

# PDP11/60

PDP-11/60 WCS DIAGNOSTIC  
MD-11-DQKUA-A

EP-DQKUA-A-DL  
COPYRIGHT © 1977  
FICHE 1 OF 2

JAN 1978  
**digital**  
MADE IN USA

This microfiche card contains a grid of 144 frames of diagnostic data. The frames are arranged in 12 rows and 12 columns. Each frame contains small, dense text, likely representing test results or system status information. The text is too small to read clearly but appears to be organized into columns and rows within each frame, possibly representing different components or test parameters.



# PDP11/60

PDP-11/60 WCS DIAGNOSTIC  
MD-11-DQKUA-A

EP-DQKUA-A-DL  
COPYRIGHT © 1977  
FICHE 2 OF 2

JAN 1978  
**digital**  
MADE IN USA



IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: MAINDEC-11-DQKUA-A-D  
PRODUCT NAME: PDP-11/60 WCS Diagnostic  
DATE CREATED: October, 1977  
LAST REVISION: October, 1977  
AUTHOR: Jim Kapadia

COPYRIGHT (C) 1977

DIGITAL EQUIPMENT CORPORATION  
146 Main Street  
Maynard, Massachusetts, USA 01754

This software is furnished to the purchaser under a license for use on a single computer system, and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL.

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.

DIGITAL assumes no responsibility for the use or reliability of its software on equipment not supplied by DIGITAL or its affiliated companies.

The following are trademarks of DIGITAL EQUIPMENT CORPORATION:

DIGITAL	FDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	



TABLE OF CONTENTS

1 0	ABSTRACT
2 0	REQUIREMENTS
2 1	HARDWARE
2 2	PRE-REQUISITE PROGRAMS
2 3	EXECUTION TIME
3 0	PROGRAM STRUCTURE AND DESCRIPTION
3 1	DIAGNOSTIC PHILOSOPHY
3 2	PROGRAM STRUCTURE
3 3	PROGRAM DESCRIPTION
4 0	STARTING PROCEDURE
4 1	PROGRAM LOADING
4 2	STARTING ADDRESS
4 3	PROGRAM AND OPERATOR ACTION
5 0	OPERATIONAL SWITCH SETTINGS
6 0	ERROR REPORTING
7 0	PROGRAM RECOVERY
7 1	POWER-FAIL
7 2	SPURIOUS ERRORS
8 0	AXDP AND ACT/APT COMPATABILITY
9 0	SUBROUTINE ABSTRACTS



## 1.0 ABSTRACT

THE WCS DIAGNOSTIC PROGRAM IS INTENDED TO BE USED FOR CHECKING THE WRITABLE CONTROL STORE OPTION OF THE PDP 11/60 PROCESSOR. IT HAS BEEN DESIGNED TO VERIFY THAT THE WCS OPERATES CORRECTLY AND ATTEMPTS TO DIAGNOSE FAULTS THAT MAY OCCUR THEREIN.

THE WCS IS A PDP 11/60 OPTION THAT PROVIDES THE USER WITH AN ABILITY TO DO HIS/HER OWN MICRO-PROGRAMMING. THE MICRO-CODE CAN BE LOADED AND STORED IN THE WCS AND EXECUTED UPON SPECIFIC INSTRUCTIONS FROM THE BASE MACHINