP
2939 030312 004737 016020 ENDSUB
2940 030316 L10040:
2941 030316 TRAP C$ESUB
2942 030316 104403
2943 030320 120427 000377 CMPB R4,#377
2944 030324 001334 BNE T1DATA
2945
2946 030326 EXIT TST
2947 030326 104432 TRAP C$EXIT
2948 030330 000132 .WORD L10036-
2949
2950 ;TEST 1 SUBR TO WRAP AND CHECK DATA
2951 030332 110477 151660 T1WRAP: MOV R4,@TSDBHI ;SEND THE DATA.
2952 030336 DELAY 10. ;GO TO SUPERVISOR, WAIT 1 MSEC.
2953 030336 012727 000012 MOV #10.,(PC)+
2954 030342 000000 .WORD 0
2955 030344 013727 002116 MOV LSDLY,(PC)+
2956 030350 000000 .WORD 0
2957 030352 005367 177772 DEC -6(PC)
2958 030356 001375 BNE .-4

```

```

2959 030360 005367 177756
2960 030364 001367
2961 030366 017737 151626 002432      MOV @TSBA,TEMP1      ;SAVE THE TSBA REG.
2962 030374
2963 030374 020337 002432      IF R3 NE TEMP1 THEN
2964 030400 001406
2965 030402      ERRDF 3,WRPER1,WRAPR1
2966 030402 104455
2967 030404 000003
2968 030406 002706
2969 030410 006044
2970 030412      CALL DROPU
2971 030412 004737 014570
2972 030416      ENDIF
2973 030416
2974 030416 017737 151602 002434      MOV @TSSR,TEMP2      ;GET TSSR.
2975 030424 012701 001400      MOV #TS.XA0!TS.XA1,R1 ;MASK OFF THE XADDR BITS.
2976 030430 005101      COM R1                ;BIT CLR ALL BUT THE
2977 030432 040137 002434      BIC R1,TEMP2          ;EXTENDED ADDR BITS.
2978 030436
2979 030436 020237 002434
2980 030442 001406
2981 030444      ERRDF 4,WRPER2,WRAPR4
2982 030444 104455
2983 030446 000004
2984 030450 002740
2985 030452 006544
2986 030454      CALL DROPU
2987 030454 004737 014570
2988 030460      ENDIF
2989 030460
2990 030460 000207      RTS PC                50122$:
2991
2992
2993
2994
2995
2996      .EVEN
2997
2998 030462      L10036:      ENDTST
2999 030462
3000 030462 104401      TRAP C$ETST
3001

```

```

3002 .SBTTL TEST 2: PDP11/TS04 WRAP TEST.
3003
3004 : TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND
3005 : THE TS04 BY WRAPPING THE FOLLOWING PATTERN:
3006 : A 1 IN A FIELD OF 0'S; A 0 IN A FIELD OF 1'S
3007
3008 : WHEN THE DATA IS WRITTEN TO THE TSDB LO BYTE, THE DATA IS
3009 : SENT TO THE TS04, VIA THE SERIAL LINE, WHERE IT IS WRAPPED
3010 : AROUND BACK OVER THE SERIAL LINE TO THE TS11. THE DATA THEN
3011 : APPEARS IN THE TSBA LO AND TSSR LO BYTES.
3012 : R4 CONTAINS THE EXPECTED TSBA RESULTS AND THE
3013 : EXPECTED TSSR RESULTS.
3014
3015 030464 BGNTST
3016 030464 T2::
3017
3018 030464 012704 000200 MOV #200,R4 ;INIT THE DATA.
3019 ;REPEAT FOLLOWING UNTIL DATA ALL SHIFTED
3020 030470 005077 151530 T2WRAP: CLR @TSSR
3021 030474 CALL WAITSR
3022 030474 004737 014616 JSR PC,WAITSR
3023 030500 032777 000200 151516 BIT #TS.SSR,@TSSR
3024 030506 001452 BEQ T2SHFT
3025 030510 BGNSUB
3026 030510 T2.1:
3027 030510 104402 TRAP C$BSUB
3028 030512 110477 151476 MOVB R4,@TSDB ;SEND DATA.
3029 030516 DELAY 10. ;GO TO SUPERVISOR, WAIT 1 MSEC.
3030 030516 012727 000012 MOV #10.,(PC)+
3031 030522 000000 .WORD 0
3032 030524 013727 002116 MOV L$DLY,(PC)+
3033 030530 000000 .WORD 0
3034 030532 005367 177772 DEC -6(PC)
3035 030536 001375 BNE -.4
3036 030540 005367 177756 DEC -22(PC)
3037 030544 001367 BNE -.20
3038 030546 017737 151446 002432 MOV @TSBA,TEMP1 ;SAVE TSBA.
3039 030554 IFB TEMP1 NE R4 THEN
3040 030554 123704 002432 CMPR TEMP1,R4
3041 030560 001406 BEQ 50123$
3042 030562 ERRDF 5,WRPER3,WRAPR2
3043 030562 104455 TRAP C$ERDF
3044 030564 000005 .WORD 5
3045 030566 003015 .WORD WRPER3
3046 030570 006220 .WORD WRAPR2
3047 030572 CALL DROPU
3048 030572 004737 014570 JSR PC,DROPU
3049 030574 ENDIF
3050 030576 50123$:
3051 030576 017737 151422 002434 MOV @TSSR,TEMP2 ;SAVE TSSR.
3052 030604 IFB TEMP2 NE R4 THEN
3053 030604 123704 002434 CMPB TEMP2,R4
3054 030610 001406 BEQ 50124$
3055 030612 ERRDF 6,WRPER3,WRAPR3
3056 030612 104455 TRAP C$ERDF
3057 030614 000006 .WORD 6
  
```

3058	030616	003015						.WORD	WRPER3
3059	030620	006372						.WORD	WRAPR3
3060	030622			CALL	DROPU				
3061	030622	004737	014570					JSR	PC,DROPU
3062	030626			ENDIF					
3063	030626						501248:		
3064	030626			ENDSUB					
3065	030626			L10042:					
3066	030626	104403						TRAP	C\$ESUB
3067	030630			CALL	SHWRAP				
3068	030630	004737	016020					JSR	PC,SHWRAP
3069	030634	120427	000377	T2SHFT:	CMPB	R4,#377			
3070	030640	001313			BNE	T2WRAP			
3071									
3072									
3073					.EVEN				
3074									
3075	030642				ENDTST				
3076	030642			L10041:					
3077	030642	104401						TRAP	C\$ETST


```

3078 .SBTTL TEST 3: SET TS04 CHARACTERISTIC VERIFICATION.
3079
3080 ; THE FUNCTION OF THIS TEST IS TO ISSUE A "SET CHARACTERISTIC" COMMAND
3081 ; TO TELL THE TS04 WHERE IN CORE THE MESSAGE PACKET RESIDES.
3082
3083 030644 BGNTST
3084 030644 T3::
3085
3086 030644 BGNSUB
3087 030644 T3.1:
3088 030644 104402 TRAP C$BSUB
3089 030646 105037 002437 CLR CTLFLG
3090 030652 005077 151346 CLR @TSSR ;INIT TS11-TS04
3091 030656 004737 014616 JSR PC,WAITSR
3092 030662 IF #TS.NBA NOTSETIN @TSSR THEN ;NBA SHOULD BE SET SINCE A COMD
3093 030662 032777 002000 151334 BIT #TS.NBA,@TSSR
3094 030670 001006 BNE 50125$
3095 030672 ERRDF 20,SCHERR,SCHER1 ;WAS ISSUED WITHOUT SET CHAR ISSUED FIRST
3096 030672 104455 TRAP C$ERDF
3097 030674 000024 .WORD 20
3098 030676 003047 .WORD SCHERR
3099 030700 010200 .WORD SCHER1
3100 030702 CALL DROPUP ;IF NBA NOTSET, THEN CALL ERROR, DROP UNIT
3101 030702 004737 014570 JSR PC,DROPUP
3102 030706 ENDIF
3103 030706 50125$:
3104 030706 ENDSUB
3105 030706 L10044:
3106 030706 104403 TRAP C$ESUB
3107
3108 030710 BGNSUB
3109 030710 T3.2:
3110 030710 104402 TRAP C$BSUB
3111 030712 CALL SCHEXE JSR PC,SCHEXE
3112 030712 004737 015746 IF #TS.NBA SETIN @TSSR THEN
3113 030716 IF #TS.NBA SETIN @TSSR THEN
3114 030716 032777 002000 151300 BIT #TS.NBA,@TSSR
3115 030724 001406 BEQ 50126$
3116 030726 ERRDF 21,SCHERR,SCHER2 TRAP C$ERDF
3117 030726 104455 .WORD 21
3118 030730 000025 .WORD SCHERR
3119 030732 003047 .WORD SCHER2
3120 030734 010332 CALL DROPUP JSR PC,DROPUP
3121 030736 004737 014570 ENDIF
3122 030736 50126$:
3123 030742 IF @TSBA NE #MSGEND THEN
3124 030742
3125 030742
3126 030742 027727 151252 002276 CMP @TSBA,#MSGEND
3127 030750 001406 BEQ 50127$
3128 030752 ERRDF 22,SCHERR,SCHER3 TRAP C$ERDF
3129 030752 104455 .WORD 22
3130 030754 000026 .WORD SCHERR
3131 030756 003047 .WORD SCHER3
3132 030760 010446 CALL DROPUP
3133 030762

```

3134	030762	004737	014570			JSR	PC,DROPU
3135	030766			ENDIF			
3136	030766					50127\$:	
3137	030766			IF MSGPKT EQ #177777 OR MSGDFL EQ #177777 THEN			
3138	030766	023727	002260	177777		CMP	MSGPKT,#177777
3139	030774	001404				BEQ	50130\$
3140	030776	023727	002262	177777		CMP	MSGDFL,#177777
3141	031004	001006				BNE	50131\$
3142	031006					50130\$:	
3143	031006			ERRDF 23,SCHERR,SCHER3			
3144	031006	104455				TRAP	C\$ERDF
3145	031010	000027				.WORD	23
3146	031012	003047				.WORD	SCHERR
3147	031014	010446				.WORD	SCHER3
3148	031016			CALL DROPU			
3149	031016	004737	014570			JSR	PC,DROPU
3150	031022			ENDIF			
3151	031022					50131\$:	
3152	031022			LET R1 := @TSSR CLR.BY #FCMASK			
3153	031022	017701	151176			MOV	@TSSR,R1
3154	031026	042701	177717			BIC	#FCMASK,R1
3155	031032			LET R1 := R1 SHIFT -4			
3156	031032	006201				ASR	R1
3157	031034	006201				ASR	R1
3158	031036	006201				ASR	R1
3159	031040	006201				ASR	R1
3160	031042			SELECT R1 OF 3 VERIFY			
3161	031042	010146				MOV	R1,-(SP)
3162	031044	003403				BLE	50132\$
3163	031046	021627	000003			CMP	(SP),#3
3164	031052	003402				BLE	50133\$
3165	031054					50132\$:	
3166	031054	012716	000004			MOV	#4,(SP)
3167	031060					50133\$:	
3168	031060	006316				ASL	(SP)
3169	031062	060716				ADD	PC,(SP)
3170	031064	063607				ADD	@(SP)+,PC
3171	031066					50134\$:	
3172	031066	000010				.WORD	50140\$-50134\$
3173	031070	000022				.WORD	50137\$-50134\$
3174	031072	000110				.WORD	50136\$-50134\$
3175	031074	000120				.WORD	50135\$-50134\$
3176	031076			CASE 1			
3177	031076					50140\$:	
3178	031076			ERRDF 24,SCHERR,SCHER4			
3179	031076	104455				TRAP	C\$ERDF
3180	031100	000030				.WORD	24
3181	031102	003047				.WORD	SCHERR
3182	031104	010542				.WORD	SCHER4
3183	031106			CASE 2			
3184	031106	000437				BR	50135\$
3185	031110					50137\$:	
3186	031110			IF #TS.SPE SETIN @TSSR OR #BPE SETIN XSTAT2 THEN			
3187	031110	032777	020000	151106		BIT	#TS.SPE,@TSSR
3188	031116	001004				BNE	50141\$
3189	031120	032737	020000	002272		BIT	#BPE,XSTAT2

3190 031126 001405
3191 031130
3192 031130
3193 031130 104455
3194 031132 000031
3195 031134 003047
3196 031136 010706
3197 031140
3198 031140 000415
3199 031142
3200 031142
3201 031142 032737 040000 002272
3202 031150 001405
3203 031152
3204 031152 104455
3205 031154 000032
3206 031156 003047
3207 031160 011122
3208 031162
3209 031162 000404
3210 031164
3211 031164
3212 031164 104455
3213 031166 000033
3214 031170 003047
3215 031172 011260
3216 031174
3217 031174
3218 031174
3219 031174
3220 031174
3221 031174 000404
3222 031176
3223 031176
3224 031176 104455
3225 031200 000034
3226 031202 003047
3227 031204 014032
3228 031206
3229 031206
3230 031206
3231 031206
3232 031206 104403
3233 031210
3234 031210 012700 000000
3235 031214 104441
3236 031216
3237 031216 105037 002437
3238 031222
3239 031222 052737 000200 002240
3240 031230
3241 031230 012777 002240 150756
3242 031236
3243 031236 004737 014616
3244 031242
3245 031242 105737 002437

ERRDF 25, SCHERR, SCHER5
ELSE
IF #SIP SETIN XSTAT2 THEN
ERRDF 26, SCHERR, SCHER6
ELSE
ERRDF 27, SCHERR, SCHER7
ENDIF
ENDIF
CASE 3
ERRDF 28, SCHERR, SCHERO
ENDSELECT
ENDSUB
L10045:
SETPRI #PRI00
LET CTLFLG :B= #0
LET SCHPKT := SCHPKT SET.BY #IE.C
LET @TSDB := #SCHPKT
CALL WAITSR
IFB CTLFLG EQ #0 THEN

50141\$: BEQ 50142\$
TRAP C\$ERDF
.WORD 25
.WORD SCHERR
.WORD SCHER5
50142\$: BR 50143\$
BIT #SIP, XSTAT2
BEQ 50144\$
TRAP C\$ERDF
.WORD 26
.WORD SCHERR
.WORD SCHER6
50144\$: BR 50145\$
TRAP C\$ERDF
.WORD 27
.WORD SCHERR
.WORD SCHER7
50145\$:
50143\$:
50136\$: BR 50135\$
TRAP C\$ERDF
.WORD 28
.WORD SCHERR
.WORD SCHERO
50135\$:
TRAP C\$ESUB
MOV #PRI00, R0
TRAP C\$SPRI
CLRB CTLFLG
BIS #IE.C, SCHPKT
MOV #SCHPKT, @TSDB
JSR PC, WAITSR
TSTB CTLFLG

3246	031246	001014				BNE	50146\$
3247	031250				ERRDF #31,NINTM		
3248	031250	104455				TRAP	C\$ERDF
3249	031252	000037				.WORD	31
3250	031254	003503				.WORD	NINTM
3251	031256	000000				.WORD	0
3252	031260				PRINTB #IOBRD		
3253	031260	012746	005646			MOV	#IOBRD,-(SP)
3254	031264	012746	000001			MOV	#1,-(SP)
3255	031270	010600				MOV	SP,R0
3256	031272	104414				TRAP	C\$PNTB
3257	031274	062706	000004			ADD	#4,SP
3258	031300				ENDIF		
3259	031300					50146\$:	
3260	031300				LET SCHPKT := SCHPKT CLR.BY #IE.C		
3261	031300	042737	000200	002240		BIC	#IE.C,SCHPKT
3262	031306				SETPRI #PRI07		
3263	031306	012700	000340			MOV	#PRI07,R0
3264	031312	104441				TRAP	C\$SPRI
3265							
3266	031314				LET TS4CL := MSGPKT+MS\$XS2 CLR.BY #177400 ;SAVE MICRO-CODE REV	LEVEL.	
3267	031314	013737	002272	002424		MOV	MSGPKT+MS\$XS2,TS
3268	031322	042737	177400	002424		BIC	#177400,TS4CL
3269							
3270	031330				EXIT TST		
3271	031330	104432				TRAP	C\$EXIT
3272	031332	000002				.WORD	L10043-
3273					.EVEN		
3274							
3275	031334				ENDTST		
3276	031334				L10043:		
3277	031334	104401				TRAP	C\$ETST
3278							
3279							

3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322

.SBTTL TEST 4: TRACK INACTIVE/ACTIVE TEST.

:TESTS 4 - 7 PERFORM DATA WRAPS ON THE P.E. READ FORMATTER
:BOARDS. COMMUNICATION BETWEEN THE PDP11 AND TS04 OCCURS BY USING
:THE DIA (DIAGNOSTIC) COMMAND WHICH SENDS A COPY OF THE DIABLK TABLE,
:RESIDING IN CORE, TO THE TS04 CONTROLLER. THE FORMAT OF THE DIABLK IS
:SHOWN IN THE FOLLOWING TABLE. NOTE THAT THE TABLE IS FILLED IN REVERSE
:ORDER, THAT IS, THE LAST LOGICAL ENTRY OF THE TABLE IS LABELED DIABLK,
:WHILE THE FIRST LOGICAL ENTRY OF THE TABLE IS LABELED DIABLK+DIAEXT, WHERE
:DIAEXT IS THE LENGTH (EXTENT) OF THE TABLE IN BYTES.

:WHEN THE DIA COMMAND IS EXECUTED, THE DIABLK IS LOADED ONTO THE TS04 STACK
:WITH THE FIRST LOGICAL ENTRY AT THE TOP OF THE STACK, AS SHOWN BELOW.
:THE TS04 THEN JUMPS TO THE P.E. WRAP TASK, IN ROM, WHERE THE FUNCTION IS
:EXECUTED USING THE REMAINING STACK ENTRIES AS ARGUMENTS.

```

:
:
:          DIABLK+DIAEXT: TS04 PE WRAP TASK ADDR LO
:                          TS04 PE WRAP TASK ADDR HI
:                          READ CONTROL (RDCTLO)
:                          FORMAT CONTROL (FMCTLO)
:                          DATA
:                          CONTROL (IOSCO)
:                          DATA
:                          CONTROL (IOSCO)
:                          DATA
:                          CONTROL (IOSCO)
:                          DATA
:                          CONTROL (IOSCO)
:                          DATA
:                          CONTROL (IOSCO)
:                          DATA
:                          CONTROL (IOSCO)
:
:          DIABLK:      DATA
:                          CONTROL (IOSCO)

```

:TEST 4 CHECKS THAT THE TRACK ACTIVE FLOP CAN SET AND CLEAR
:IN NRZI MODE FOR ALL CHANNELS. IF THE DATA DOES NOT MAKE A TRANSITION
:WHEN THE WRITE FLAG IS UP, THE TRACK ACTIVE FLOP WILL CLEAR.
:HOWEVER, IT WILL SET IF THERE IS A DATA TRANSITION WHILE THE
:WRITE FLAG IS UP.

```

3323 031336          BGNTST
3324 031336          T4::
3325
3326 031336          CALL    SCHEXE          ;SET CHAR
3327 031336 004737 015746          JSR    PC,SCHEXE
3328 031342          CALL    WAITMT
3329 031342 004737 015174          JSR    PC,WAITMT
3330
3331          ; THIS SUBTEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING
3332          ; ALL 0'S DATA. THE PATTERN IS AS FOLLOWS FOR EACH CHANNEL:
3333          ;
3334          ; DATA:          000000
3335          ; WRTFLG:        011100
3336          ; TRACK ACTIVE:  SHOULD BE 0 FOR ALL TRACKS
3337          ;
3338          ;INITIALIZAITON
3339
3340 031346          BGNSUB
3341 031346          T4.1:
3342 031346 104402          TRAP    C$BSUB
3343
3344 031350 004737 016154          JSR    PC,PEINIT          ;GO LOAD DIABLK INDEX AND TS04 TASK ADDRESS
3345 031354 112745 000001          MOVB   #FC.NRZ,-(R5)      ;SET NRZ1 MODE IN FMT CNTRL.
3346
3347          ;SETUP DATA & CONTROL WORDS IN DIABLK.
3348
3349 031360 005003          CLR    R3                ;SETUP DATA FOR WORD 1
3350 031362 012704 000040          MOV    #IS.NRZ,R4        ;SETUP CONTROL WORD 1
3351 031366 004737 015626          JSR    PC,TALOAD         ;LOAD IT.
3352 031372 012704 000044          MOV    #IS.NRZ!IS.WRF,R4 ;SET WRITE FLAG IN CONTROL WORD 2
3353 031376 004737 015626          JSR    PC,TALOAD         ;LOAD DATA & CNTRL WORD 2
3354 031402 004737 015626          JSR    PC,TALOAD         ;LOAD DATA & CNTRL WORD 3
3355 031406 004737 015626          JSR    PC,TALOAD         ;LOAD DATA & CNTRL WORD 4
3356 031412 012704 000040          MOV    #IS.NRZ,R4        ;SETUP CNTRL WORD 5
3357 031416 004737 015626          JSR    PC,TALOAD         ;LOAD DATA & CNTRL WORD 5
3358 031422 004737 015626          JSR    PC,TALOAD         ;LOAD DATA & CNTRL WORD 6.
3359 031426 004737 015722          JSR    PC,DIAEXE        ;EXECUTE THE CMD.
3360
3361          ;NOW CHECK FOR ERRORS.
3362
3363 031432 005037 002436          CLR    ERRFLG           ;CLR THE ERROR FLAG.
3364 031436 005037 002342          CLR    EXTRAC           ;TRACK ACTIVE 0 FOR ALL CHAN
3365 031442 004737 016044          JSR    PC,TKACER        ;CHECK FOR ERRORS.
3366 031446
3367 031446 013737 002360 002406          LET   ACTRK1 := ACTRAC
3368 031454
3369 031454 105737 002436          IFB   ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
3370 031460 001407          TSTB   ERRFLG
3371 031462 105737 002212          BEQ   50147$
3372 031466 001404          TSTB   CMPFLG
3373 031470          BEQ   50147$
3374 031470 104456          EPRHRD 7,TAER1,TAEM
3375 031472 000007          TRAP   C$ERHRD
3376 031474 003100          .WORD 7
3377 031476 006716          .WORD TAER1
3378 031500          .WORD TAEM
          ENDIF

```

```
3379 031500 50147$:  
3380  
3381 031500 ENDSUB  
3382 031500 L10047:  
3383 031500 104403 TRAP C$ESUB  
3384  
3385 ;THIS SUBTEST FORCES TRACK ACTIVE TO SET BY WRITING THE FOLLOWING  
3386 ;PATTERN ON EACH CHANNEL:  
3387  
3388 : DATA: 110000  
3389 : WRTFLG: 011100  
3390 : TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.  
3391  
3392 031502 BGNSUB  
3393 031502 T4.2:  
3394 031502 104402 TRAP C$BSUB  
3395  
3396 ; INITIALIZATION  
3397  
3398 031504 004737 016154 JSR PC,PEINIT ;  
3399 031510 112745 000001 MOVB #FC.NRZ,-(R5) ;NRZI MODE  
3400  
3401 ;SET UP DATA & CNTRL WORDS IN DIABLK  
3402  
3403 031514 012703 177777 MOV #-1,R3 ;WORD 1 DATA  
3404 031520 012704 000040 MOV #IS.NRZ,R4 ;WORD 1 CONTROL  
3405 031524 004737 015626 JSR PC,TALOAD ;LOAD IT  
3406 031530 012704 000044 MOV #IS.NRZ!IS.WRF,R4 ;WORD 2 CONTROL--WRT FLAG SET  
3407 031534 004737 015626 JSR PC,TALOAD ;LOAD IT WORD 2 (DATA & CONTROL)  
3408 031540 005003 CLR R3 ;WORD 3 DATA.  
3409 031542 004737 015626 JSR PC,TALOAD ;LOAD DATA & CNTRL - WORD 3  
3410 031546 004737 015626 JSR PC,TALOAD ;LOAD DATA & CNTRL - WORD 4  
3411 031552 012704 000040 MOV #IS.NRZ,R4 ;CLR WRITE FLAG IN WORD 5  
3412 031556 004737 015626 JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 5  
3413 031562 004737 015626 JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 6  
3414 031566 004737 015722 JSR PC,DIAEXE ;DO THE TS04 WRAP.  
3415  
3416 ;CHECK FOR ERRORS  
3417  
3418 031572 005037 002436 CLR ERRFLG ;INIT ERROR FLAG.  
3419 031576 012737 000777 002342 MOV #777,EXTRAC ;SHOULD BE DATA.  
3420 031604 004737 016044 JSR PC,TKACER ;CHECK FOR ERRORS  
3421 031610  
3422 031610 013737 002360 002410 LET ACTRK2 := COMP ACTRAC MOV ACTRAC,ACTRK2  
3423 031616 005137 002410 COM ACTRK2  
3424 031622  
3425 031622 042737 177000 002410 LET ACTRK2 := ACTRK2 CLR.BY #177000 BIC #177000,ACTRK2  
3426 031630  
3427 031630 105737 012436 IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN TSTB ERRFLG  
3428 031634 001407 BEQ 50150$  
3429 031636 105737 002212 TSTB CMPFLG  
3430 031642 001404 BEQ 50150$  
3431 031644  
3432 031644 104456 TRAP C$ERHRD  
3433 031646 000011 .WORD 9  
3434 031650 003150 .WORD TAER2
```

```

3435 031652 006716 .WORD TAEM
3436 031654
3437 031654 50150$:
3438
3439 031654 ENDSUB
3440 031654 L10050:
3441 031654 104403 TRAP C$ESUB
3442
3443 ;THIS SUBTEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING THE FOLLOWING
3444 ;PATTERN ON EACH CHANNEL:
3445 :
3446 : DATA: 111111
3447 : WRTFLG: 011100
3448 : TRACK ACTIVE: SHOULD BE 0 FOR ALL TRACKS
3449
3450 031656 BGNSUB
3451 031656 T4.3:
3452 031656 104402 TRAP C$BSUB
3453
3454 ; INITIALIZATION
3455
3456 031660 004737 016154 JSR PC,PEINIT
3457 031664 112745 000001 MOVB #FC.NRZ,-(R5) ;NRZI MODE.
3458
3459 ;SET UP DATA AND CONTROL WORDS IN DIABLK.
3460
3461 031670 012703 177777 MOV #-1,R3 ;SET UP DATA.
3462 031674 012704 000040 MOV #IS.NRZ,R4 ;SET UP CNTRL WORD.
3463 031700 004737 015626 JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 1
3464 031704 012704 000044 MOV #IS.NRZ!IS.WRF,R4 ;WORD 2 CNTRL
3465 031710 004737 015626 JSR PC,TALOAD ;LOAD DATA - CNTRL WORD 2
3466 031714 004737 015626 JSR PC,TALOAD ;LOAD DATA - CNTRL WORD 3
3467 031720 004737 015626 JSR PC,TALOAD ;LOAD DATA - CNTRL WORD 4
3468 031724 012704 000040 MOV #IS.NRZ,R4 ;CNTRL WORD 5
3469 031730 004737 015626 JSR PC,TALOAD ;LOAD DATA - CNTRL WORD 5
3470 031734 004737 015626 JSR PC,TALOAD ;LOAD DATA - CNTRL WORD 6
3471 031740 004737 015722 JSR PC,DIAEXE ;DO THE WRAP.
3472
3473 ;CHECK FOR ERRORS
3474
3475 031744 005037 002436 CLR ERRFLG ;INIT ERROR FLAG.
3476 031750 005037 002342 CLR EXTRAC ;SHOULD BE DATA=0.
3477 031754 004737 016044 JSR PC,TKACER ;CHECK FOR ERRORS
3478 031760
3479 031760 013737 002360 002412 LET ACTRK3 := ACTRAC
3480 031766
3481 031766 105737 002436 IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
3482 031772 001407
3483 031774 105737 002212
3484 032000 001404
3485 032002
3486 032002 104456 TRAP C$ERHRD
3487 032004 000010 .WORD 8
3488 032006 003100 .WORD TAER1
3489 032010 006716 .WORD TAEM
3490 032012

```


50151\$:

3491 032012
3492
3493 032012
3494 032012
3495 032012 104403
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505 032014
3506 032014
3507 032014 104402
3508
3509
3510
3511 032016 004737 016154
3512 032022 112745 000001
3513
3514
3515
3516 032026 005003
3517 032030 012704 000040
3518 032034 004737 015626
3519 032040 012704 000044
3520 032044 004737 015626
3521 032050 012703 177777
3522 032054 004737 015626
3523 032060 004737 015626
3524 032064 012704 000040
3525 032070 004737 015626
3526 032074 004737 015626
3527 032100 004737 015722
3528
3529
3530
3531 032104 005037 002436
3532 032110 012737 000777 002342
3533 032116 004737 016044
3534 032122
3535 032122 013737 002360 002414
3536 032130 005137 002414
3537 032134
3538 032134 042737 177000 002414
3539 032142
3540 032142 105737 002436
3541 032146 001407
3542 032150 105737 002212
3543 032154 001404
3544 032156
3545 032156 104456
3546 032160 000012

ENDSUB
L10051:

TRAP C\$ESUB

;THIS SUBTEST FORCES THE TRACK ACTIVE FLOP TO SET BY WRITING THE FOLLOWING
;PATTERN ON EACH CHANNEL:

DATA: 001111
WRTFLG: 011100
TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.

BGNSUB
T4.4:

TRAP C\$BSUB

; INITIALIZATION.

JSR PC,PEINIT
MOVB #FC.NRZ,-(R5) ;NRZI MODE

;SET UP DATA AND CNTRL WORD IN DIABLK.

CLR R3 ;WORD 1 DATA
MOV #IS.NRZ,R4 ;WORD 1 CNTRL - WRITE FLAG CLR
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 1
MOV #IS.NRZ!IS.WRF,R4 ;WORD 2 CNTRL-SET WRT FLAG
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 2
MOV #-1,R3 ;WORD 3 DATA-DO A DATA TRANSITION
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 3
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 4
MOV #IS.NRZ,R4 ;CNTRL WORD 5 - CLR WRITE FLAG.
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 5
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 6
JSR PC,DIAEXE ;GO WRAP THE DATA.

;CHECK FOR ERRORS.

CLR ERRFLG ;INIT ERROR FLAG.
MOV #777,EXTRAC ;TRACK ACTIVE SHOULD BE SET FOR ALL TRACKS.
JSR PC,TKACER ;CHECK FOR ERRORS.

LET ACTRK4 := COMP ACTRAC

MOV ACTRAC,ACTRK4
COM ACTRK4

LET ACTRK4 := ACTRK4 CLR.BY #177000

BIC #177000,ACTRK4

IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN

TSTB ERRFLG
Q 50152\$
TB CMPFLG
BEQ 50152\$

ERRHRD 10,TAER2,TAEM

TRAP C\$ERHRD
.WORD 10

3547	032162	003150					.WORD	TAER2
3548	032164	006716					.WORD	TAEM
3549	032166			ENDIF				
3550	032166					50152\$:		
3551								
3552	032166			ENDSUB				
3553	032166		L10052:					
3554	032166	104403					TRAP	C\$ESUB
3555								
3556								
3557	032170			LET R1 := ACTRK1 OR ACTRK2 OR ACTRK3				
3558	032170	013701	002406				MOV	ACTRK1,R1
3559	032174	053701	002410				BIS	ACTRK2,R1
3560	032200	053701	002412				BIS	ACTRK3,R1
3561	032204			LET R1 := R1 OR ACTRK4				
3562	032204	053701	002414				BIS	ACTRK4,R1
3563	032210			IF R1 NE #0 THEN				
3564	032210	005701					TST	R1
3565	032212	001406					BEQ	50153\$
3566	032214			ERRHRD	11,TAERR			
3567	032214	104456					TRAP	C\$ERHRD
3568	032216	000013					.WORD	11
3569	032220	003220					.WORD	TAERR
3570	032222	000000					.WORD	0
3571	032224			CALL	MSORT1			
3572	032224	004737	017554				JSR	PC,MSORT1
3573	032230			ENDIF				
3574	032230					50153\$:		
3575								
3576				.EVEN				
3577								
3578	032230			ENDTST				
3579	032230		L10046:					
3580	032230	104401					TRAP	C\$ETST
3581								

3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593 032232
3594 032232
3595
3596 032232
3597 032232 004737 015746
3598 032236
3599 032236 004737 015174
3600 032242
3601 032242 005001
3602 032244 000402
3603 032246
3604 032246 062701 000002
3605 032252
3606 032252 020127 000012
3607 032256 003003
3608 032260
3609 032260 005061 002372
3610 032264
3611 032264 000770
3612 032266
3613
3614 032266
3615 032266
3616 032266 104402
3617
3618
3619
3620 032270 005037 002416
3621 032274 012702 000777
3622 032300 005003
3623 032302 004737 016126
3624 032306 004737 015722
3625
3626
3627
3628 032312 005037 002436
3629 032316 004737 016504
3630 032322
3631 032322 105737 002436
3632 032326 001407
3633 032330 105737 002212
3634 032334 001404
3635 032336
3636 032336 104456
3637 032340 000014

```
.SBTTL TEST 5: P.E. DATA TEST.

;TEST 5 WRAPS A DATA PATTERN TO CHECK EACH
;TRACK FOR BIT PICKUPS AND DROPS.
;
;REGISTER USAGE IS AS FOLLOWS:
; R2=PREAMBLE DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
; R3=1ST BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8. THIS
; IS THE DATA OF INTEREST AFTER EXECUTING THE TSO4 DIA COMMAND.
; R4=2ND BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.

T5::
    BGNTST
    CALL SCHEXE ;SET CHAR
    CALL WAITMT
    INCR R1 FROM #0 TO #12 BY #2
    CLR R1
    BR 50154$
    ADD #2,R1
    CMP R1,#12
    BGT 50156$
    CLR ORDTBL(R1) ;CLEAR 6 LOC OF ORED WRAP REG TABLE
    ENDINC
    BR 50155$

T5.1:
    BGNSUB
    TRAP C$BSUB

;THIS SUBTEST WRAPS AN ALL 0'S PATTERN.

    CLR DTKIDN ;CLR DEAD TRK REG.
    MOV #777,R2 ;SETUP THE PREAMBLE DATA.
    CLR R3 ;BYTE 2 DATA. ALL 0'S.
    JSR PC,DATBLD ;BUILD THE DIAG BLK.
    JSR PC,DIAEXE ;EXECUTE IT.

;CHECK FOR ERRORS.

    CLR ERRFLG ;INIT ERROR FLAG.
    JSR PC,PEERCK ;CHECK FOR ERRORS.
    IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN

    TSTB ERRFLG
    BEQ 50157$
    TSTB CMPFLG
    BEQ 50157$

    ERRHRD 12,DATER,SKDAEM

    TRAP C$ERHRD
    .WORD 12
```



```
3694 ;CHECK FOR ERRORS & REPORT IF ANY
3695
3696 032466 005037 002436 CLR ERRFLG ;INIT ERROR FLAG.
3697 032472 004737 016504 JSR PC,PEERCK ;CHECK FOR ERRORS
3698 032476 IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
3699 032476 105737 002436 TSTB ERRFLG
3700 032502 001407 BEQ 50161$
3701 032504 105737 002212 TSTB CMPFLG
3702 032510 001404 BEQ 50161$
3703 032512 ERRHRD 14,DATER,SKDAEM
3704 032512 104456 TRAP C$ERHRD
3705 032514 000016 .WORD 14
3706 032516 003243 .WORD DATER
3707 032520 007176 .WORD SKDAEM
3708 032522 ENDIF
3709 032522 50161$:
3710
3711 ;IF NOT DONE, SHIFT THE 1 AND DO IT AGAIN.
3712
3713 032522 000241 CLC ;CLEAR THE CARRY BIT
3714 032524 006203 ASR R3 ;SHIFT THE DATA
3715 032526 103353 BCC 1$ ;BRANCH IF THE SHIFTING '1' HASN'T
3716 ;BEEN SHIFTED INTO THE CARRY POSITION
3717 ;YET.
3718
3719 032530 ENDSUB
3720 032530 L10056:
3721 032530 104403 TRAP C$ESUB
3722
3723 032532 BGNSUB
3724 032532 T5.4:
3725 032532 104402 TRAP C$BSUB
3726
3727 ;THIS SUBTEST RIPPLES A 0 IN A FIELD OF 1'S.
3728
3729 032534 005037 002416 CLR DTKIDN ;CLR DEAD TRK IDN REG.
3730 032540 012702 000777 MOV #777,R2 ;SETUP THE PREAMBLE DATA IN R2 BITS 0-8.
3731 032544 012703 000377 MOV #377,R3 ;SETUP 1ST BYTE OF DATA.
3732 032550 004737 016126 JSR PC,DATBLD ;BUILD THE DIA BLK.
3733 032554 004737 015722 JSR PC,DIAEXE ;EXECUTE THE DATA WRAP.
3734
3735 ;CHECK FOR ERRORS & REPORT IF ANY.
3736
3737 032560 005037 002436 CLR ERRFLG ;INIT ERROR FLAG.
3738 032564 004737 016504 JSR PC,PEERCK ;CHECK FOR ERRORS
3739 032570 IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
3740 032570 105737 002436 TSTB ERRFLG
3741 032574 001407 BEQ 50162$
3742 032576 105737 002212 TSTB CMPFLG
3743 032602 001404 BEQ 50162$
3744 032604 ERRHRD 15,DATER,SKDAEM
3745 032604 104456 TRAP C$ERHRD
3746 032606 000017 .WORD 15
3747 032610 003243 .WORD DATER
3748 032612 007176 .WORD SKDAEM
3749 032614 ENDIF
```

```
3750 032614                                     50162$:
3751
3752
3753                                     ;IF NOT DONE, SHIFT THE 0 AND DO IT AGAIN.
3754
3755 032614 052703 001000                       BIS    #BIT9,R3       ;PREPARE FOR SHIFT.
3756 032620 006203                               ASR    R3              ;SHIFT THE DATA. HAS THE SHIFITNG 0
3757                                             ;REACHED THE CARRY BIT YET?
3758 032622 103752                               BCS    1$             ;IF NOT, CONTINUE.
3759
3760 032624                                     ENDSUB
3761 032624                                     L10057:
3762 032624 104403                                     TRAP   C$ESUB
3763
3764
3765 032626                                     LET R1 := ORORDY OR ORTRAC OR ORDATA
3766 032626 013701 002372                               MOV    ORORDY,R1
3767 032632 053701 002376                               BIS    ORTRAC,R1
3768 032636 053701 002400                               BIS    ORDATA,R1
3769 032642                                     LET R1 := R1 OR ORTRDD
3770 032642 053701 002402                               BIS    ORTRDD,R1
3771 032646                                     IF R1 NE #0 THEN
3772 032646 005701                                     TST    R1
3773 032650 001406                                     BEQ    50163$
3774 032652                                     ERRHRD    30,DATER
3775 032652 104456                                     TRAP   C$ERHRD
3776 032654 000036                                     .WORD  30
3777 032656 003243                                     .WORD  DATER
3778 032660 000000                                     .WORD  0
3779 032662                                     CALL  MSORT2
3780 032662 004737 020216                               JSR    PC,MSORT2
3781 032666                                     ENDIF
3782 032666                                     50163$:
3783
3784 032666                                     EXIT   TST
3785 032666 104432                                     TRAP   C$EXIT
3786 032670 000002                                     .WORD  L10053-.
3787
3788                                     .EVEN
3789 032672                                     L10053:
3790 032672                                     ENDTST
3791 032672 104401                                     TRAP   C$ETST
```

3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805 032674
3806 032674
3807
3808 032674
3809 032674 004737 015746
3810 032700
3811 032700 004737 015174
3812 032704
3813 032704 005001
3814 032706 000402
3815 032710
3816 032710 062701 000002
3817 032714
3818 032714 020127 000012
3819 032720 003003
3820 032722
3821 032722 005061 002372
3822 032726
3823 032726 000770
3824 032730
3825
3826 032730
3827 032730
3828 032730 104402
3829
3830
3831
3832
3833 032732 005037 002416
3834 032736 012702 000377
3835 032742 004737 016154
3836 032746 112745 000012
3837 032752 010203
3838 032754 005103
3839 032756 042703 177000
3840 032762 010204
3841 032764 004737 016176
3842 032770 005037 002344
3843 032774 012737 000777 002350
3844
3845 033002 005037 002340
3846 033006 004737 015722
3847

.SBTTL TEST 6: P.E. SKEW TEST

:THE NEXT TWO SUBTESTS SKEW THE DATA ON A TRACK BY ONE BYTE WITH RESPECT TO ALL
:THE OTHER TRACKS. THAT IS, THE DATA IS ONE BYTE LATE ON THE ONE TRACK.
:EACH TRACK IS TESTED FOR SKEW IN THIS MANNER. REGISTER ASSIGNMENTS
:ARE AS FOLLOWS:

:
: R2=PREAMBLE DATA
: R3=BYTE 1 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK. THAT
: TRACK CONTAINS PREAMBLE DATA)
: R4=BYTE 2 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK. THAT
: TRACK CONTAINS BYTE 1 DATA)

T6:: BGNTST

CALL SCHEXE

JSR PC,SCHEXE

CALL WAITMT

JSR PC,WAITMT

INCR R1 FROM #0 TO #12 BY #2

CLR R1

BR 50164\$

50165\$:

ADD #2,R1

50164\$:

CMP R1,#12

BGT 50166\$

LET ORDTBL(R1) := #0

;CLEAR 6 LOC OF ORED WRAP REG TABLES

CLR ORDTBL(R1)

ENDINC

BR 50165\$

50166\$:

T6.1: BGNSUB

TRAP C\$BSUB

:THIS SUBTEST WRITES AN ALL 1'S PREAMBLE (SKEWED), AN ALL 0'S
:BYTE 1 DATA (SKEWED), AND AN ALL 1'S BYTE 2 DATA (SKEWED).

1\$: CLR DTKIDN ;CLR DEAD TRK IDN REG.
MOV #377,R2 ;SET UP PREAMBLE DATA. START BY SKEWING TRK 9.
JSR PC,PEINIT ;SET WRAP TASK ADDR IN DIABLK AND INIT INDEX.
MOVB #FC.DAT!FC.VCO,-(R5) ;SET DATA & VCO MODE IN FORMAT CNTRL.
MOV R2,R3 ;SETUP BYTE 1 DATA
COM R3 ;=0
BIC #177000,R3 ;CLR GARBAGE BITS
MOV R2,R4 ;SETUP BYTE 2 DATA=1.
JSR PC,PEDATA ;LOAD DATA IN DIABLK
CLR EXDATA ;THE EXPECTED DESKEWED DATA IN EXDATA IS 0
MOV #777,EXODTR ;THE EXPECTED DESKEWED DATA IN EXODTR IS
;1 FOR EACH TRACK.
CLR EX1DTR ;THE EXPECTED DESKEWED DATA IN EX1DTR IS 0
JSR PC,DIAEXE ;EXECUTE THE DATA WRAP.

```

3848 ;CHECK FOR ERRORS AND REPORT IF ANY
3849
3850 033012 005037 002436 CLR ERRFLG ;INIT ERROR FLAG.
3851 033016 004737 016504 JSR PC,PEERCK ;CHECK FOR ERRORS
3852 033022 IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
3853 033022 105737 002436 TSTB ERRFLG
3854 033026 001407 BEQ 50167$
3855 033030 105737 002212 TSTB CMPFLG
3856 033034 001404 BEQ 50167$
3857 033036 ERRHRD 16,SKEWER,SKDAEM
3858 033036 104456 TRAP C$ERHRD
3859 033040 000020 .WORD 16
3860 033042 003270 .WORD SKEWER
3861 033044 007176 .WORD SKDAEM
3862 033046
3863 033046 ENDIF 50167$:
3864
3865 ;SHIFT THE SKEW TO THE NEXT TRACK AND REPEAT.
3866
3867 033046 052702 001000 BIS #BIT9,R2 ;PREPARE FOR SHIFT OF SKEWED TRK.
3868 033052 006202 ASR R2 ;SHIFT THE PREAMBLE DATA. HAVE ALL THE
3869 ;TRACKS BEEN SKEWED?
3870 033054 103732 BCS 1$ ;BR IF NOT. CONTINUE.
3871
3872 033056 ENDSUB
3873 033056 L10061:
3874 033056 104403 TRAP C$ESUB
3875
3876 033060 BGNSUB
3877 033060 T6.2:
3878 033060 104402 TRAP C$BSUB
3879
3880 ;THIS SUBTEST SENDS AN ALL 1'S PREAMBLE (SKEWED), AN ALL 1'S
3881 ;BYTE 1 DATA (SKEWED) AND AN ALL 0'S BYTE 2 DATA (SKEWED).
3882
3883 033062 005037 002416 CLR DTKIDN ;CLR DEAD TRK IDEN REG.
3884 033066 012702 000377 MOV #377,R2 ;SET UP PREAMBLE DATA. START WITH CHAN 9 SKEWED.
3885 033072 004737 016154 JSR PC,PEINIT ;SET WRAP TASK ADR IN DIABLK & INIT INDEX.
3886 033076 112745 000012 MOV #FC.DAT!FC.VCO,-(R5) ;SET DATA & VCO MODE IN FMT CNTRL.
3887 033102 012703 000777 MOV #777,R3 ;SETUP BYTE 1 DATA=1
3888 033106 010204 MOV R2,R4 ;SETUP BYTE 2 DATA
3889 033110 005104 COM R4 ;=0
3890 033112 042704 177000 BIC #177000,R4 ;CLR GARBAGE BITS
3891 033116 004737 016176 JSR PC,PEDATA ;LOAD THE DATA IN DIABLK
3892 033122 012737 000777 MOV #777,EXDATA ;THE EXPECTED DESKEWED DATA IS 1
3893 033130 005037 002350 CLR EXODTR ;THE EXPECTED '0 OR DEAD' DATA=1 FOR EACH TRACK.
3894 033134 012737 000777 MOV #777,EX1DTR ;THE EXPECTED DESKEWED '1 OR DEAD' DATA=1 FOR ALL TRACKS
3895 033142 004737 015722 JSR PC,DIAEXE ;EXECUTE THE WRAP.
3896
3897 ;CHECK FOR ERRORS
3898
3899 033146 005037 002436 CLR ERRFLG ;INIT ERROR FLAG.
3900 033152 004737 016504 JSR PC,PEERCK ;CHECK FOR ERRORS.
3901 033156 IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
3902 033156 105737 002436 TSTB ERRFLG
3903 033162 001407 BEQ 50170$

```


3904	033164	105737	002212			TSTB	CMPFLG
3905	033170	001404				BEQ	50170\$
3906	033172			ERRHRD	16,SKEWER,SKDAEM		
3907	033172	104456				TRAP	C\$ERHRD
3908	033174	000020				.WORD	16
3909	033176	003270				.WORD	SKEWER
3910	033200	007176				.WORD	SKDAEM
3911	033202			ENDIF			
3912	033202						50170\$:
3913							
3914							
3915							
3916	033202	052702	001000	BIS	#BIT9,R2		;PREPARE TO SHIFT SKEWED TRK.
3917	033206	006202		ASR	R2		;SKEW THE NEXT TRACK. ARE WE DONE YET?
3918	033210	103730		BCS	1\$;BR IF NOT.
3919							
3920	033212			ENDSUB			
3921	033212			L10062:			
3922	033212	104403				TRAP	C\$ESUB
3923							
3924	033214			LET R1 :=	ORORDY OR ORTRAC OR ORDATA		
3925	033214	013701	002372			MOV	ORORDY,R1
3926	033220	053701	002376			BIS	ORTRAC,R1
3927	033224	053701	002400			BIS	ORDATA,R1
3928	033230			LET R1 :=	R1 OR ORTRDD		
3929	033230	053701	002402			BIS	ORTRDD,R1
3930	033234			IF R1 NE	#0 THEN		
3931	033234	005701				TST	R1
3932	033236	001406				BEQ	50171\$
3933	033240			ERRHRD	16,SKEWER		
3934	033240	104456				TRAP	C\$ERHRD
3935	033242	000020				.WORD	16
3936	033244	003270				.WORD	SKEWER
3937	033246	000000				.WORD	0
3938	033250			CALL	MSORT2		
3939	033250	004737	020216			JSR	PC,MSORT2
3940	033254			ENDIF			
3941	033254						50171\$:
3942							
3943							
3944							
3945							
3946	033254			ENDTST			
3947	033254			L10060:			
3948	033254	104401				TRAP	C\$ETST

```

3949          .SBTTL TEST 7: P.E. DEAD TRACK TEST.
3950
3951          ;TEST 7 CHECKS THE DEAD TRACK LOGIC BY RIPPLING A DEAD TRACK THRU
3952          ;A FIELD OF LIVE TRACKS AND ONE LIVE TRACK THRU A FIELD OF DEAD
3953          ;TRACKS.  ADDITIONALLY, EACH SUBTEST WILL SEND 1'S DATA AND 0'S
3954          ;DATA  IN ORDER TO TEST THE 1'S OR DEAD REGISTER AND THE
3955          ;0'S OR DEAD REGISTER.
3956
3957          ;REGISTER USAGE:
3958          ;      R2=PREAMBLE ALL 1'S CHARACTER
3959          ;      R3=1ST DATA BYTE (BITS 0-8)
3960          ;      R4=2ND DATA BYTE (BITS 0-8)
3961
3962          ;      DTKIDN=DEAD TRACK DEFINED IN BITS 0-8 (0=LIVE TRK; 1=DEAD TRK.)
3963
3964 033256      BGNTST
3965 033256      T7::
3966
3967 033256      CALL    SCHEXE                      JSR    PC,SCHEXE
3968 033256 004737 015746
3969 033262      CALL    WAITMT                      JSR    PC,WAITMT
3970 033262 004737 015174
3971 033266      INCR R1 FROM #0 TO #12 BY #2
3972 033266 005001
3973 033270 000402
3974 033272
3975 033272 062701 000002
3976 033276
3977 033276 020127 000012
3978 033302 003003
3979 033304      LET ORDTBL(R1) := #0                ;CLEAR 6 LOC OF ORED WRAP REG TABLES
3980 033304 005061 002372
3981 033310
3982 033310 000770
3983 033312
3984
3985 033312      BGNSUB
3986 033312      T7.1:
3987 033312 104402
3988
3989          ;THIS SUBTEST RIPPLES A DEAD TRACK IN A FIELD OF LIVE TRACKS.
3990
3991 033314 012702 000777      MOV    #777,R2                ;PREAMBLE
3992 033320 012703 000777      MOV    #777,R3                ;BYTE 1 DATA.
3993 033324 005004
3994 033326 012737 000400 002416 1$:  CLR    R4                    ;BYTE 2 DATA.
3995 033334 004737 016154      MOV    #400,DTKIDN            ;SETUP TO SHIFT 1 DEAD TRACK.
3996 033340 112745 000012      JSR    PC,PEINIT             ;SET DATA AND VCO MODE IN FMT CONTROL.
3997 033344 004737 016176      MOVB  #FC.DAT!FC.VCO,-(R5)
3998 033350 004737 015722      JSR    PC,PEDATA             ;LOAD DATA IN DIABLK AND CALCULATE EXPECTED DATA
3999
4000          ;CHECK FOR ERRORS AND REPORT IF ANY.
4001
4002 033354 005037 002436      JSR    PC,DIAEXE            ;EXECUTE IT.
4003 033360 004737 016504      CLR    ERRFLG                ;INIT ERROR FLAG.
4004 033364
4004          JSR    PC,PEERCK            ;CHECK FOR ERRORS.
4004          IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
  
```

```

4005 033364 105737 002436
4006 033370 001407
4007 033372 105737 002212
4008 033376 001404
4009 033400
4010 033400 104456
4011 033402 000021
4012 033404 003310
4013 033406 007176
4014 033410
4015 033410
4016
4017
4018
4019 033410 000241
4020 033412 006237 002416
4021 033416 103346
4022
4023
4024
4025 033420 105703
4026 033422 001404
4027 033424 005003
4028 033426 012704 000777
4029 033432 000735
4030 033434
4031 033434 104432
4032 033436 000002
4033
4034 033440
4035 033440
4036 033440 104403
4037
4038 033442
4039 033442
4040 033442 104402
4041
4042
4043
4044 033444 012702 000777
4045 033450 012703 000777
4046 033454 005004
4047 033456 012737 000377 002416 1$:
4048 033464 004737 016154 2$:
4049 033470 112745 000012
4050 033474 004737 016176
4051 033500 004737 015722
4052
4053
4054
4055 033504 005037 002436
4056 033510 004737 016504
4057 033514
4058 033514 105737 002436
4059 033520 001407
4060 033522 105737 002212

ERRHRD 17,DDER,SKDAEM

ENDIF

50175$:

;SHIFT THE DEAD TRACK AND REPEAT.

CLC ;CLR CARRY TO PREPARE FOR SHIFT.
ASR DTKIDN ;SHIFT DEAD TRACK
BCC 2$ ;BR IF ALL TRACKS NOT KILLED YET.

;CHANGE THE DATA AND REPEAT.

TSTB R3 ;DONE?
BEQ 4$ ;BR IF SO.
CLR R3 ;BYTE 1 DATA.
MOV #777,R4 ;BYTE 2 DATA.
BR 1$ ;REPEAT TEST WITH COMPLEMENTED DATA.
EXIT SUB

4$:

TRAP C$EXIT
.WORD L10064-.

ENDSUB

L10064:

TRAP C$ESUB

BGNSUB

17.2:

TRAP C$BSUB

;THIS SUBTEST RIPPLES A LIVE TRACK IN A FIELD OF DEAD TRACKS.

MOV #777,R2 ;PREAMBLE.
MOV #777,R3 ;BYTE 1 DATA.
CLR R4 ;BYTE 2 DATA.
MOV #377,DTKIDN ;SETUP TO SHIFT 1 LIVE TRACK.
JSR PC,PEINIT ;SET WRAP TSK ADR AND DIABLK INDEX.
MOVB #FC.DAT!FC.VCO,-(R5) ;SET DATA AND VCO MODE IN FMT CNTRL.
JSR PC,PEDATA ;LOAD DIABLD AND GENERATE EXPECTED DATA.
JSR PC,DIAEXE ;DO IT.

;CHECK FOR ERRORS.

CLR ERRFLG ;INIT ERROR FLAG.
JSR PC,PEERCK ;CHECK FOR ERRORS.
IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN

TSTB ERRFLG
BEQ 50176$
TSTB CMPFLG

```

```
4061 033526 001404 ERRHRD 17,DDER,SKDAEM BEQ 50176$
4062 033530
4063 033530 104456 TRAP C$ERHRD
4064 033532 000021 .WORD 17
4065 033534 003310 .WORD DDER
4066 033536 007176 .WORD SKDAEM
4067 033540 ENDIF
4068 033540 50176$:
4069
4070 ;SHIFT THE LIVE TRACK AND REPEAT.
4071
4072 033540 052737 001000 002416 BIS #BIT9,DTKIDN ;PREPARE FOR DATA SHIFT.
4073 033546 006237 002416 ASR DTKIDN ;SHIFT IT. DONE?
4074 033552 103744 BCS 2$ ;BR IF NOT, CONTINUE.
4075
4076 ;CHANGE DATA AND REPEAT
4077
4078 033554 105703 TSTB R3 ;DONE?
4079 033556 001404 BEQ 4$ ;BR IF SO.
4080 033560 005003 CLR R3 ;BYTE 1 DATA.
4081 033562 012704 000777 MOV #777,R4 ;BYTE 2 DATA.
4082 033566 000733 BR 1$ ;CONTINUE.
4083
4084
4085 033570 4$: EXIT SUB TRAP C$EXIT
4086 033570 104432 .WORD L10065-.
4087 033572 000002
4088
4089 033574 ENDSUB
4090 033574 L10065:
4091 033574 104403 TRAP C$ESUB
4092
4093 033576 LET R1 := ORORDY OR ORTRAC OR ORDATA
4094 033576 013701 002372 MOV ORORDY,R1
4095 033602 053701 002376 BIS ORTRAC,R1
4096 033606 053701 002400 BIS ORDATA,R1
4097 033612 LET R1 := R1 OR ORTRDD
4098 033612 053701 002402 BIS ORTRDD,R1
4099 033616 IF R1 NE #0 THEN
4100 033616 005701 TST R1
4101 033620 001406 BEQ 50177$
4102 033622 ERRHRD 17,DDER TRAP C$ERHRD
4103 033622 104456 .WORD 17
4104 033624 000021 .WORD DDER
4105 033626 003310 .WORD 0
4106 033630 000000
4107 033632 CALL MSORT2 JSR PC,MSORT2
4108 033632 004737 020216
4109 033636 ENDIF
4110 033636 50177$:
4111
4112 .EVEN
4113
4114 033636 L10063: ENDTST
4115 033636
4116 033636 104401 TRAP C$ETST
```

4117
 4118
 4119
 4120
 4121
 4122
 4123
 4124
 4125
 4126
 4127
 4128
 4129 033640
 4130 033640
 4131
 4132
 4133
 4134 033640 004737 015746
 4135 033644
 4136 033644 004737 015174
 4137 033650 012737 001777 002420
 4138 033656 004737 016154
 4139 033662 112745 000003
 4140 033666 004737 016722
 4141 033672 004737 015722
 4142
 4143
 4144
 4145 033676 005037 002436
 4146 033702 004737 017022
 4147 033706 004737 017530
 4148
 4149 033712
 4150 033712 105737 002436
 4151 033716 001411
 4152 033720 105237 002437
 4153 033724
 4154 033724 105737 002212
 4155 033730 001404
 4156 033732
 4157 033732 104456
 4158 033734 000022
 4159 033736 003373
 4160 033740 007772
 4161 033742
 4162 033742
 4163 033742
 4164 033742
 4165
 4166
 4167
 4168 033742 005337 002420
 4169 033746 100343
 4170 033750
 4171 033750 105737 002437
 4172 033754 001404

```

.SBTTL TEST 8: LOOKUP TABLE TEST
;THIS TEST VEFIFIES THAT THE CONTENTS OF THE ROM LOOKUP TABLE ARE CORRECT.
;THE ROM CONTENTS IN ADDRESS 0-1777 ARE CHECKED.
:
:      DATA & REGISTER USAGE:
:
:      ROMLKI = ROM LOOKUP TABLE ADDRESS.
:      ERRFLG = ERROR FLAG.
:      R5 = DIABLK INDEX.
:
T8::  BGNTST
:
:OPEN A ROM LOOKUP TABLE LOCATION.
JSR   PC,SCHEXE           ;EXECUTE AN SCH COMMAND.
CALL  WAITMT
:
:JSR   PC,WAITMT
MOV   #1777,ROMLKI       ;INIT THE ROM TABLE ADDRESS
JSR   PC,PEINIT          ;SETUP TASK ADR + READ CNTRL.
MOVB  #FC.VCO!FC.NRZ,-(R5) ;SET VCO + NRZ1 MODE IN FORMAT CNTRL.
JSR   PC,ROMLOK          ;LOAD THE DIABLK.
JSR   PC,DIAEXE          ;EXECUTE IT.
:
:CHECK FOR VALID DATA IN THAT LOCATION.
CLR   ERRFLG             ;INIT ERROR FLAG
JSR   PC,ROMEX           ;GET EXPECTED DATA
JSR   PC,ROMCK           ;GET ACTUAL DATA AND COMPARE WITH
:EXPECTED.
IFB ERRFLG NE #0 THEN
:
:      INCB CTLFLG
:      IFB CMPFLG NE #0 THEN
:
:      ERRHRD 18,ROMER,ROMEM
:
:      TRAP   C$ERHRD
:      .WORD 18
:      .WORD ROMER
:      .WORD ROMEM
:
:      50201$:
:      50200$:
:
:UPDATE THE ADDRESS AND REPEAT THE TEST UNTIL DONE.
DEC   ROMLKI             ;DECR ROM ADDRESS - DONE?
BPL   1$                 ;OR IF NOT.
IFB CTLFLG NE #0 THEN
:
:      TSTB   CTLFLG
:      BEQ   50202$

```

4173	033756		ERRHRD 18,ROMER,ROMLER		
4174	033756	104456			TRAP C\$ERHRD
4175	033760	000022			.WORD 18
4176	033762	003373			.WORD ROMER
4177	033764	014142			.WORD ROMLER
4178	033766		ENDIF		
4179	033766			50202\$:	
4180					
4181					
4182					
4183	033766		EXIT TST		
4184	033766	104432			TRAP C\$EXIT
4185	033770	000002			.WORD L10066-
4186			.EVEN		
4187					
4188	033772		ENDTST		
4189	033772		L10066:		
4190	033772	104401			TRAP C\$ETST

```
4191
4192
4193
4194 033774
4195 033774
4196
4197 033774
4198 033774 004737 015746
4199 034000
4200 034000 004737 015174
4201 034004
4202 034004 012727 000764
4203 034010 000000
4204 034012 013727 002116
4205 034016 000000
4206 034020 005367 177772
4207 034024 001375
4208 034026 005367 177756
4209 034032 001367
4210 034034
4211 034034 004737 020704
4212
4213 034040
4214 034040
4215 034040 104401

.SBTTL TEST 9: IN-LINE MICRO DIAGNOSTIC TEST
T9::
BGNTST
CALL SCHEXE
CALL WAITMT
DELAY 500.
CALL MSORT3
ENDTST
L10067:

JSR PC,SCHEXE
JSR PC,WAITMT
MOV #500.,(PC)+
.WORD 0
MOV L$DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -.4
DEC -22(PC)
BNE -.20
JSR PC,MSORT3
TRAP C$ETST
```

```
4216
4217
4218
4219 034042
4220 034042
4221
4222 034042 005077 146156
4223 034046
4224 034046 004737 014616
4225 034052
4226 034052 004737 015746
4227 034056
4228 034056 004737 020704
4229
4230
4231 034062
4232 034062
4233 034062 104401
4234
4235 034064
4236
```

.SBTTL TEST 10: INIT MICRO DIAGNOSTIC TEST

BGNTST

T10::

CLR @TSSR
CALL WAITSR JSR PC,WAITSR

CALL SCHEXE JSR PC,SCHEXE

CALL MSORT3 JSR PC,MSORT3

.EVEN
ENDTST

L10070:

TRAP C\$ETST

ENDMOD


```

4237      .TITLE PARAMETER CODING
4238
4239      .SBTTL  HARDWARE PARAMETER CODING SECTION
4240
4241      034064      BGNMOD
4242
4243      :++
4244      : THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
4245      : THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
4246      : MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4247      : INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
4248      : MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4249      : WITH THE OPERATOR.
4250      :--
4251
4252      034064      BGNHRD
4253      034064      000024      .WORD  L10071-L$HARD/2
4254      034066      L$HARD::
4255
4256      034066      GPRMA  TS4ADR,0,0,160002,177564,YES      .WORD  T$CODE
4257      034066      000031      .WORD  TS4ADR
4258      034070      034112      .WORD  T$LLOLIM
4259      034072      160002      .WORD  T$HILIM
4260      034074      177564
4261      034076      GPRMD  TS4VCT,2,0,777,60,776,YES      .WORD  T$CODE
4262      034076      001032      .WORD  TS4VCT
4263      034100      034127      .WORD  777
4264      034102      000777      .WORD  T$LLOLIM
4265      034104      000060      .WORD  T$HILIM
4266      034106      000776
4267
4268      034110      EXIT HRD
4269      034110      013004      .WORD  T$CODE
4270
4271      034112      051524  051123  040440  TS4ADR: .NLIST  BEX
4271      034127      126    041505  047524  TS4VCT: .ASCIZ  /TSSR ADDRESS/
4271      .LIST  BEX
4271      .EVEN
4272
4273
4274      034136      ENDHRD
4275
4276      034136      L10071:      .EVEN

```

```

4277          .SBTTL  SOFTWARE PARAMETER CODING SECTION
4278
4279          ;++
4280          ;IT CONTAINS MACRO THAT ARE USED BY THE SUPERVISOR
4281          ;TO BUILD SOFT P-YABLES.  THE MACROS ARE INTERPRETED
4282          ;BY THE SUPERVISOR AS DATA STRUCTURES.  THE MACROS
4283          ;ALLOW THE SUPERVISOR TO COMMUNICATE WITH THE OPERATOR.
4284          ;--
4285
4286          BGNSFT
4287          034136 000034          .WORD L10072-L$SOFT/2
4288          034136
4289          034140
4290          GPRML  ENBMSG,0,1,YES
4291          034140 000130          .WORD  T$CODE
4292          034142 034150          .WORD  ENBMSG
4293          034144 000001          .WORD  1
4294
4295          EXIT SFT
4296          034146 031004          .WORD  T$CODE
4297
4298          034150 047105 041101 042514 ENBMSG: .NLIST BEX
          .ASCIZ /ENABLE DATA COMPARE ERROR PRINTS FOR TESTS 4-7/
          .LIST BEX
4299
4300          034230          .EVEN
4301
4302          034230          ENDSFT
4303
4304          034230          L10072:
4305
4306          ;*****
4307          ;*****
4308          ;      PATCH AREA
4309
4310          034230 000140          PATCH:: .BLKW 96.
4311          ;*****
4312          ;*****
4313          034530          LASTAD
4314
4315          034530 034552          .EVEN
4316          034532 000007          .WORD  T$FREE
4317          034534          .WORD  T$SIZE
4318          L$LAST::
4319          034534          ENDMOD
4320
4321          .SBTTL  HARD CODED P-TABLE
4322
4323          ;++
4324          ;DIAG IS PRE-PARAMETERIZED PER TBL
4325          ;--
4326
4327          034534          BGNSETUP 1
4328          034534          BGNPTAB
4329          034534 000000          .WORD  0
4330          034536 000000          .WORD  0

```

4331 034540 000000
4332 034542 000000
4333 034544 000002
4334 034546
4335 034546 172522
4336 034550 000224
4337 034552
4338 034552
4339 034552
4340
4341 000001

L10073: 172522
224
ENDPTAB
L10075: ENDSETUP
.END

.WORD 0
.WORD 0
.WORD L10075-./2-1

CSGETB= 000026	4#												
CSGETW= 000027	4#												
CSGMAN= 000043	4#												
CSGPHR= 000042	4#	1231	2644										
CSGPLO= 000030	4#												
CSGPRI= 000040	4#												
CSINIT= 000011	4#	2680											
CSINLP= 000020	4#												
CSMANI= 000050	4#												
CSMEM = 000031	4#												
CSMSG = 000023	4#	673	699	725	751	777	809	863	884	912	935	958	982
	1006	1029	1067	1104	1117	1140							
CSOPEN= 000034	4#												
CSPNTB= 000014	4#	660	681	707	733	759	897	920	943	967	991	1014	1038
	1052	1089	1113	1125	1426	2085	2117	2129	2141	2173	2205	2245	2274
	2347	2356	2365	2395	2404	2413	2444	2476	2493	2891	3256		
CSPNTF= 000017	4#	1286	1350	1381	1437	2714	2728						
CSPNTS= 000016	4#												
CSPNTX= 000015	4#	690	716	742	768	786	793	800	818	824	834	844	854
	875	903	926	949	973	997	1020	1095	1131	1208	1340		
CSQIO = 000377	4#												
CSRDBU= 000007	4#												
CSREFG= 000047	4#	2621	2628										
CSRESE= 000033	4#	2878											
CSREVI= 000003	4#	63											
CSRFLA= 000021	4#												
CSRPT = 000025	4#												
CSSEFG= 000046	4#												
CSSPRI= 000041	4#	2779	3235	3264									
CSSEVC= 000037	4#	2664	2699										
CSTPRI= 000013	4#												
C18SDA= 000010 G	362#	1758											
C18SOR= 000004 G	359#	1755											
C18STA= 000007 G	361#	1588	1757										
C18STD= 000011 G	363#	1759											
C18SOD= 000012 G	364#	1760											
C18SD= 000005 G	360#	1756											
DAEM 007406	815	859#											
DAEMA 007461	821	859#											
DAEMB 007530	831	859#											
DAEMC 007617	841	859#											
DAEMD 007701	851	859#											
DASKDD 003336 G	625#												
DATBLD 016126 G	1610#	3623	3656	3691	3732								
DATER 003243 G	625#	3638	3671	3706	3747	3777							
DDER 003310 G	625#	4012	4065	4105									
DESCM 002440 G	625#												
DFPTBL 002204 G	154#												
DIA = 100006 G	343#	512											
DIABLK 002316 G	513	550#	1512	1628									
DIAEXE 015722 G	1530#	3359	3414	3471	3527	3624	3657	3692	3733	3846	3895	3998	4051
	4141												
DIAEXT= 000020 G	342#	515	1628										
DIAGMC= 000000	4												
DIAPKT 002250 G	512#	1532											
DROPU 014570 G	1211	1226#	1342	2716	2730	2894	2971	2987	3048	3061	3101	3122	3134

LOT =	000010	G	248#							
LUNIT	027574		1227	1230	2625*	2632*	2634	2657	2667	2674#
LSACP	002110	G	94#							
LSAPT	002036	G	52#							
LSAU	030112	G	79	2826#						
LSAUT	002070	G	78#							
LSAUTO	027600	G	95	2690#						
LSCCP	002106	G	92#							
LSCLEA	030052	G	93	2772#						
LSCO	002032	G	48#							
LSDEPO	002011	G	30#							
LSDESC	002150	G	85	133#						
LSDESP	002076	G	84#							
LSDEVP	002060	G	70#							
LSDISP	002124	G	55	114#						
LSDLY	002116	G	100#	1264	1307	1399	2611	2955	3032	4204
LSDTP	002040	G	54#							
LSDTYP	002034	G	50#							
LSDU	030074	G	81	2799#						
LSDUT	002072	G	80#							
LSDVTY	002174	G	71	140#						
LSEF	002052	G	65#							
LSETP	002102	G	88#							
LSEXP1	002044	G	58#							
LSEXP2	002046	G	60#							
LSEXP4	002064	G	74#							
LSEXP5	002066	G	76#							
LSHARD	034066	G	37	4253	4254#					
LSHIME	002120	G	102#							
LSHPCP	002016	G	36#							
LSHPTP	002022	G	40#							
LSHW	002204	G	41	152	153#					
LSICP	002104	G	90#							
LSINIT	027316	G	91	2598#						
LSLADP	002026	G	44#							
LSLAST	034534	G	45	4317#	4340					
LSLOAD	002100	G	86#							
LSLUN	002074	G	82#							
LSMREV	002050	G	62#							
LSNAME	002000	G	19#							
LSPRIO	002042	G	56#							
LSPROT	027310	G	97	2583#						
LSPRT	002112	G	96#							
LSREPP	002062	G	72#							
LSREV	002010	G	28#							
LSOFT	034140	G	39	4287	4288#					
LSSPC	002056	G	68#							
LSSPCP	002020	G	38#							
LSPTP	002024	G	42#							
LSSTA	002030	G	46#							
LSW	002212	G	43	170	171#					
LSTEST	002114	G	98#							
LSIML	002014	G	34#							
LSUNIT	002012	G	32#							
L10000	002210		152	160#						
L10001	002214		170	179#						

L10002	006042	669	672#	
L10003	006216	694	698#	
L10004	006300	720	724#	
L10005	006542	746	750#	
L10006	006714	772	776#	
L10007	007174	804	808#	
L10010	007770	858	862#	
L10011	010140	879	883#	
L10012	010330	907	911#	
L10013	010444	930	934#	
L10014	010540	953	957#	
L10015	010704	977	981#	
L10016	011120	1001	1005#	
L10017	011256	1024	1028#	
L10020	011434	1061	1066#	
L10021	014140	1099	1103#	
L10022	014162	1116#		
L10023	014366	1135	1139#	
L10024	014376	1159#		
L10025	014406	1167#		
L10026	014416	1175#		
L10027	014426	1183#		
L10031	027576	2670	2679#	
L10032	027744	2737#		
L10033	030072	2783	2789#	
L10034	030110	2809	2815#	
L10035	030126	2836	2842#	
L10036	030462	2948	2999#	
L10037	030210	2903#		
L10040	030316	2941#		
L10041	030642	3076#		
L10042	030626	3065#		
L10043	031334	3272	3276#	
L10044	030706	3105#		
L10045	031206	3231#		
L10046	032230	3579#		
L10047	031500	3382#		
L10050	031654	3440#		
L10051	032012	3494#		
L10052	032166	3553#		
L10053	032672	3786	3790#	
L10054	032346	3644#		
L10055	032432	3677#		
L10056	032530	3720#		
L10057	032624	3761#		
L10060	033254	3947#		
L10061	033056	3873#		
L10062	033212	3921#		
L10063	033636	4115#		
L10064	033440	4032	4035#	
L10065	033574	4087	4090#	
L10066	033772	4185	4189#	
L10067	034040	4214#		
L10070	034062	4232#		
L10071	034136	4253	4269	4276#
L10072	034230	4287	4296	4304#

Parameter	Code	1	2	3	4	5	6	7	8	9	10	11	12	13
RD.90	= 000006 G	426#												
RFC	002264 G	521#												
ROMCK	017530 G	2028#	4147											
ROMEM	007772 G	866#	4160											
ROMEMA	010036	872	880#											
ROMER	003373 G	625#	4159	4176										
ROMEX	017022 G	1885#	4146											
ROMLER	014142 G	1108#	4177											
ROMLKI	002420 G	601#	871	1816	1824	1891	1904	1906	1910	1923	1929	1935	1937	1945
		1963	2005	4137*	4168*									
ROMLOK	016722 G	1813#	4140											
ROMBLK	= 000013 G	365#	2029											
SCH	= 140004 G	344#	503											
SCHBLK	002276 G	504	530#											
SCHERA	010244	900	908#											
SCHERB	010376	923	931#											
SCHERC	010512	946	954#											
SCHERD	010634	970	978#											
SCHERE	010752	988	1002#											
SCHERF	011036	994	1002#											
SCHERG	011166	1011	1025#											
SCHERH	011210	1017	1025#											
SCHERI	011350	1035	1062#											
SCHERL	014076	1086	1100#											
SCHERM	014121	1092	1100#											
SCHERN	010606	964	978#											
SCHERR	003047 G	625#	3098	3119	3131	3146	3181	3195	3206	3214	3226			
SCHERO	014032 G	1084#	3227											
SCHER1	010200 G	892#	3099											
SCHER2	010332 G	915#	3120											
SCHER3	010446 G	938#	3132	3147										
SCHER4	010542 G	962#	3182											
SCHER5	010706 G	986#	3196											
SCHER6	011122 G	1009#	3207											
SCHER7	011260 G	1033#	3215											
SCHEXE	015746 G	1540#	3112	3327	3597	3809	3968	4134	4198	4226				
SCHEXT	= 000010 G	340#	506											
SCHPKT	002240 G	503#	1542	3239*	3241	3261*								
SFPTBL	002212 G	172#												
SHWRAP	016020 G	1571#	2939	3068										
SIP	= 040000 G	347#	3201											
SKDAEM	007176 G	812#	3639	3672	3707	3748	3861	3910	4013	4066				
SKWER	003270 G	625#	3860	3909	3936									
SPECON	003422 G	625#												
SSROFF	002566 G	625#	1200	1332	2725									
SSRON	002623 G	625#	2885											
SVCGBL	= 000000	4#	19	20	28	29	30	31	32	33	34	35	36	37
		38	39	40	41	42	43	44	45	46	47	48	49	50
		51	52	53	54	55	56	57	58	59	60	61	62	63
		65	66	68	69	70	71	72	73	74	75	76	77	78
		79	80	81	82	83	84	85	86	87	88	89	90	91
		92	93	94	95	96	97	98	99	100	101	102	103	114
		115	133	134	140	141	153	154	155	171	172	173	649	650
		676	677	702	703	728	729	754	755	780	781	812	813	866
		867	892	893	915	916	938	939	962	963	986	987	1009	1010
		1033	1034	1084	1085	1108	1109	1120	1121	1155	1156	1163	1164	1171

SVCINS= 000001

1172	1179	1180	2583	2584	2598	2599	2690	2691	2772	2773	2799	2800
2826	2827	4254	4255	4288	4289	4317#	4318					
4#	20	21	22	23	24	25	26	27	29	31	33	35
37	39	41	43	45	47	49	51	53	55	57	59	61
63	64	66	67	69	71	73	75	77	79	81	83	85
87	89	91	93	95	97	99	101	103	113	115	116	117
118	119	120	121	122	123	124	134	138	141	142	152	170
657	658	659	660	661	668	669	673	678	679	680	681	682
684	685	686	687	688	689	690	691	693	694	699	704	705
706	707	708	710	711	712	713	714	715	716	717	719	720
725	730	731	732	733	734	736	737	738	739	740	741	742
743	745	746	751	756	757	758	759	760	762	763	764	765
766	767	768	769	771	772	777	782	783	784	785	786	787
789	790	791	792	793	794	796	797	798	799	800	801	803
804	809	814	815	816	817	818	819	821	822	823	824	825
827	828	829	830	831	832	833	834	835	837	838	839	840
841	842	843	844	845	847	848	849	850	851	852	853	854
855	857	858	863	868	869	870	871	872	873	874	875	876
878	879	884	894	895	896	897	898	900	901	902	903	904
906	907	912	917	918	919	920	921	923	924	925	926	927
929	930	935	940	941	942	943	944	946	947	948	949	950
952	953	958	964	965	966	967	968	970	971	972	973	974
976	977	982	988	989	990	991	992	994	995	996	997	998
1000	1001	1006	1011	1012	1013	1014	1015	1017	1018	1019	1020	1021
1023	1024	1029	1035	1036	1037	1038	1039	1049	1050	1051	1052	1053
1060	1061	1067	1086	1087	1088	1089	1090	1092	1093	1094	1095	1096
1098	1099	1104	1110	1111	1112	1113	1114	1117	1122	1123	1124	1125
1126	1128	1129	1130	1131	1132	1134	1135	1140	1160	1168	1176	1184
1190	1198	1199	1200	1201	1203	1204	1205	1206	1207	1208	1209	1227
1228	1230	1231	1232	1234	1236	1262	1263	1264	1265	1266	1267	1268
1269	1271	1283	1284	1285	1286	1287	1305	1306	1307	1308	1309	1310
1311	1312	1314	1330	1331	1332	1333	1335	1336	1337	1338	1339	1340
1341	1347	1348	1349	1350	1351	1378	1379	1380	1381	1382	1397	1398
1399	1400	1401	1402	1403	1404	1406	1423	1424	1425	1426	1427	1429
1434	1435	1436	1437	1438	2082	2083	2084	2085	2086	2114	2115	2116
2117	2118	2126	2127	2128	2129	2130	2138	2139	2140	2141	2142	2170
2171	2172	2173	2174	2202	22	2204	2205	2206	2242	2243	2244	2245
2246	2271	2272	2273	2274	2275	2344	2345	2346	2347	2348	2353	2354
2355	2356	2357	2362	2363	2364	2365	2366	2392	2393	2394	2395	2396
2401	2402	2403	2404	2405	2410	2411	2412	2413	2414	2428	2429	2430
2431	2441	2442	2443	2444	2445	2473	2474	2475	2476	2477	2489	2490
2491	2492	2493	2494	2604	2605	2606	2607	2609	2610	2611	2612	2613
2614	2615	2616	2620	2621	2623	2627	2628	2630	2639	2644	2646	2660
2661	2662	2663	2664	2665	2669	2670	2680	2695	2696	2697	2698	2699
2700	2704	2705	2710	2711	2712	2713	2714	2715	2724	2725	2726	2727
2728	2729	2738	2775	2776	2778	2779	2782	2783	2790	2808	2809	2816
2835	2836	2843	2874	2876	2878	2883	2884	2885	2886	2888	2889	2890
2891	2892	2899	2904	2922	2924	2934	2942	2947	2948	2953	2954	2955
2956	2957	2958	2959	2960	2966	2967	2968	2969	2982	2983	2984	2985
3000	3027	3030	3031	3032	3033	3034	3035	3036	3037	3043	3044	3045
3046	3056	3057	3058	3059	3066	3077	3088	3096	3097	3098	3099	3106
3110	3117	3118	3119	3120	3129	3130	3131	3132	3144	3145	3146	3147
3179	3180	3181	3182	3193	3194	3195	3196	3204	3205	3206	3207	3212
3213	3214	3215	3224	3225	3226	3227	3232	3234	3235	3248	3249	3250
3251	3253	3254	3255	3256	3257	3263	3264	3271	3272	3277	3342	3374
3375	3376	3377	3383	3394	3432	3433	3434	3435	3441	3452	3486	3487

PARAMETER CODING
CZTSIB.P11

MACY11 30(1046)
19-AUG-79 20:26

19-AUG-79 20:27 PAGE 126
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0154

T\$GMAN= 000000
T\$HILI= 000776
T\$LAST= 000001
T\$LOLI= 000060
T\$LSYM= 010000

4#													
4257#	4260	4262#	4266										
4#	4315#	4328											
4257#	4259	4262#	4265										
4#	161	180	673	699	725	751	777	809	863	884	912	935	
958	982	1006	1029	1067	1104	1117	1140	1160	1168	1176	1184	2680	
2738	2790	2816	2843	2904	2942	3000	3066	3077	3106	3232	3277	3383	
3441	3495	3554	3580	3645	3678	3721	3762	3791	3874	3922	3948	4036	
4091	4116	4190	4215	4233	4277	4305							

T\$LTNO= 000012
T\$NEST= 177777

4318#													
4#	9#	152#	160#	170#	179#	182#	186#	649#	672#	676#	698#	702#	
724#	728#	750#	754#	776#	780#	808#	812#	862#	866#	883#	892#	911#	
915#	934#	938#	957#	962#	981#	986#	1005#	1009#	1028#	1033#	1066#	1084#	
1103#	1108#	1116#	1120#	1139#	1155#	1159#	1163#	1167#	1171#	1175#	1179#	1183#	
2574#	2583#	2588#	2592#	2598#	2679#	2690#	2737#	2772#	2789#	2799#	2815#	2826#	
2842#	2846#	2851#	2871#	2874#	2876#	2898#	2903#	2922#	2924#	2933#	2941#	2999#	
3017#	3027#	3065#	3076#	3085#	3088#	3105#	3110#	3231#	3276#	3325#	3342#	3382#	
3394#	3440#	3452#	3494#	3507#	3553#	3579#	3595#	3616#	3644#	3649#	3677#	3682#	
3720#	3725#	3761#	3790#	3807#	3828#	3873#	3878#	3921#	3947#	3966#	3987#	4035#	
4040#	4090#	4115#	4131#	4189#	4196#	4214#	4221#	4232#	4236#	4242#	4253#	4269	
4275#	4287#	4296	4303#	4320#									

T\$NS0 = 000000
T\$NS1 = 000005

9#	182	186#	2574	2583#	2588	2592#	2846	2851#	4236	4242#	4320		
152#	160	170#	179	649#	672	676#	698	702#	724	728#	750	754#	
776	780#	808	812#	862	866#	883	892#	911	915#	934	938#	957	
962#	981	986#	1005	1009#	1028	1033#	1066	1084#	1103	1108#	1116	1120#	
1139	1155#	1159	1163#	1167	1171#	1175	1179#	1183	2598#	2679	2690#	2737	
2772#	2789	2799#	2815	2826#	2842	2871#	2999	3017#	3076	3085#	3276	3325#	
3579	3595#	3790	3807#	3947	3966#	4115	4131#	4189	4196#	4214	4221#	4232	

T\$NS2 = 000002

4253#	4269	4275	4287#	4296	4303								
2874#	2903	2922#	2941	3027#	3065	3088#	3105	3110#	3231	3342#	3382	3394#	
3440	3452#	3494	3507#	3553	3616#	3644	3649#	3677	3682#	3720	3725#	3761	
3828#	3873	3878#	3921	3987#	4035	4040#	4090						

T\$NS3 = 000003
T\$PCNT= 000000
T\$PTAB= 010074
T\$PTHV= 000001
T\$PTNU= 000001
T\$SAVL= 177777
T\$SEGL= 177777
T\$SEKO= 010000
T\$SIZE= 000007
T\$SUBN= 000000

2876#	2898	2924#	2933										
4328#	4329#												
4329#	4335												
33	4340#												
4#	4335#	4340											
4#													
4#	2876#	2898#	2900	2924#	2933#	2935							
2876#	2898	2924#	2933										
4316	4340#												
4#	2870#	2873#	2921#	3016#	3026#	3084#	3087#	3109#	3324#	3341#	3393#	3451#	
3506#	3594#	3615#	3648#	3681#	3724#	3806#	3827#	3877#	3965#	3986#	4039#	4130#	
4195#	4220#												

T\$TAGL= 177777
T\$TAGN= 010076

4#													
4#	152#	170#	649#	676#	702#	728#	754#	780#	812#	866#	892#	915#	
938#	962#	986#	1009#	1033#	1084#	1108#	1120#	1155#	1163#	1171#	1179#	2583#	
2598#	2690#	2772#	2799#	2826#	2871#	2874#	2922#	3017#	3027#	3085#	3088#	3110#	
3325#	3342#	3394#	3452#	3507#	3595#	3616#	3649#	3682#	3725#	3807#	3828#	3878#	
3966#	3987#	4040#	4131#	4196#	4221#	4253#	4287#	4328#	4329#	4333#			
4#													
115#	116#	117#	118#	119#	120#	121#	122#	123#	124#	125#	160#	179#	
182#	668#	669	672#	693#	694	698#	719#	720	724#	745#	746	750#	
771#	772	776#	803#	804	808#	857#	858	862#	878#	879	883#	906#	
907	911#	929#	930	934#	952#	953	957#	976#	977	981#	1000#	1001	
1005#	1023#	1024	1028#	1060#	1061	1066#	1098#	1099	1103#	1116#	1134#	1135	
1139#	1159#	1167#	1175#	1183#	2574#	2588#	2669#	2670	2679#	2737#	2782#	2783	

T\$TEMP= 000000

1788	1794	1795	1796	1797	2029	2030	2061	2062	2065	2066	2067	2070
2071	2072	2075	2076	2077	2079	2091	2092	2095	2096	2097	2100	2101
2102	2105	2106	2107	2109	2110	2111	2123	2135	2154	2157	2158	2159
2160	2163	2164	2165	2168	2176	2178	2179	2182	2187	2188	2189	2190
2191	2194	2196	2197	2200	2218	2221	2222	2225	2226	2227	2230	2231
2232	2235	2236	2237	2239	2265	2266	2267	2268	2288	2291	2296	2297
2303	2306	2309	2310	2316	2317	2422	2423	2424	2425	2433	2435	2436
2437	2438	2450	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462
2463	2464	2466	2467	2470	2479	2480	2481	2482	2625	2626	2632	2633
2634	2635	2636	2657	2658	2659	2667	2668	2693	2694	2702	2703	2707
2721	2758	2759	2880	2894	2901	2927	2937	2939	2963	2971	2979	2987
3022	3040	3048	3053	3061	3068	3093	3101	3112	3114	3122	3126	3134
3138	3149	3153	3154	3155	3156	3160	3187	3201	3237	3238	3239	3240
3241	3242	3243	3245	3261	3262	3267	3268	3269	3327	3329	3367	3368
3369	3422	3424	3425	3426	3427	3479	3480	3481	3535	3537	3538	3539
3540	3558	3559	3560	3561	3562	3563	3564	3572	3597	3599	3601	3602
3604	3605	3608	3609	3610	3631	3664	3699	3740	3766	3767	3768	3769
3770	3771	3772	3780	3809	3811	3813	3814	3816	3817	3820	3821	3822
3853	3902	3925	3926	3927	3928	3929	3930	3931	3939	3968	3970	3972
3973	3975	3976	3979	3980	3981	4005	4058	4094	4095	4096	4097	4098
4099	4100	4108	4136	4150	4154	4171	4198	4200	4211	4224	4226	4228
4#	1255	1260	1273	1280	1298	1303	1316	1323	1344	1353	1355	1375
1408	1413	1420	1431	1440	1442	1788	1792	2079	2088	2111	2120	2123
2132	2135	2144	2148	2150	2152	2154	2160	2184	2208	2210	2212	2218
2239	2248	2253	2255	2259	2268	2277	2288	2291	2293	2298	2300	2303
2306	2311	2313	2318	2320	2425	2438	2447	2450	2486	2496	2498	2500
2636	2641	2707	2718	2721	2732	2734	2880	2896	2927	2931	2963	2973
2979	2989	3040	3050	3053	3063	3093	3103	3114	3124	3126	3136	3138
3140	3151	3187	3189	3198	3201	3209	3217	3219	3245	3259	3369	3371
3379	3427	3429	3437	3481	3483	3491	3540	3542	3550	3564	3574	3631
3633	3641	3664	3666	3674	3699	3701	3709	3740	3742	3750	3772	3782
3853	3855	3863	3902	3904	3912	3931	3941	4005	4007	4015	4058	4060
4068	4100	4110	4150	4154	4162	4164	4171	4179				
4#	1247	1275	1290	1318	1385	1415	1741	1751	3601	3611	3813	3823
3972	3982											
4#												
4#	1246	1278	1289	1321	1384	1418						
4#	652	655	656	663	664	1041	1042	1044	1047	1048	1055	1056
1157	1158	1165	1166	1173	1174	1181	1182	1255	1259	1280	1298	1302
1323	1326	1327	1328	1329	1375	1393	1394	1395	1396	1408	1412	1420
1552	1553	1555	1558	1559	1560	1561	1590	1591	1631	1632	1749	1750
1762	1763	1782	1787	1788	1794	1795	1796	1797	2029	2030	2061	2065
2066	2070	2071	2075	2076	2077	2079	2091	2095	2096	2100	2101	2105
2106	2107	2109	2110	2111	2123	2135	2154	2157	2159	2160	2163	2164
2165	2168	2176	2178	2179	2182	2187	2188	2189	2190	2191	2194	2196
2200	2218	2221	2225	2226	2230	2231	2235	2236	2237	2239	2265	2267
2268	2288	2291	2293	2296	2297	2303	2306	2309	2310	2316	2317	2422
2424	2425	2433	2435	2436	2437	2438	2450	2453	2454	2455	2456	2457
2458	2459	2461	2462	2463	2464	2466	2467	2470	2479	2480	2481	2482
2625	2626	2632	2633	2634	2635	2636	2657	2659	2667	2668	2693	2694
2702	2703	2721	2758	2759	2880	2927	2963	2979	3093	3114	3126	3138
3140	3153	3155	3156	3160	3187	3189	3201	3237	3238	3239	3240	3241
3242	3261	3262	3267	3269	3367	3368	3422	3424	3425	3426	3479	3480
3535	3537	3538	3539	3558	3560	3561	3562	3563	3564	3609	3610	3766
3768	3769	3770	3771	3772	3821	3822	3925	3927	3928	3929	3930	3931
3980	3981	4094	4096	4097	4098	4099	4100					

\$F\$IF = 000110

\$F\$INC = 000210

\$F\$L00 = 000200

\$F\$NAM = 000160

\$F\$NO = 000403

PARAMETER CODING
CZTSIB.P11

MACY11 30(1046)
19-AUG-79 20:26

19-AUG-79 20:27 PAGE 132
CROSS REFERENCE TABLE -- USER SYMBOLS

E 13

SEQ 0160

\$ISK1 = 000001
\$ISK2 = 000001
\$ISK3 = 000001
\$LOCTA= 177777

1298#	1316	1323#	1353	1408#	1413	1420#	1440	2111#	2152	2154#	2210	227#
2253	2291#	2298	2303#	2318	2438#	2498	2721#	2732	3201#	3217	4154#	4.62
2123#	2150	2160#	2208	2306#	2311	2450#	2496					
2135#	2148											
4#	651	653	665	666	1043	1045	1057	1058	1248	1249	1251	1253
1256	1258	1260	1273	1275	1276	1278	1281	1291	1292	1294	1296	1299
1301	1303	1316	1318	1319	1321	1324	1344	1345	1353	1355	1376	1386
1387	1389	1391	1409	1411	1413	1415	1416	1418	1421	1431	1432	1440
1442	1554	1556	1562	1563	1742	1743	1745	1747	1751	1752	1789	1792
2080	2088	2089	2112	2120	2121	2124	2132	2133	2136	2144	2145	2148
2150	2152	2155	2161	2184	2185	2208	2210	2212	2219	2240	2248	2249
2253	2255	2256	2259	2269	2277	2289	2292	2294	2298	2300	2301	2304
2307	2311	2313	2314	2318	2320	2327	2329	2330	2332	2336	2337	2338
2339	2340	2342	2350	2351	2359	2360	2368	2375	2377	2378	2380	2384
2385	2386	2387	2388	2390	2398	2399	2407	2408	2416	2426	2439	2447
2448	2451	2471	2484	2486	2487	2496	2498	2500	2637	2641	2708	2718
2719	2722	2732	2734	2881	2896	2928	2931	2964	2973	2980	2989	3041
3050	3054	3063	3094	3103	3115	3124	3127	3136	3139	3141	3142	3151
3162	3164	3165	3167	3171	3172	3173	3174	3175	3177	3184	3185	3188
3190	3191	3198	3199	3202	3209	3210	3217	3219	3221	3222	3229	3246
3259	3370	3372	3379	3428	3430	3437	3482	3484	3491	3541	3543	3550
3565	3574	3602	3603	3605	3607	3611	3612	3632	3634	3641	3665	3667
3674	3700	3702	3709	3741	3743	3750	3773	3782	3814	3815	3817	3819
3823	3824	3854	3856	3863	3903	3905	3912	3932	3941	3973	3974	3976
3978	3982	3983	4006	4008	4015	4059	4061	4068	4101	4110	4151	4155
4162	4164	4172	4179									

\$LSTCN= 177777

4#	651	652	654	665	666	1043	1044	1046	1057	1058	1246	1247
1249	1250	1251	1254	1255	1257	1260	1261	1273	1275	1276	1278	1280
1282	1289	1290	1292	1293	1294	1297	1298	1300	1303	1304	1316	1318
1319	1321	1323	1325	1345	1346	1353	1355	1375	1377	1384	1385	1387
1388	1389	1392	1408	1410	1413	1415	1416	1418	1420	1422	1432	1433
1440	1442	1554	1555	1557	1562	1563	1741	1743	1744	1745	1748	1751
1752	1788	1790	1792	2079	2081	2089	2090	2111	2113	2121	2122	2123
2125	2133	2134	2135	2137	2145	2146	2148	2150	2152	2154	2156	2160
2162	2185	2186	2208	2210	2212	2218	2220	2239	2241	2249	2250	2253
2256	2257	2259	2268	2270	2277	2288	2290	2291	2295	2298	2301	2302
2303	2305	2306	2308	2311	2314	2315	2318	2320	2326	2328	2330	2333
2337	2338	2339	2340	2368	2369	2374	2376	2378	2381	2385	2386	2387
2388	2416	2417	2425	2427	2438	2440	2448	2449	2450	2452	2471	2472
2483	2487	2488	2496	2498	2500	2636	2638	2641	2707	2709	2719	2720
2721	2723	2732	2734	2880	2882	2896	2927	2929	2931	2963	2965	2973
2979	2981	2989	3040	3042	3050	3053	3055	3063	3093	3095	3103	3114
3116	3124	3126	3128	3136	3138	3143	3151	3161	3163	3165	3168	3172
3173	3174	3175	3187	3192	3199	3200	3201	3203	3210	3211	3217	3219
3229	3230	3245	3247	3259	3369	3373	3379	3427	3431	3437	3481	3485
3491	3540	3544	3550	3564	3566	3574	3601	3603	3604	3605	3608	3611
3612	3631	3635	3641	3664	3668	3674	3699	3703	3709	3740	3744	3750
3772	3774	3782	3813	3815	3816	3817	3820	3823	3824	3853	3857	3863
3902	3906	3912	3931	3933	3941	3972	3974	3975	3976	3979	3982	3983
4005	4009	4015	4058	4062	4068	4100	4102	4110	4150	4152	4154	4156
4162	4164	4171	4173	4179								

\$LSTIN= 000001

4#	652	653	655	663	665	1041	1044	1045	1047	1055	1057	1157
1165	1173	1181	1211	1247	1248	1250	1252	1253	1255	1256	1258	1275
1280	1281	1290	1291	1293	1295	1296	1298	1299	1301	1318	1323	1324
1326	1328	1344	1375	1376	1385	1386	1388	1390	1391	1393	1395	1408
1409	1411	1415	1420	1421	1431	1552	1555	1556	1558	1560	1562	1590

PARAMETER CODING
CZTSIB.P11

MACY11 30(1046)
19-AUG-79 20:26

19-AUG-79 20:27 PAGE 133
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0161

1631	1741	1742	1744	1746	1747	1749	1751	1762	1782	1783	1784	1785	
1786	1788	1789	1794	1796	2029	2061	2062	2063	2064	2066	2067	2068	
2069	2071	2072	2073	2074	2076	2079	2080	2088	2091	2092	2093	2094	
2096	2097	2098	2099	2101	2102	2103	2104	2106	2109	2111	2112	2120	
2123	2124	2132	2135	2136	2144	2154	2155	2157	2158	2160	2161	2163	
2165	2166	2167	2176	2177	2179	2180	2181	2184	2187	2189	2191	2192	
2193	2196	2197	2198	2199	2218	2219	2221	2222	2223	2224	2226	2227	
2228	2229	2231	2232	2233	2234	2236	2239	2240	2243	2255	2265	2266	
2268	2269	2288	2289	2291	2292	2293	2294	2296	2300	2303	2304	2306	
2307	2309	2313	2316	2326	2327	2328	2329	2331	2333	2334	2335	2337	
2338	2339	2340	2350	2359	2374	2375	2376	2377	2379	2381	2382	2383	
2385	2386	2387	2388	2398	2407	2422	2423	2425	2426	2433	2434	2436	
2438	2439	2447	2450	2451	2453	2455	2457	2459	2460	2462	2464	2465	
2467	2468	2469	2479	2481	2483	2484	2486	2625	2632	2634	2636	2637	
2657	2658	2667	2693	2702	2707	2708	2718	2721	2722	2758	2880	2881	
2894	2901	2927	2928	2937	2939	2963	2964	2971	2979	2980	2987	3022	
3040	3041	3048	3053	3054	3061	3068	3093	3094	3101	3112	3114	3115	
3122	3126	3127	3134	3138	3139	3140	3141	3149	3153	3154	3156	3157	
3158	3159	3161	3162	3163	3164	3166	3168	3169	3170	3172	3173	3174	
3175	3184	3187	3188	3189	3190	3198	3201	3202	3209	3221	3237	3239	
3241	3243	3245	3246	3261	3267	3268	3327	3329	3367	3369	3370	3371	
3372	3422	3423	3425	3427	3428	3429	3430	3479	3481	3482	3483	3484	
3535	3536	3538	3540	3541	3542	3543	3558	3559	3560	3562	3564	3565	
3572	3597	3599	3601	3602	3604	3606	3607	3609	3611	3631	3632	3633	
3634	3664	3665	3666	3667	3699	3700	3701	3702	3740	3741	3742	3743	
3766	3767	3768	3770	3772	3773	3780	3809	3811	3813	3814	3816	3818	
3819	3821	3823	3853	3854	3855	3856	3902	3903	3904	3905	3925	3926	
3927	3929	3931	3932	3939	3968	3970	3972	3973	3975	3977	3978	3980	
3982	4005	4006	4007	4008	4058	4059	4060	4061	4094	4095	4096	4098	
4100	4101	4108	4136	4150	4151	4154	4155	4171	4172	4198	4200	4211	
4224	4226	4228											
	4#	651	652	654	665	666	1043	1044	1046	1057	1058	1246	1247
1249	1250	1251	1254	1255	1257	1260	1261	1273	1275	1276	1278	1280	
1282	1289	1290	1292	1293	1294	1297	1298	1300	1303	1304	1316	1318	
1319	1321	1323	1325	1344	1345	1346	1353	1355	1375	1377	1384	1385	
1387	1388	1389	1392	1408	1410	1413	1415	1416	1418	1420	1422	1431	
1432	1433	1440	1442	1554	1555	1557	1562	1563	1741	1743	1744	1745	
1748	1751	1752	1788	1790	1792	2079	2081	2088	2089	2090	2111	2113	
2120	2121	2122	2123	2125	2132	2133	2134	2135	2137	2144	2145	2146	
2148	2150	2152	2154	2156	2160	2162	2184	2185	2186	2208	2210	2212	
2218	2220	2239	2241	2248	2249	2250	2253	2255	2256	2257	2259	2268	
2270	2277	2288	2290	2291	2295	2298	2300	2301	2302	2303	2305	2306	
2308	2311	2313	2314	2315	2318	2320	2326	2328	2330	2337	2342	2343	
2350	2351	2352	2359	2360	2361	2368	2369	2374	2376	2378	2385	2390	
2391	2398	2399	2400	2407	2408	2409	2416	2417	2425	2427	2438	2440	
2447	2448	2449	2450	2452	2471	2472	2483	2486	2487	2488	2496	2498	
2500	2636	2638	2641	2707	2709	2718	2719	2720	2721	2723	2732	2734	
2880	2882	2896	2927	2929	2931	2963	2965	2973	2979	2981	2989	3040	
3042	3050	3053	3055	3063	3093	3095	3103	3114	3116	3124	3126	3128	
3136	3138	3143	3151	3161	3163	3165	3172	3177	3178	3184	3185	3186	
3187	3192	3198	3199	3200	3201	3203	3209	3210	3211	3217	3219	3221	
3222	3223	3229	3230	3245	3247	3259	3369	3373	3379	3427	3431	3437	
3481	3485	3491	3540	3544	3550	3564	3566	3574	3601	3603	3604	3605	
3608	3611	3612	3631	3635	3641	3664	3668	3674	3699	3703	3709	3740	
3744	3750	3772	3774	3782	3813	3815	3816	3817	3820	3823	3824	3853	
3857	3863	3902	3906	3912	3931	3933	3941	3972	3974	3975	3976	3979	

\$LSTST= 177777

	3982	3983	4005	4009	4015	4058	4062	4068	4100	4102	4110	4150	4152
	4154	4156	4162	4164	4171	4173	4179						
\$LSITA= 000001	4#	651	666	1043	1058	1249	1251	1260	1273	1276	1278	1292	1294
	1303	1316	1319	1321	1345	1353	1355	1387	1389	1413	1416	1418	1432
	1440	1442	1554	1563	1743	1745	1752	1792	2089	2121	2133	2145	2148
	2150	2152	2185	2208	2210	2212	2249	2253	2256	2259	2277	2298	2301
	2311	2314	2318	2320	2330	2332	2336	2342	2351	2360	2368	2378	2380
	2384	2390	2399	2408	2416	2448	2471	2487	2496	2498	2500	2641	2719
	2732	2734	2896	2931	2973	2989	3050	3063	3103	3124	3136	3142	3151
	3165	3167	3171	3177	3185	3191	3199	3210	3217	3219	3222	3229	3259
	3379	3437	3491	3550	3574	3603	3605	3612	3641	3674	3709	3750	3782
	3815	3817	3824	3863	3912	3941	3974	3976	3983	4015	4068	4110	4162
	4164	4179											
\$MCALL= 000000	2#	4											
\$NESTL= 177777	4#	651#	665#	1043#	1057#	1246#	1247#	1255#	1260	1273#	1275#	1278#	1280#
	1289#	1290#	1298#	1303	1316#	1318#	1321#	1323#	1344	1353#	1355#	1375#	1384#
	1385#	1408#	1413#	1415#	1418#	1420#	1431	1440#	1442#	1554#	1562#	1741#	1751#
	1788#	1792#	2079#	2088	2111#	2120	2123#	2132	2135#	2144	2148#	2150#	2152#
	2154#	2160#	2184	2208#	2210#	2212#	2218#	2239#	2248	2253#	2255	2259#	2268#
	2277#	2288#	2291#	2298#	2300	2303#	2306#	2311#	2313	2318#	2320#	2326#	2342
	2350	2359	2368#	2374#	2390	2398	2407	2416#	2425#	2438#	2447	2450#	2471#
	2483#	2486	2496#	2498#	2500#	2636#	2641#	2707#	2718	2721#	2732#	2734#	2880#
	2896#	2927#	2931#	2963#	2973#	2979#	2989#	3040#	3050#	3053#	3063#	3093#	3103#
	3114#	3124#	3126#	3136#	3138#	3151#	3161#	3177	3184	3187#	3198	3201#	3209
	3217#	3219#	3221	3229#	3245#	3259#	3369#	3379#	3427#	3437#	3481#	3491#	3540#
	3550#	3564#	3574#	3601#	3611#	3631#	3641#	3664#	3674#	3699#	3709#	3740#	3750#
	3772#	3782#	3813#	3823#	3853#	3863#	3902#	3912#	3931#	3941#	3972#	3982#	4005#
	4015#	4058#	4068#	4100#	4110#	4150#	4154#	4162#	4164#	4171#	4179#		
\$NSKO = 000110	651#	665	1043#	1057	1246#	1278	1280#	1355	1375#	1442	1554#	1562	1741#
	1751	1788#	1792	2079#	2088	2212	2218#	2255	2259	2268#	2277	2288#	2300
	2320	2326#	2342	2350	2359	2368	2374#	2390	2398	2407	2416	2425#	2500
	2636#	2641	2707#	2718	2734	2880#	2896	2927#	2931	2963#	2973	2979#	2989
	3040#	3050	3053#	3063	3093#	3103	3114#	3124	3126#	3136	3138#	3151	3161#
	3177	3184	3221	3229	3245#	3259	3369#	3379	3427#	3437	3481#	3491	3540#
	3550	3564#	3574	3601#	3611	3631#	3641	3664#	3674	3699#	3709	3740#	3750
	3772#	3782	3813#	3823	3853#	3863	3902#	3912	3931#	3941	3972#	3982	4005#
	4015	4058#	4068	4100#	4110	4150#	4164	4171#	4179				
\$NSK1 = 000110	1247#	1275	1289#	1321	1323#	1344	1353	1384#	1418	1420#	1431	1440	2111#
	2120	2152	2154#	2210	2239#	2248	2253	2291#	2298	2303#	2313	2318	2438#
	2447	2498	2721#	2732	3187#	3198	3219	4154#	4162				
\$NSK2 = 000110	1255#	1260	1273	1290#	1318	1385#	1415	2123#	2132	2150	2160#	2184	2208
	2306#	2311	2450#	2486	2496	3201#	3209	3217					
\$NSK3 = 000130	1298#	1303	1316	1408#	1413	2135#	2144	2148	2471#	2483			
\$SAVLE= 177777	4#	665#	666#	1057#	1058#	1250#	1254#	1293#	1297#	1388#	1392#	1562#	1563#
	1744#	1748#	2328#	2330#	2376#	2378#	3163#	3165#	3604#	3608#	3816#	3820#	3975#
	3979#												
\$SSKO = 050173	665#	666	1057#	1058	1250#	1254	1293#	1297	1388#	1392	1562#	1563	1744#
	1748	2328#	2330	2376#	2378	3163#	3165	3604#	3608	3816#	3820	3975#	3979
\$TAGLE= 177777	4#	652#	654#	665#	1044#	1046#	1057#	1249#	1251#	1254#	1257#	1260#	1261#
	1273#	1275#	1276#	1282#	1292#	1294#	1297#	1300#	1303#	1304#	1316#	1318#	1319#
	1325#	1345#	1346#	1353#	1355#	1377#	1387#	1389#	1392#	1410#	1413#	1415#	1416#
	1422#	1432#	1433#	1440#	1442#	1555#	1557#	1562#	1743#	1745#	1748#	1751#	1752#
	1790#	1792#	2081#	2089#	2090#	2113#	2121#	2122#	2125#	2133#	2134#	2137#	2145#
	2146#	2148#	2150#	2152#	2156#	2162#	2185#	2186#	2208#	2210#	2212#	2220#	2241#
	2249#	2250#	2253#	2256#	2257#	2259#	2270#	2277#	2290#	2295#	2298#	2301#	2302#
	2305#	2308#	2311#	2314#	2315#	2318#	2320#	2337#	2342	2350	2351	2359	2360

2368#	2369#	2385#	2390	2398	2399	2407	2408	2416#	2417#	2427#	2440#	2448#
2449#	2452#	2472#	2483#	2487#	2488#	2496#	2498#	2500#	2638#	2641#	2709#	2719#
2720#	2723#	2732#	2734#	2882#	2896#	2929#	2931#	2965#	2973#	2981#	2989#	3042#
3050#	3055#	3063#	3095#	3103#	3116#	3124#	3128#	3136#	3143#	3151#	3172#	3177
3184	3185	3192#	3199#	3200#	3203#	3210#	3211#	3217#	3219#	3221	3222	3229#
3230#	3247#	3259#	3373#	3379#	3431#	3437#	3485#	3491#	3544#	3550#	3566#	3574#
3603#	3605#	3608#	3611#	3612#	3635#	3641#	3668#	3674#	3703#	3709#	3744#	3750#
3774#	3782#	3815#	3817#	3820#	3823#	3824#	3857#	3863#	3906#	3912#	3933#	3941#
3974#	3976#	3979#	3982#	3983#	4009#	4015#	4062#	4068#	4102#	4110#	4152#	4156#
4162#	4164#	4173#	4179#									
4#	651	652#	653	654#	1043	1044#	1045	1046#	1246#	1248	1249#	1250#
1253	1254#	1256	1257#	1261#	1281	1282#	1289#	1291	1292#	1293#	1296	1297#
1299	1300#	1304#	1324	1325#	1344	1346#	1376	1377#	1384#	1386	1387#	1388#
1391	1392#	1409	1410#	1421	1422#	1431	1433#	1554	1555#	1556	1557#	1742
1743#	1744#	1747	1748#	1789	1790#	2080	2081#	2088	2090#	2112	2113#	2120
2122#	2124	2125#	2132	2134#	2136	2137#	2144	2146#	2155	2156#	2161	2162#
2184	2186#	2219	2220#	2240	2241#	2248	2250#	2255	2257#	2269	2270#	2289
2290#	2292	2294	2295#	2300	2302#	2304	2305#	2307	2308#	2313	2315#	2327
2328#	2329	2332	2333#	2336	2337#	2375	2376#	2377	2380	2381#	2384	2385#
2426	2427#	2439	2440#	2447	2449#	2451	2452#	2471	2472#	2486	2488#	2637
2638#	2708	2709#	2718	2720#	2722	2723#	2881	2882#	2928	2929#	2964	2965#
2980	2981#	3041	3042#	3054	3055#	3094	3095#	3115	3116#	3127	3128#	3139
3141	3142	3143#	3162	3163#	3164	3167	3168#	3171	3172#	3188	3190	3191
3192#	3198	3200#	3202	3203#	3209	3211#	3246	3247#	3370	3372	3373#	3428
3430	3431#	3482	3484	3485#	3541	3543	3544#	3565	3566#	3602	3603#	3604#
3607	3608#	3632	3634	3635#	3665	3667	3668#	3700	3702	3703#	3741	3743
3744#	3773	3774#	3814	3815#	3816#	3819	3820#	3854	3856	3857#	3903	3905
3906#	3932	3933#	3973	3974#	3975#	3978	3979#	4006	4008	4009#	4059	4061
4062#	4101	4102#	4151	4152#	4155	4156#	4172	4173#				
655#	656#	663#	664#	665#	666#	1041#	1042#	1047#	1048#	1055#	1056#	1057#
1058#	1157#	1158#	1165#	1166#	1173#	1174#	1181#	1182#	1251#	1254#	1260#	1273#
1275#	1276#	1278#	1279	1294#	1297#	1303#	1316#	1318#	1319#	1321#	1322	1326#
1327#	1328#	1329#	1344#	1345#	1353#	1355#	1389#	1392#	1393#	1394#	1395#	1396#
1413#	1415#	1416#	1418#	1419	1431#	1432#	1440#	1442#	1552#	1553#	1558#	1559#
1560#	1561#	1562#	1563#	1590#	1591#	1631#	1632#	1745#	1748#	1749#	1750#	1751#
1752#	1762#	1763#	1782#	1787#	1792#	1794#	1795#	1796#	1797#	2029#	2030#	2061#
2065#	2066#	2070#	2071#	2075#	2076#	2077#	2088#	2089#	2091#	2095#	2096#	2100#
2101#	2105#	2106#	2107#	2109#	2110#	2120#	2121#	2132#	2133#	2144#	2145#	2148#
2150#	2152#	2157#	2159#	2163#	2164#	2165#	2168#	2176#	2178#	2179#	2182#	2184#
2185#	2187#	2188#	2189#	2190#	2191#	2194#	2196#	2200#	2208#	2210#	2212#	2221#
2225#	2226#	2230#	2231#	2235#	2236#	2237#	2248#	2249#	2253#	2255#	2256#	2259#
2265#	2267#	2277#	2296#	2297#	2298#	2300#	2301#	2309#	2310#	2311#	2313#	2314#
2316#	2317#	2318#	2320#	2330#	2337#	2338#	2339#	2340#	2342#	2350#	2351#	2359#
2360#	2368#	2369#	2378#	2385#	2386#	2387#	2388#	2390#	2398#	2399#	2407#	2408#
2416#	2417#	2422#	2424#	2433#	2435#	2436#	2437#	2447#	2448#	2453#	2454#	2455#
2456#	2457#	2458#	2459#	2461#	2462#	2463#	2464#	2466#	2467#	2470#	2479#	2480#
2481#	2482#	2483#	2484	2486#	2487#	2496#	2498#	2500#	2625#	2626#	2632#	2633#
2634#	2635#	2641#	2657#	2659#	2667#	2668#	2693#	2694#	2702#	2703#	2718#	2719#
2732#	2734#	2758#	2759#	2896#	2931#	2973#	2989#	3050#	3063#	3103#	3124#	3136#
3151#	3153#	3155#	3156#	3160#	3165#	3172#	3173#	3174#	3175#	3177#	3184#	3185#
3198#	3199#	3209#	3210#	3217#	3219#	3221#	3222#	3229#	3230#	3237#	3238#	3239#
3240#	3241#	3242#	3259#	3261#	3262#	3267#	3269#	3367#	3368#	3379#	3422#	3424#
3425#	3426#	3437#	3479#	3480#	3491#	3535#	3537#	3538#	3539#	3550#	3558#	3560#
3561#	3562#	3563#	3574#	3605#	3608#	3609#	3610#	3611#	3612#	3641#	3674#	3709#
3750#	3766#	3768#	3769#	3770#	3771#	3782#	3817#	3820#	3821#	3822#	3823#	3824#
3863#	3912#	3925#	3927#	3928#	3929#	3930#	3941#	3976#	3979#	3980#	3981#	3982#

\$TAGNU= 050203

\$TEMP = 050202

\$TSK0 = 050202

3983#	4015#	4068#	4094#	4096#	4097#	4098#	4099#	4110#	4162#	4164#	4179#	
652#	665	1044#	1057	1249#	1251	1254#	1276	1282#	1355	1377#	1442	1555#
1562	1743#	1745	1748#	1752	1790#	1792	2081#	2089	2090#	2212	2220#	2256
2257#	2259	2270#	2277	2290#	2301	2302#	2320	2337#	2369	2385#	2417	2427#
2500	2638#	2641	2709#	2719	2720#	2734	2882#	2896	2929#	2931	2965#	2973
2981#	2989	3042#	3050	3055#	3063	3095#	3103	3116#	3124	3128#	3136	3143#
3151	3172#	3230	3247#	3259	3373#	3379	3431#	3437	3485#	3491	3544#	3550
3566#	3574	3603#	3605	3608#	3612	3635#	3641	3668#	3674	3703#	3709	3744#
3750	3774#	3782	3815#	3817	3820#	3824	3857#	3863	3906#	3912	3933#	3941
3974#	3976	3979#	3983	4009#	4015	4062#	4068	4102#	4110	4152#	4164	4173#
4179												

\$TSK1 = 050201

654#	665	1046#	1057	1254#	1275	1292#	1294	1297#	1319	1325#	1345	1346#
1353	1387#	1389	1392#	1416	1422#	1432	1433#	1440	1557#	1562	1748#	1751
2113#	2121	2122#	2152	2156#	2210	2241#	2249	2250#	2253	2295#	2298	2305#
2314	2315#	2318	2337#	2360	2361#	2369	2385#	2408	2409#	2417	2440#	2448
2449#	2498	2723#	2732	3172#	3222	3223#	3230	3608#	3611	3820#	3823	3979#
3982	4156#	4162										

\$TSK2 = 177777

1257#	1260	1261#	1273	1297#	1318	1392#	1415	2125#	2133	2134#	2150	2162#
2185	2186#	2208	2308#	2311	2337#	2351	2352#	2369	2385#	2399	2400#	2417
2452#	2487	2488#	2496	3172#	3185	3186#	3230					

\$TSK3 = 177777

1300#	1303	1304#	1316	1410#	1413	2137#	2145	2146#	2148	2337#	2342	2343#
2369	2385#	2390	2391#	2417	2472#	2483	3172#	3177	3178#	3230		
2337#	2350	2359	2368	2385#	2398	2407	2416	3172#	3184	3221	3229	

\$TSK4 = 050135

\$TSK5 = 050143

\$TSK6 = 050145

\$\$ARGC= 000000

\$\$BYTE= 000402

4#	652#	1044#	1255#	1280#	1298#	1323#	1375#	1408#	1420#	1555#	1788#	2079#
2111#	2123#	2135#	2154#	2160#	2218#	2239#	2268#	2288#	2291#	2293#	2303#	2306#
2425#	2438#	2450#	2636#	2707#	2721#	2880#	2927#	2963#	2979#	3040#	3053#	3093#
3114#	3126#	3138#	3140#	3187#	3189#	3201#	3245#	3369#	3371#	3427#	3429#	3481#
3483#	3540#	3542#	3564#	3631#	3633#	3664#	3666#	3699#	3701#	3740#	3742#	3772#
3853#	3855#	3902#	3904#	3931#	4005#	4007#	4058#	4060#	4100#	4150#	4154#	4171#

\$\$CASE= 000404

4#	2341#	2342#	2350	2351#	2359	2360#	2389#	2390#	2398	2399#	2407	2408#
3176#	3177#	3184	3185#	3221	3222#							

\$\$DST = 000003

4#	1783#	2062#	2067#	2072#	2092#	2097#	2102#	2165#	2179#	2191#	2197#	2222#
2227#	2232#	2467#										

\$\$ELOC= 000402

4#	1255#	1259#	1260	1273#	1280#	1298#	1302#	1303	1316#	1323#	1344	1353#
1355#	1375#	1408#	1412#	1413#	1420#	1431	1440#	1442#	1788#	1792#	2079#	2088
2111#	2120	2123#	2132	2135#	2144	2148#	2150#	2152#	2154#	2160#	2184	2208#
2210#	2212#	2218#	2239#	2248	2253#	2255	2259#	2268#	2277#	2288#	2291#	2298#
2300	2303#	2306#	2311#	2313	2318#	2320#	2425#	2438#	2447	2450#	2486	2496#
2498#	2500#	2636#	2641#	2707#	2718	2721#	2732#	2734#	2880#	2896#	2927#	2931#
2963#	2973#	2979#	2989#	3040#	3050#	3053#	3063#	3093#	3103#	3114#	3124#	3126#
3136#	3138#	3151#	3187#	3198	3201#	3209	3217#	3219#	3245#	3259#	3369#	3379#
3427#	3437#	3481#	3491#	3540#	3550#	3564#	3574#	3631#	3641#	3664#	3674#	3699#
3709#	3740#	3750#	3772#	3782#	3853#	3863#	3902#	3912#	3931#	3941#	4005#	4015#
4058#	4068#	4100#	4110#	4150#	4154#	4162#	4164#	4171#	4179#			

\$\$ERFL= 000000

\$\$FLAG= 000001

4#	1278#	1321#	1418#									
4#	651#	652#	654#	1043#	1044#	1046#	1255#	1257#	1273#	1280#	1282#	1298#
1300#	1316#	1323#	1325#	1353#	1355#	1375#	1377#	1408#	1410#	1413#	1420#	1422#
1440#	1442#	1554#	1555#	1557#	1788#	1790#	1792#	2079#	2081#	2111#	2113#	2123#
2125#	2135#	2137#	2148#	2150#	2152#	2154#	2156#	2160#	2162#	2208#	2210#	2212#
2218#	2220#	2239#	2241#	2253#	2259#	2268#	2270#	2277#	2288#	2290#	2291#	2293#
2295#	2298#	2303#	2305#	2306#	2308#	2311#	2318#	2320#	2425#	2427#	2438#	2440#
2450#	2452#	2496#	2498#	2500#	2636#	2638#	2641#	2707#	2709#	2721#	2723#	2732#
2734#	2880#	2882#	2896#	2927#	2929#	2931#	2963#	2965#	2973#	2979#	2981#	2989#

	3040#	3042#	3050#	3053#	3055#	3063#	3093#	3095#	3103#	3114#	3116#	3124#	3126#
	3128#	3136#	3138#	3140#	3143#	3151#	3187#	3189#	3192#	3201#	3203#	3217#	3219#
	3245#	3247#	3259#	3369#	3371#	3373#	3379#	3427#	3429#	3431#	3437#	3481#	3483#
	3485#	3491#	3540#	3542#	3544#	3550#	3564#	3566#	3574#	3631#	3633#	3635#	3641#
	3664#	3666#	3668#	3674#	3699#	3701#	3703#	3709#	3740#	3742#	3744#	3750#	3772#
	3774#	3782#	3853#	3855#	3857#	3863#	3902#	3904#	3906#	3912#	3931#	3933#	3941#
	4005#	4007#	4009#	4015#	4058#	4060#	4062#	4068#	4100#	4102#	4110#	4150#	4152#
	4154#	4156#	4162#	4164#	4171#	4173#	4179#						
\$\$FROM= 000000	4#	1211#	2894#	2901#	2937#	2939#	2971#	2987#	3022#	3048#	3061#	3068#	3101#
	3112#	3122#	3134#	3149#	3243#	3327#	3329#	3572#	3597#	3599#	3780#	3809#	3811#
\$\$LOC = 033754	3939#	3968#	3970#	4108#	4136#	4198#	4200#	4211#	4224#	4226#	4228#		
	4#	653#	654	1045#	1046	1256#	1257	1281#	1282	1299#	1300	1324#	1325
	1376#	1377	1409#	1410	1421#	1422	1556#	1557	1789#	1790	2080#	2081	2112#
	2113	2124#	2125	2136#	2137	2155#	2156	2161#	2162	2219#	2220	2240#	2241
	2269#	2270	2289#	2290	2292#	2293	2294#	2295	2304#	2305	2307#	2308	2329#
	2330	2377#	2378	2426#	2427	2439#	2440	2451#	2452	2484#	2485	2637#	2638
	2708#	2709	2722#	2723	2881#	2882	2928#	2929	2964#	2965	2980#	2981	3041#
	3042	3054#	3055	3094#	3095	3115#	3116	3127#	3128	3141#	3142	3164#	3165
	3190#	3191	3202#	3203	3246#	3247	3370#	3371	3372#	3373	3428#	3429	3430#
	3431	3482#	3483	3484#	3485	3541#	3542	3543#	3544	3565#	3566	3632#	3633
	3634#	3635	3665#	3666	3667#	3668	3700#	3701	3702#	3703	3741#	3742	3743#
	3744	3773#	3774	3854#	3855	3856#	3857	3903#	3904	3905#	3906	3932#	3933
	4006#	4007	4008#	4009	4059#	4060	4061#	4062	4101#	4102	4151#	4152	4155#
	4156	4172#	4173										
\$\$LOCN= 000000	4#												
\$\$REG = 177777	4#												
\$\$RETU= 000000	4#												
\$\$RTN1= 000000	4#												
\$\$RTN2= 000000	4#												
\$\$SRC = 000027	4#	1783#	2062#	2067#	2072#	2092#	2097#	2102#	2165#	2179#	2191#	2197#	2222#
	2227#	2232#	2467#										
\$\$TGSV= 050135	4#	2337#	2338	2339	2340	2368#	2369	2385#	2386	2387	2388	2416#	2417
	3172#	3173	3174	3175	3229#	3230							
\$\$TGS1= 000001	4#	2342#	2343	2351#	2352	2360#	2361	2390#	2391	2399#	2400	2408#	2409
	3177#	3178	3185#	3186	3222#	3223							
\$\$TGS2= 000000	4#												
\$\$TO = 000000	4#	1211#	1212	2894#	2895	2901#	2902	2937#	2938	2939#	2940	2971#	2972
	2987#	2988	3022#	3023	3048#	3049	3061#	3062	3068#	3069	3101#	3102	3112#
	3113	3122#	3123	3134#	3135	3149#	3150	3243#	3244	3327#	3328	3329#	3330
	3572#	3573	3597#	3598	3599#	3600	3780#	3781	3809#	3810	3811#	3812	3939#
	3940	3968#	3969	3970#	3971	4108#	4109	4136#	4137	4198#	4199	4200#	4201
	4211#	4212	4224#	4225	4226#	4227	4228#	4229					
\$\$\$TAG= 050000	4#												
= 034552	6#	138#	142#	492#	501#	510#	526	630#	653	654	669	694	720
	746	772	804	806#	858	879	881#	887#	907	930	932#	953	955#
	977	979#	1001	1024	1026#	1045	1046	1061	1082#	1099	1101#	1135	1256
	1257	1267	1269	1281	1282	1299	1300	1310	1312	1324	1325	1362#	1376
	1377	1402	1404	1409	1410	1421	1422	1556	1557	1789	1790	2080	2081
	2112	2113	2124	2125	2136	2137	2155	2156	2161	2162	2219	2220	2240
	2241	2269	2270	2289	2290	2292	2293	2294	2295	2304	2305	2307	2308
	2329	2330	2377	2378	2426	2427	2439	2440	2451	2452	2484	2485	2571#
	2614	2616	2637	2638	2670	2708	2709	2722	2723	2762#	2783	2809	2836
	2881	2882	2928	2929	2948	2958	2960	2964	2965	2980	2981	3035	3037
	3041	3042	3054	3055	3094	3095	3115	3116	3127	3128	3141	3142	3164
	3165	3190	3191	3202	3203	3246	3247	3272	3370	3371	3372	3373	3428
	3429	3430	3431	3482	3483	3484	3485	3541	3542	3543	3544	3565	3566

PARAMETER CODING
CZTSIB.P11

19-AUG-79 20:26

MACY11 30(1046)

19-AUG-79 20:27 PAGE 138

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0166

3632	3633	3634	3635	3665	3666	3667	3668	3700	3701	3702	3703	3741
3742	3743	3744	3773	3774	3786	3854	3855	3856	3857	3903	3904	3905
3906	3932	3933	4006	4007	4008	4009	4032	4059	4060	4061	4062	4087
4101	4102	4151	4152	4155	4156	4172	4173	4185	4207	4209	4269	4296
4300#	4310#	4333	4340									

.CORD = 000004 G
 .INCOR= 000002 G
 .MULT = 000200 G
 .PREAM= 000040 G
 .RDFMK= 000100 G
 .OOF9 = 000010 G
 .8OF9 = 000001 G
 .9OF9 = 000020 G

333# 1919
 334# 1925
 328# 1954
 330# 1994
 329# 2014
 332# 1931
 335# 1900
 331# 1939

HEADER	2#	4#	18												
IF	2#	1254	1279	1297	1322	1374	1407	1419	1787	2078	2110	2122	2134	2153	2159
	2217	2238	2267	2287	2290	2302	2305	2424	2437	2449	2635	2720	2879	2926	2962
IFB	2#	2706	3039	3052	3244	3368	3426	3480	3539	3630	3663	3698	3739	3852	3901
	4004	4057	4149	4153	4170										
IFCOND	2#														
IF.ERR	2#														
IF.NO.	2#														
INCR	2#	1246	1289	1384	1740	3600	3812	3971							
INCRU	2#														
INLINE	2#														
INLOOP	2#	4#													
IOSETU	2#	4#													
IOSTAR	2#	4#													
KT11	2#	4#													
LASTAD	2#	4#	4313												
LEAVE	2#	1257	1300	1410											
LET	2#	654	662	1040	1046	1054	1156	1164	1172	1180	1325	1327	1392	1394	1551
	1557	1559	1589	1630	1748	1761	1781	1793	1795	2028	2060	2065	2070	2075	2090
	2095	2100	2105	2108	2156	2162	2164	2175	2178	2186	2188	2190	2195	2220	2225
	2230	2235	2264	2295	2308	2315	2421	2432	2435	2452	2454	2456	2458	2461	2463
	2466	2478	2480	2624	2631	2633	2656	2666	2692	2701	2757	3152	3155	3236	3238
	3240	3260	3266	3366	3421	3424	3478	3534	3537	3557	3560	3561	3608	3765	3768
	3769	3820	3924	3927	3928	3979	4093	4096	4097						
LOCAL	2#														
LOOP	2#														
MANUAL	2#	4#													
MEMORY	2#	4#													
MSBYTE	2#	4#	19#	25	26	27									
MSCHEC	2#	4#	668#	693#	719#	745#	771#	803#	857#	878#	906#	929#	952#	976#	1000#
	1023#	1060#	1098#	1134#	2669#	2782#	2808#	2835#	2947#	3271#	3785#	4031#	4086#	4184#	4269#
MSCNTO	2#	4#	4257#	4262#	4291#										
MSCOUN	2#	4#	657#	678#	684#	704#	710#	730#	736#	756#	762#	782#	789#	796#	814#
	821#	827#	837#	847#	868#	894#	900#	917#	923#	940#	946#	964#	970#	988#	994#
	1011#	1017#	1035#	1049#	1086#	1092#	1110#	1122#	1128#	1203#	1283#	1335#	1347#	1378#	1423#
	1434#	2082#	2114#	2126#	2138#	2170#	2202#	2242#	2271#	2344#	2353#	2362#	2392#	2401#	2410#
	2441#	2473#	2489#	2710#	2724#	2888#	3253#								
MSDATA	2#	4#	19#	28	30	32	34	36	38	40	42	44	46	48	50
	52	54	56	58#	60	62	65	68	70	72	74	76	78	80	82
	84	86	88	90	92	94	96	98	100	102	133#	140#			
MSDECR	2#	4#	160#	179#	182#	672#	698#	724#	750#	776#	808#	862#	883#	911#	934#
	957#	981#	1005#	1028#	1066#	1103#	1116#	1139#	1159#	1167#	1175#	1183#	2574#	2588#	2679#
	2737#	2789#	2815#	2842#	2846#	2898#	2903#	2933#	2941#	2999#	3065#	3076#	3105#	3231#	3276#
	3382#	3440#	3494#	3553#	3579#	3644#	3677#	3720#	3761#	3790#	3873#	3921#	3947#	4035#	4090#
	4115#	4189#	4214#	4232#	4236#	4275#	4303#	4320#	4329#						
MSDEFA	2#	4#	4257#	4262#	4291#										
MSENDE	2#	4#	160#	179#	182#	672#	698#	724#	750#	776#	808#	862#	883#	911#	934#
	957#	981#	1005#	1028#	1066#	1103#	1116#	1139#	1159#	1167#	1175#	1183#	2574#	2679#	2737#
	2789#	2815#	2842#	2846#	2898#	2903#	2933#	2941#	2999#	3065#	3076#	3105#	3231#	3276#	3382#
	3440#	3494#	3553#	3579#	3644#	3677#	3720#	3761#	3790#	3873#	3921#	3947#	4035#	4090#	4115#
	4189#	4214#	4232#	4236#	4275#	4303#	4320#								
MSERRI	2#	4#	1198#	1330#	2428#	2604#	2883#	2966#	2982#	3043#	3056#	3096#	3117#	3129#	3144#
	3179#	3193#	3204#	3212#	3224#	3248#	3374#	3432#	3486#	3545#	3567#	3636#	3669#	3704#	3745#
	3775#	3858#	3907#	3934#	4010#	4063#	4103#	4157#	4174#						

M\$FSCA	2#	4#														
M\$ECS	2#	4#														
M\$EXCP	2#	4#	4257#	4262#												
M\$EXIT	2#	4#	668#	693#	719#	745#	771#	803#	857#	878#	906#	929#	952#	976#	1000#	
	1023#	1060#	1098#	1134#	2669#	2670#	2782#	2783#	2808#	2835#	2947#	2948#	3271#	3272#	3785#	
	3786#	4031#	4032#	4086#	4087#	4184#	4185#	4269#	4296#							
M\$EXSE	2#	4#	668#	693#	719#	745#	771#	803#	857#	878#	906#	929#	952#	976#	1000#	
	1023#	1060#	1098#	1134#	2669#	2782#	2808#	2835#	2947#	3271#	3785#	4031#	4086#	4184#	4269#	
	4296#															
M\$EXTJ	2#	4#	668#	669#	693#	694#	719#	720#	745#	746#	771#	772#	803#	804#	857#	
	858#	878#	879#	906#	907#	929#	930#	952#	953#	976#	977#	1000#	1001#	1023#	1024#	
	1060#	1061#	1098#	1099#	1134#	1135#	2669#	2782#	2808#	2809#	2835#	2836#	2947#	3271#	3785#	
	4031#	4086#	4184#	4269#	4296#											
M\$GEN	2#	4#	19#	28#	30#	32#	34#	36#	38#	40#	42#	44#	46#	48#	50#	
	52#	54#	56#	58#	60#	62#	65#	68#	70#	72#	74#	76#	78#	80#	82#	
	84#	86#	88#	90#	92#	94#	96#	98#	100#	102#	114#	133#	140#	153#	154#	
	160#	171#	172#	179#	649#	672#	676#	698#	702#	724#	728#	750#	754#	776#	780#	
	808#	812#	862#	866#	883#	892#	911#	915#	934#	938#	957#	962#	981#	986#	1005#	
	1009#	1028#	1033#	1066#	1084#	1103#	1108#	1116#	1120#	1139#	1155#	1159#	1163#	1167#	1171#	
	1175#	1179#	1183#	2583#	2598#	2679#	2690#	2737#	2772#	2789#	2799#	2815#	2826#	2842#	2870#	
	2873#	2898#	2903#	2921#	2933#	2941#	2999#	3016#	3026#	3065#	3076#	3084#	3087#	3105#	3109#	
	3231#	3276#	3324#	3341#	3382#	3393#	3440#	3451#	3494#	3506#	3553#	3579#	3594#	3615#	3644#	
	3648#	3677#	3681#	3720#	3724#	3761#	3790#	3806#	3827#	3873#	3877#	3921#	3947#	3965#	3986#	
	4035#	4039#	4090#	4115#	4130#	4189#	4195#	4214#	4220#	4232#	4254#	4276#	4288#	4304#	4317#	
	4334#	4338#														
M\$GENB	2#	4#														
M\$GETS	2#	4#	160#	179#	182#	672#	698#	724#	750#	776#	808#	862#	883#	911#	934#	
	957#	981#	1005#	1028#	1066#	1103#	1116#	1139#	1159#	1167#	1175#	1183#	2574#	2588#	2679#	
	2737#	2789#	2815#	2842#	2846#	2898#	2903#	2933#	2941#	2999#	3065#	3076#	3105#	3231#	3276#	
	3382#	3440#	3494#	3553#	3579#	3644#	3677#	3720#	3761#	3790#	3873#	3921#	3947#	4035#	4090#	
	4115#	4189#	4214#	4232#	4236#	4269#	4275#	4296#	4303#	4320#						
M\$GETT	2#	4#	668#	693#	719#	745#	771#	803#	857#	878#	906#	929#	952#	976#	1000#	
	1023#	1060#	1098#	1134#	2669#	2782#	2808#	2835#	2947#	3271#	3785#	4031#	4086#	4184#	4269#	
	4296#															
M\$GNGB	2#	4#	9#	19#	28#	30#	32#	34#	36#	38#	40#	42#	44#	46#	48#	
	50#	52#	54#	56#	58#	60#	62#	65#	68#	70#	72#	74#	76#	78#	80#	
	82#	84#	86#	88#	90#	92#	94#	96#	98#	100#	102#	113#	114#	133#	140#	
	152#	153#	154#	170#	171#	172#	186#	649#	676#	702#	728#	754#	780#	812#	866#	
	892#	915#	938#	962#	986#	1009#	1033#	1084#	1108#	1120#	1155#	1163#	1171#	1179#	2583#	
	2592#	2598#	2690#	2772#	2799#	2826#	2851#	4242#	4253#	4254#	4287#	4288#	4314#	4317#		
M\$GNIN	2#	4#	19#	20#	21#	22#	23#	24#	25#	26#	27#	28#	29#	30#	31#	
	32#	33#	34#	35#	36#	37#	38#	39#	40#	41#	42#	43#	44#	45#	46#	
	47#	48#	49#	50#	51#	52#	53#	54#	55#	56#	57#	58#	59#	60#	61#	
	62#	63#	64#	65#	66#	67#	68#	69#	70#	71#	72#	73#	74#	75#	76#	
	77#	78#	79#	80#	81#	82#	83#	84#	85#	86#	87#	88#	89#	90#	91#	
	92#	93#	94#	95#	96#	97#	98#	99#	100#	101#	102#	103#	113#	115#	116#	
	117#	118#	119#	120#	121#	122#	123#	124#	133#	134#	138#	140#	141#	142#	152#	
	170#	657#	658#	659#	660#	661#	668#	669#	673#	678#	679#	680#	681#	682#	684#	
	685#	686#	687#	688#	689#	690#	691#	693#	694#	699#	704#	705#	706#	707#	708#	
	710#	711#	712#	713#	714#	715#	716#	717#	719#	720#	725#	730#	731#	732#	733#	
	734#	736#	737#	738#	739#	740#	741#	742#	743#	745#	746#	751#	756#	757#	758#	
	759#	760#	762#	763#	764#	765#	766#	767#	768#	769#	771#	772#	777#	782#	783#	
	784#	785#	786#	787#	789#	790#	791#	792#	793#	794#	796#	797#	798#	799#	800#	
	801#	803#	804#	809#	814#	815#	816#	817#	818#	819#	821#	822#	823#	824#	825#	
	827#	828#	829#	830#	831#	832#	833#	834#	835#	837#	838#	839#	840#	841#	842#	
	843#	844#	845#	847#	848#	849#	850#	851#	852#	853#	854#	855#	857#	858#	863#	

	812#	818#	824#	834#	844#	854#	863#	866#	875#	884#	892#	897#	903#	912#	915#
	920#	926#	935#	938#	943#	949#	958#	962#	967#	973#	982#	986#	991#	997#	1006#
	1009#	1014#	1020#	1029#	1033#	1038#	1052#	1067#	1084#	1089#	1095#	1104#	1108#	1113#	1117#
	1120#	1125#	1131#	1140#	1155#	1163#	1171#	1179#	1190#	1198#	1208#	1228#	1231#	1236#	1271#
	1286#	1314#	1330#	1340#	1350#	1381#	1406#	1426#	1429#	1437#	2085#	2117#	2129#	2141#	2173#
	2205#	2245#	2274#	2347#	2356#	2365#	2395#	2404#	2413#	2428#	2444#	2476#	2493#	2583#	2592#
	2598#	2604#	2621#	2628#	2639#	2644#	2664#	2669#	2680#	2690#	2699#	2705#	2714#	2728#	2738#
	2772#	2776#	2779#	2782#	2790#	2799#	2816#	2826#	2843#	2851#	2870#	2871#	2873#	2874#	2876#
	2878#	2883#	2891#	2899#	2904#	2921#	2922#	2924#	2934#	2942#	2947#	2966#	2982#	3000#	3016#
	3017#	3026#	3027#	3043#	3056#	3066#	3077#	3084#	3085#	3087#	3088#	3096#	3106#	3109#	3110#
	3117#	3129#	3144#	3179#	3193#	3204#	3212#	3224#	3232#	3235#	3248#	3256#	3264#	3271#	3277#
	3324#	3325#	3341#	3342#	3374#	3383#	3393#	3394#	3432#	3441#	3451#	3452#	3486#	3495#	3506#
	3507#	3545#	3554#	3567#	3580#	3594#	3595#	3615#	3616#	3636#	3645#	3648#	3649#	3669#	3678#
	3661#	3682#	3704#	3721#	3724#	3725#	3745#	3762#	3775#	3785#	3791#	3806#	3807#	3827#	3828#
	3858#	3874#	3877#	3878#	3907#	3922#	3934#	3948#	3965#	3966#	3986#	3987#	4010#	4031#	4036#
	4039#	4040#	4063#	4086#	4091#	4103#	4116#	4130#	4131#	4157#	4174#	4184#	4190#	4195#	4196#
	4215#	4220#	4221#	4233#	4242#	4253#	4287#	4328#	4329#	4333	4335				
MSIOSE	2#	4#													
M\$LDRO	2#	4#	1227#	1230#	2620#	2627#	2644#	2704#	2775#	2778#	3234#	3263#			
MSMASK	2#	4#													
MSMCHI	2#	4#													
MSMCLO	2#	4#													
MSMSK1	2#	4#													
M\$POP	2#	4#	160#	179#	182#	672#	698#	724#	750#	776#	808#	862#	883#	911#	934#
	957#	981#	1005#	1028#	1066#	1103#	1116#	1139#	1159#	1167#	1175#	1183#	2574#	2588#	2679#
	2737#	2789#	2815#	2842#	2846#	2898#	2903#	2933#	2941#	2999#	3065#	3076#	3105#	3231#	3276#
	3382#	3440#	3494#	3553#	3579#	3644#	3677#	3720#	3761#	3790#	3873#	3921#	3947#	4035#	4090#
M\$PRIN	4115#	4189#	4214#	4232#	4236#	4275#	4303#	4320#							
	2#	4#	657#	678#	684#	704#	710#	730#	736#	756#	762#	782#	789#	796#	814#
	821#	827#	837#	847#	868#	894#	900#	917#	923#	940#	946#	964#	970#	988#	994#
	1011#	1017#	1035#	1049#	1086#	1092#	1110#	1122#	1128#	1203#	1283#	1335#	1347#	1378#	1423#
	1434#	2082#	2114#	2126#	2138#	2170#	2202#	2242#	2271#	2344#	2353#	2362#	2392#	2401#	2410#
M\$PUSH	2441#	2473#	2489#	2710#	2724#	2888#	3253#								
	2#	4#	9#	152#	170#	186#	649#	676#	702#	728#	754#	780#	812#	866#	892#
	915#	938#	962#	986#	1009#	1033#	1084#	1108#	1120#	1155#	1163#	1171#	1179#	2583#	2592#
	2598#	2690#	2772#	2799#	2826#	2851#	2870#	2871#	2873#	2874#	2876#	2921#	2922#	2924#	3016#
	3017#	3026#	3027#	3084#	3085#	3087#	3088#	3109#	3110#	3324#	3325#	3341#	3342#	3393#	3394#
	3451#	3452#	3506#	3507#	3594#	3595#	3615#	3616#	3648#	3649#	3681#	3682#	3724#	3725#	3806#
	3807#	3827#	3828#	3877#	3878#	3965#	3966#	3986#	3987#	4039#	4040#	4130#	4131#	4195#	4196#
M\$PUT	4220#	4221#	4242#	4253#	4287#										
	2#	4#	657#	678#	684#	704#	710#	730#	736#	756#	762#	782#	789#	796#	814#
	821#	827#	837#	847#	868#	894#	900#	917#	923#	940#	946#	964#	970#	988#	994#
	1011#	1017#	1035#	1049#	1086#	1092#	1110#	1122#	1128#	1203#	1283#	1335#	1347#	1378#	1423#
	1434#	2082#	2114#	2126#	2138#	2170#	2202#	2242#	2271#	2344#	2353#	2362#	2392#	2401#	2410#
M\$PUT1	2441#	2473#	2489#	2660#	2695#	2710#	2724#	2888#	3253#						
	2#	4#	657#	658	678#	679	684#	685	686	687	688	704#	705	710#	711
	712	713	714	730#	731	736#	737	738	739	740	756#	757	762#	763	764
	765	766	782#	783	784	789#	790	791	796#	797	798	814#	815	816	821#
	822	827#	828	829	830	831	832	837#	838	839	840	841	842	847#	848
	849	850	851	852	868#	869	870	871	872	873	894#	895	900#	901	917#
	918	923#	924	940#	941	946#	947	964#	965	970#	971	988#	989	994#	995
	1011#	1012	1017#	1018	1035#	1036	1049#	1050	1086#	1087	1092#	1093	1110#	1111	1122#
	1123	1128#	1129	1203#	1204	1205	1206	1283#	1284	1335#	1336	1337	1338	1347#	1348
	1378#	1379	1423#	1424	1434#	1435	2082#	2083	2114#	2115	2126#	2127	2138#	2139	2170#
	2171	2202#	2203	2242#	2243	2271#	2272	2344#	2345	2353#	2354	2362#	2363	2392#	2393
	2401#	2402	2410#	2411	2441#	2442	2473#	2474	2489#	2490	2491	2660#	2661	2662	2663

E 14

PARAMETER CODING MACY11 30(1046) 19-AUG-79 20:27 PAGE 146
 CZTSIB.P11 19-AUG-79 20:26 CROSS REFERENCE TABLE -- MACRO NAMES SEQ 0173

	2695#	2696	2697	2698	2710#	2711	2712	2724#	2725	2726	2888#	2889	3253#	3254
MSRADI	2#	4#	4257#	4262#	4291#									
MSRBRO	2#	4#												
MSRNRO	2#	4#	1230#	1232	2644#	2645								
MSETS	2#	4#	9#	152#	170#	186#	649#	676#	702#	728#	754#	780#	812#	866# 892#
	915#	938#	962#	986#	1009#	1033#	1084#	1108#	1120#	1155#	1163#	1171#	1179#	2583# 2592#
	2598#	2690#	2772#	2799#	2826#	2851#	2871#	2874#	2876#	2922#	2924#	3017#	3027#	3085# 3088#
	3110#	3325#	3342#	3394#	3452#	3507#	3595#	3616#	3649#	3682#	3725#	3807#	3828#	3878# 3966#
	3987#	4040#	4131#	4196#	4221#	4242#	4253#	4287#						
MSSTAR	2#	4#												
MSSVC	2#	4#	657#	660	668#	672#	673	678#	681	684#	690	693#	698#	699 704#
	707	710#	716	719#	724#	725	730#	733	736#	742	745#	750#	751	756# 759
	762#	768	771#	776#	777	782#	786	789#	793	796#	800	803#	808#	809 814#
	818	821#	824	827#	834	837#	844	847#	854	857#	862#	863	868#	875 878#
	883#	884	894#	897	900#	903	906#	911#	912	917#	920	923#	926	929# 934#
	935	940#	943	946#	949	952#	957#	958	964#	967	970#	973	976#	981# 982
	988#	991	994#	997	1000#	1005#	1006	1011#	1014	1017#	1020	1023#	1028#	1029 1035#
	1038	1049#	1052	1060#	1066#	1067	1086#	1089	1092#	1095	1098#	1103#	1104	1110# 1113
	1116#	1117	1122#	1125	1128#	1131	1134#	1139#	1140	1190#	1198	1203#	1208	1227# 1228
	1230#	1231	1236#	1271#	1283#	1286	1314#	1330	1335#	1340	1347#	1350	1378#	1381 1406#
	1423#	1426	1429#	1434#	1437	2082#	2085	2114#	2117	2126#	2129	2138#	2141	2170# 2173
	2202#	2205	2242#	2245	2271#	2274	2344#	2347	2353#	2356	2362#	2365	2392#	2395 2401#
	2404	2410#	2413	2428	2441#	2444	2473#	2476	2489#	2493	2604	2620#	2621	2627# 2628
	2639#	2644#	2660#	2664	2669#	2679#	2680	2695#	2699	2704#	2705	2710#	2714	2724# 2728
	2737#	2738	2775#	2776	2778#	2779	2782#	2789#	2790	2808#	2815#	2816	2835#	2842# 2843
	2873#	2874	2876#	2878#	2883	2888#	2891	2898#	2899	2903#	2904	2921#	2922	2924# 2933#
	2934	2941#	2942	2947#	2966	2982	2999#	3000	3026#	3027	3043	3056	3065#	3066 3076#
	3077	3087#	3088	3096	3105#	3106	3109#	3110	3117	3129	3144	3179	3193	3204 3212
	3224	3231#	3232	3234#	3235	3248	3253#	3256	3263#	3264	3271#	3276#	3277	3341# 3342
	3374	3382#	3383	3393#	3394	3432	3440#	3441	3451#	3452	3486	3494#	3495	3506# 3507
	3545	3553#	3554	3567	3579#	3580	3615#	3616	3636	3644#	3645	3648#	3649	3669 3677#
	3678	3681#	3682	3704	3720#	3721	3724#	3725	3745	3761#	3762	3775	3785#	3790# 3791
	3827#	3828	3858	3873#	3874	3877#	3878	3907	3921#	3922	3934	3947#	3948	3986# 3987
	4010	4031#	4035#	4036	4039#	4040	4063	4086#	4090#	4091	4103	4115#	4116	4157 4174
	4184#	4189#	4190	4214#	4215	4232#	4233	4269#	4296#					
MSTLAB	2#	4#	660#	673#	681#	690#	699#	707#	716#	725#	733#	742#	751#	759# 768#
	777#	786#	793#	800#	809#	818#	824#	834#	844#	854#	863#	875#	884#	897# 903#
	912#	920#	926#	935#	943#	949#	958#	967#	973#	982#	991#	997#	1006#	1014# 1020#
	1029#	1038#	1052#	1067#	1089#	1095#	1104#	1113#	1117#	1125#	1131#	1140#	1190#	1198# 1208#
	1228#	1231#	1236#	1271#	1286#	1314#	1330#	1340#	1350#	1381#	1406#	1426#	1429#	1437# 2085#
	2117#	2129#	2141#	2173#	2205#	2245#	2274#	2347#	2356#	2365#	2395#	2404#	2413#	2428# 2444#
	2476#	2493#	2604#	2621#	2628#	2639#	2644#	2664#	2669#	2680#	2699#	2705#	2714#	2728# 2738#
	2776#	2779#	2782#	2790#	2816#	2843#	2874#	2876#	2878#	2883#	2891#	2899#	2904#	2922# 2924#
	2934#	2942#	2947#	2966#	2982#	3000#	3027#	3043#	3056#	3066#	3077#	3088#	3096#	3106# 3110#
	3117#	3129#	3144#	3179#	3193#	3204#	3212#	3224#	3232#	3235#	3248#	3256#	3264#	3271# 3277#
	3342#	3374#	3383#	3394#	3432#	3441#	3452#	3486#	3495#	3507#	3545#	3554#	3567#	3580# 3616#
	3636#	3645#	3649#	3669#	3678#	3682#	3704#	3721#	3725#	3745#	3762#	3775#	3785#	3791# 3828#
	3858#	3874#	3878#	3907#	3922#	3934#	3948#	3987#	4010#	4031#	4036#	4040#	4063#	4086# 4091#
	4103#	4116#	4157#	4174#	4184#	4190#	4215#	4233#						
MSTSTL	2#	4#	660#	673#	681#	690#	699#	707#	716#	725#	733#	742#	751#	759# 768#
	777#	786#	793#	800#	809#	818#	824#	834#	844#	854#	863#	875#	884#	897# 903#
	912#	920#	926#	935#	943#	949#	958#	967#	973#	982#	991#	997#	1006#	1014# 1020#
	1029#	1038#	1052#	1067#	1089#	1095#	1104#	1113#	1117#	1125#	1131#	1140#	1190#	1198# 1208#
	1228#	1231#	1236#	1271#	1286#	1314#	1330#	1340#	1350#	1381#	1406#	1426#	1429#	1437# 2085#
	2117#	2129#	2141#	2173#	2205#	2245#	2274#	2347#	2356#	2365#	2395#	2404#	2413#	2428# 2444#
	2476#	2493#	2604#	2621#	2628#	2639#	2644#	2664#	2669#	2680#	2699#	2705#	2714#	2728# 2738#

\$ADDON	2#	651	652	654	665	1043	1044	1046	1057	1246	1247	1249	1250	1254	1255
		1257	1261	1280	1282	1289	1290	1292	1293	1297	1298	1300	1304	1323	1346
		1375	1377	1384	1385	1387	1388	1392	1408	1410	1420	1422	1433	1554	1557
		1562	1741	1743	1744	1748	1788	1790	2079	2081	2090	2111	2113	2122	2125
		2134	2135	2137	2146	2154	2156	2160	2162	2186	2218	2220	2239	2241	2257
		2268	2270	2288	2290	2291	2295	2302	2303	2305	2306	2308	2315	2326	2333
		2337	2374	2376	2381	2385	2425	2427	2438	2440	2449	2450	2452	2471	2488
		2636	2638	2707	2709	2720	2721	2723	2880	2882	2927	2929	2963	2965	2981
		3040	3042	3053	3055	3093	3095	3114	3116	3126	3128	3138	3143	3161	3168
		3172	3187	3192	3200	3201	3203	3211	3245	3247	3369	3373	3427	3431	3485
		3540	3544	3564	3566	3601	3603	3604	3608	3631	3635	3664	3668	3699	3740
		3744	3772	3774	3813	3815	3816	3820	3853	3857	3902	3906	3931	3933	3974
		3975	3979	4005	4009	4058	4062	4100	4102	4150	4152	4154	4156	4171	
\$AND	2#	2291	3369	3427	3481	3540	3631	3664	3699	3740	3853	3902	4005	4058	
\$BRANC	2#	653	665	1045	1057	1248	1253	1256	1258	1275	1281	1291	1296	1299	1301
		1318	1324	1344	1376	1391	1409	1411	1415	1421	1431	1556	1562	1742	1747
		1751	1789	2080	2088	2112	2120	2124	2132	2136	2144	2155	2161	2184	2240
		2248	2255	2269	2289	2292	2294	2300	2304	2307	2313	2327	2329	2350	2375
		2377	2398	2407	2426	2439	2447	2451	2484	2486	2637	2708	2718	2722	2928
		2964	2980	3041	3054	3094	3115	3127	3139	3141	3162	3164	3184	3188	3198
		3202	3209	3221	3246	3370	3372	3428	3430	3482	3484	3541	3543	3565	3607
		3611	3632	3634	3665	3667	3700	3702	3741	3743	3773	3814	3819	3823	3856
		3903	3905	3932	3973	3978	3982	4006	4008	4059	4061	4101	4151	4172	
\$BRCOD	2#	1252	1295	1390	1746	3138	3187	3606	3818	3977					
\$CALL	2#	1211	2894	2901	2937	2939	2971	2987	3022	3048	3061	3068	3101	3112	3122
		3134	3149	3243	3327	3329	3572	3597	3780	3809	3811	3939	3968	3970	4108
		4136	4198	4200	4211	4224	4226	4228							
\$CHECK	2#	652	1044	1255	1280	1298	1323	1375	1408	1420	1555	1788	2079	2111	2123
		2135	2154	2160	2218	2239	2268	2288	2291	2303	2306	2425	2438	2450	2707
		2721	2880	2927	2963	2979	3040	3053	3093	3114	3126	3138	3187	3201	3369
		3427	3481	3540	3564	3631	3664	3699	3740	3772	3853	3902	3931	4005	4100
		4150	4154	4171											
\$CKOP1	2#	655	1041	1047	1157	1165	1173	1181	1247	1290	1326	1328	1385	1393	1395
		1552	1558	1590	1741	1749	1762	2029	2076	2106	2109	2163	2176	2187	2189
		2236	2433	2464	2625	2634	2667	2693	2702	3237	3241	3367	3422	3479	3601
		3609	3813	3821	3972	3980									
\$CKOP2	2#	663	1055	1250	1293	1388	1560	1744	1782	1794	1796	2061	2066	2071	2091
		2096	2101	2157	2165	2179	2191	2196	2221	2226	2231	2265	2296	2309	2422
		2436	2453	2455	2457	2459	2462	2467	2479	2481	2632	2657	2758	3153	3239
		3261	3267	3425	3538	3558	3560	3562	3604	3766	3768	3770	3816	3925	3929
		3975	4094	4096	4098										
\$CKR6	2#	1783	2062	2067	2072	2092	2097	2102	2165	2179	2191	2197	2222	2227	2232
		2467													
\$CMND	2#	652	1044	1255	1280	1298	1323	1375	1408	1420	1555	1788	2079	2111	2123
		2135	2154	2160	2218	2239	2268	2288	2291	2293	2303	2306	2425	2438	2636
		2707	2721	2880	2927	2963	2979	3040	3053	3093	3114	3126	3138	3140	3189
		3201	3245	3369	3371	3427	3429	3481	3483	3540	3542	3564	3631	3633	3666
		3699	3701	3740	3742	3772	3853	3855	3902	3904	3931	4005	4007	4058	4100
		4150	4154	4171											
\$COMPA	2#	652	1044	1247	1255	1280	1290	1298	1323	1375	1385	1408	1420	1555	1741
		1788	2079	2111	2123	2135	2154	2160	2218	2239	2268	2288	2291	2303	2425
		2438	2450	2636	2707	2721	2880	2927	2963	2979	3040	3053	3093	3114	3138
		3187	3201	3245	3369	3427	3481	3540	3564	3601	3631	3664	3699	3740	3813
		3853	3902	3931	3972	4005	4058	4100	4150	4154	4171				
\$COUNT	2#	1211	2894	2901	2937	2939	2971	2987	3022	3048	3061	3068	3101	3112	3122
		3134	3149	3243	3327	3329	3572	3597	3780	3809	3811	3939	3968	3970	4108

	4136	4198	4200	4211	4224	4226	4228								
\$DO	2#	652	1044	1555											
\$ELSE	2#														
\$ERRMS	2#														
\$EX1FA	2#														
\$EX1FO	2#														
\$EX1F2	2#														
\$EX1F3	2#														
\$GENBR	2#	653	665	1045	1057	1248	1253	1256	1258	1275	1281	1291	1296	1299	1301
		1318	1324	1344	1376	1386	1391	1409	1415	1421	1431	1556	1562	1742	1747
		1751	1789	2080	2088	2112	2120	2124	2136	2144	2155	2161	2184	2219	2240
		2248	2255	2269	2289	2292	2294	2300	2307	2313	2327	2329	2350	2359	2375
		2377	2398	2407	2426	2439	2447	2451	2484	2486	2637	2708	2718	2722	2881
		2964	2980	3041	3054	3094	3115	3127	3139	3141	3162	3164	3184	3188	3190
		3202	3209	3221	3246	3370	3372	3428	3430	3482	3484	3541	3543	3565	3602
		3611	3632	3634	3665	3667	3700	3702	3741	3743	3773	3814	3819	3823	3854
		3903	3905	3932	3973	3978	3982	4006	4008	4059	4061	4101	4151	4155	4172
\$GENTA	2#	651	666	1043	1058	1249	1251	1260	1273	1276	1278	1292	1294	1303	1316
		1319	1321	1345	1353	1355	1387	1389	1413	1416	1418	1432	1440	1442	1554
		1743	1745	1752	1792	2089	2121	2133	2145	2148	2150	2152	2185	2208	2210
		2249	2253	2256	2259	2277	2298	2301	2311	2314	2318	2320	2330	2332	2336
		2351	2360	2368	2378	2380	2384	2390	2399	2408	2416	2448	2471	2487	2496
		2500	2641	2719	2732	2734	2896	2931	2973	2989	3050	3063	3103	3124	3136
		3151	3165	3167	3171	3177	3185	3191	3199	3210	3217	3219	3222	3229	3259
		3437	3491	3550	3574	3603	3605	3612	3641	3674	3709	3750	3782	3815	3817
		3863	3912	3941	3974	3976	3983	4015	4068	4110	4162	4164	4179		
\$IF	2#	1255	1280	1298	1323	1375	1408	1420	1788	2079	2111	2123	2135	2154	2160
		2218	2239	2268	2288	2291	2303	2306	2425	2438	2450	2636	2707	2721	2880
		2963	2979	3040	3053	3093	3114	3126	3138	3187	3201	3245	3369	3427	3481
		3564	3631	3664	3699	3740	3772	3853	3902	3931	4005	4058	4100	4150	4171
\$IFCOD	2#	652	1044	1255	1280	1298	1323	1375	1408	1420	1555	1788	2079	2111	2123
		2135	2154	2160	2218	2239	2268	2288	2291	2293	2303	2306	2328	2376	2425
		2450	2483	2636	2707	2721	2880	2927	2963	2979	3040	3053	3093	3114	3126
		3163	3189	3201	3245	3369	3371	3427	3429	3481	3483	3540	3542	3564	3631
		3664	3666	3699	3701	3740	3742	3772	3853	3855	3902	3904	3931	4005	4007
		4060	4100	4150	4154	4171									
\$IFCON	2#														
\$IFOPR	2#	653	1045	1256	1281	1299	1324	1376	1409	1421	1556	1789	2080	2112	2124
		2136	2155	2161	2219	2240	2289	2292	2294	2304	2307	2329	2377	2426	2439
		2451	2484	2637	2708	2722	2881	2928	2964	2980	3041	3054	3094	3115	3127
		3164	3190	3202	3246	3370	3372	3428	3430	3482	3484	3541	3543	3565	3632
		3665	3667	3700	3702	3741	3743	3773	3854	3856	3903	3905	3932	4006	4008
		4061	4101	4151	4155	4172									
\$LET	2#	655	663	1041	1047	1055	1157	1165	1173	1181	1326	1328	1393	1395	1552
		1558	1560	1590	1631	1749	1762	1794	1796	2029	2061	2066	2071	2076	2091
		2096	2101	2106	2109	2157	2163	2176	2179	2187	2189	2191	2196	2221	2226
		2231	2236	2265	2296	2309	2316	2422	2433	2436	2453	2455	2457	2459	2462
		2467	2479	2481	2625	2632	2634	2657	2667	2693	2702	2758	3153	3156	3237
		3241	3261	3267	3367	3422	3425	3479	3535	3538	3558	3560	3562	3609	3766
		3770	3821	3925	3927	3929	3980	4094	4096	4098					
\$LPCNT	2#	1247	1290	1385	1741	3601	3813	3972							
\$OPABS	2#														
\$OPADD	2#	663	1055	1250	1293	1388	1560	1744	1796	2457	2462	2479	2632	2758	3604
		3816	3975												
\$OPAND	2#	2062	2067	2072	2092	2097	2102	2165	2179	2191	2197	2222	2227	2232	2467
\$OPCD1	2#	2106	2109	2176	2189	2433	2464	3422	3535						

PARAMETER	CODING	2106	2109	2176	2189	2433	2464											
\$OPSWB	2#	2106	2109	2176	2189	2433	2464											
\$OPXOR	2#	1783																
\$OR	2#	3138	3187															
\$PUT	2#																	
\$STRUC	2#																	
\$SUBON	2#	665	666	1057	1058	1251	1254	1260	1273	1275	1276	1278	1294	1297	1303			
	1316	1318	1319	1321	1345	1353	1355	1389	1392	1413	1415	1416	1418	1432	1440			
	1442	1562	1563	1745	1748	1751	1752	1792	2089	2121	2133	2145	2148	2150	2152			
	2185	2208	2210	2212	2249	2253	2256	2259	2277	2298	2301	2311	2314	2318	2320			
	2330	2338	2339	2340	2368	2369	2378	2386	2387	2388	2416	2417	2448	2483	2487			
	2496	2498	2500	2641	2719	2732	2734	2896	2931	2973	2989	3050	3063	3103	3124			
	3136	3151	3165	3173	3174	3175	3199	3210	3217	3219	3229	3230	3259	3379	3437			
	3491	3550	3574	3605	3608	3611	3612	3641	3674	3709	3750	3782	3817	3820	3823			
	3824	3863	3912	3941	3976	3979	3982	3983	4015	4068	4110	4162	4164	4179				
\$THEN	2#	1255	1280	1298	1323	1375	1408	1420	1788	2079	2111	2123	2135	2154	2160			
	2218	2239	2268	2288	2293	2303	2306	2425	2438	2450	2636	2707	2721	2880	2927			
	2963	2979	3040	3053	3093	3114	3126	3140	3189	3201	3245	3371	3429	3483	3542			
	3564	3633	3666	3701	3742	3772	3855	3904	3931	4007	4060	4100	4150	4154	4171			
\$TILA	2#																	
\$TILO	2#																	
\$UNTLL2	2#																	
\$UNTLL3	2#																	
\$WHILE	2#	651	1043	1554														
\$DEFA	2#																	
\$ENDS	2#	2369	2417	3230														
\$ERRO	2#																	
\$GEN	2#	651	666	1043	1058	1249	1251	1260	1273	1276	1278	1292	1294	1303	1316			
	1319	1321	1345	1353	1355	1387	1389	1413	1416	1418	1432	1440	1442	1554	1563			
	1743	1745	1752	1792	2089	2121	2133	2145	2148	2150	2152	2185	2208	2210	2212			
	2249	2253	2256	2259	2277	2298	2301	2311	2314	2318	2320	2330	2332	2336	2342			
	2351	2360	2368	2378	2380	2384	2390	2399	2408	2416	2448	2471	2487	2496	2498			
	2500	2641	2719	2732	2734	2896	2931	2973	2989	3050	3063	3103	3124	3136	3142			
	3151	3165	3167	3171	3177	3185	3191	3199	3210	3217	3219	3222	3229	3259	3379			
	3437	3491	3550	3574	3603	3605	3612	3641	3674	3709	3750	3782	3815	3817	3824			
	3863	3912	3941	3974	3976	3983	4015	4068	4110	4162	4164	4179						
\$GETS	2#	665	666	1057	1058	1251	1254	1260	1273	1275	1276	1278	1294	1297	1303			
	1316	1318	1319	1321	1344	1345	1353	1355	1389	1392	1413	1415	1416	1418	1431			
	1432	1440	1442	1562	1563	1745	1748	1751	1752	1792	2088	2089	2120	2121	2132			
	2133	2144	2145	2148	2150	2152	2184	2185	2208	2210	2212	2248	2249	2253	2255			
	2256	2259	2277	2298	2300	2301	2311	2313	2314	2318	2320	2330	2342	2350	2351			
	2359	2360	2368	2369	2378	2390	2398	2399	2407	2408	2416	2417	2447	2448	2483			
	2486	2487	2496	2498	2500	2641	2718	2719	2732	2734	2896	2931	2973	2989	3050			
	3063	3103	3124	3136	3151	3165	3177	3184	3185	3198	3199	3209	3210	3217	3219			
	3221	3222	3229	3230	3259	3379	3437	3491	3550	3574	3605	3608	3611	3612	3641			
	3674	3709	3750	3782	3817	3820	3823	3824	3863	3912	3941	3976	3979	3982	3983			
	4015	4068	4110	4162	4164	4179												
\$GETT	2#	1260	1303	1344	1431	2088	2120	2132	2144	2184	2248	2255	2300	2313	2342			
	2350	2351	2359	2360	2390	2398	2399	2407	2408	2447	2486	2718	3177	3184	3185			
	3198	3209	3221	3222														
\$LPCN	2#	1250	1293	1388	1744	3604	3816	3975										
\$POP	2#	665	666	1057	1058	1251	1254	1260	1273	1275	1276	1278	1294	1297	1303			
	1316	1318	1319	1321	1345	1353	1355	1389	1392	1413	1415	1416	1418	1432	1440			
	1442	1562	1563	1745	1748	1751	1752	1792	2089	2121	2133	2145	2148	2150	2152			
	2185	2208	2210	2212	2249	2253	2256	2259	2277	2298	2301	2311	2314	2318	2320			
	2330	2368	2369	2378	2416	2417	2448	2483	2487	2496	2498	2500	2641	2719	2732			
	2734	2896	2931	2973	2989	3050	3063	3103	3124	3136	3151	3165	3199	3210	3217			

	3219	3229	3230	3259	3379	3437	3491	3550	3574	3605	3608	3611	3612	3641	3674
	3709	3750	3782	3817	3820	3823	3824	3863	3912	3941	3976	3979	3982	3983	4015
	4068	4110	4162	4164	4179										
\$\$PUSH	2#	651	652	654	665	1043	1044	1046	1057	1246	1247	1249	1250	1254	1255
	1257	1261	1280	1282	1289	1290	1292	1293	1297	1298	1300	1304	1323	1325	1346
	1375	1377	1384	1385	1387	1388	1392	1408	1410	1420	1422	1433	1554	1555	1557
	1562	1741	1743	1744	1748	1788	1790	2079	2081	2090	2111	2113	2122	2123	2125
	2134	2135	2137	2146	2154	2156	2160	2162	2186	2218	2220	2239	2241	2250	2257
	2268	2270	2288	2290	2291	2295	2302	2303	2305	2306	2308	2315	2326	2328	2337
	2374	2376	2385	2425	2427	2438	2440	2449	2450	2452	2471	2472	2488	2636	2638
	2707	2709	2720	2721	2723	2880	2882	2927	2929	2963	2965	2979	2981	3040	3042
	3053	3055	3093	3095	3114	3116	3126	3128	3138	3143	3161	3163	3172	3187	3192
	3200	3201	3203	3211	3245	3247	3369	3373	3427	3431	3481	3485	3540	3544	3564
	3566	3601	3603	3604	3608	3631	3635	3664	3668	3699	3703	3740	3744	3772	3774
	3813	3815	3816	3820	3853	3857	3902	3906	3931	3933	3972	3974	3975	3979	4005
	4009	4058	4062	4100	4102	4150	4152	4154	4156	4171	4173				
\$\$SELE	2#	2337	2338	2339	2340	2385	2386	2387	2388	3172	3173	3174	3175		
\$\$SETS	2#	651	652	654	665	1043	1044	1046	1057	1246	1247	1249	1250	1254	1255
	1257	1261	1280	1282	1289	1290	1292	1293	1297	1298	1300	1304	1323	1325	1346
	1375	1377	1384	1385	1387	1388	1392	1408	1410	1420	1422	1433	1554	1555	1557
	1562	1741	1743	1744	1748	1788	1790	2079	2081	2090	2111	2113	2122	2123	2125
	2134	2135	2137	2146	2154	2156	2160	2162	2186	2218	2220	2239	2241	2250	2257
	2268	2270	2288	2290	2291	2295	2302	2303	2305	2306	2308	2315	2326	2328	2337
	2343	2352	2361	2374	2376	2385	2391	2400	2409	2425	2427	2438	2440	2449	2450
	2452	2471	2472	2488	2636	2638	2707	2709	2720	2721	2723	2880	2882	2927	2929
	2963	2965	2979	2981	3040	3042	3053	3055	3093	3095	3114	3116	3126	3128	3138
	3143	3161	3163	3172	3178	3186	3187	3192	3200	3201	3203	3211	3223	3245	3247
	3369	3373	3427	3431	3481	3485	3540	3544	3564	3566	3601	3603	3604	3608	3631
	3635	3664	3668	3699	3703	3740	3744	3772	3774	3813	3815	3816	3820	3853	3857
	3902	3906	3931	3933	3972	3974	3975	3979	4005	4009	4058	4062	4100	4102	4150
\$\$SETT	2#	2343	2352	2361	2391	2400	2409	3178	3186	3223					

. ABS. 034552 000

ERRORS DETECTED: 0

CZTSIB,CZTSIB/CRF/SOL/EQ:ONEFILE=NLISTF.P11,SVC.SML,SPMAC.SML,LISTF.P11,CZTSIB.P11
 RUN-TIME: 71 77 7 SECONDS
 RUN-TIME RATIO: 170/156=1.0
 CORE USED: 28K (56 PAGES)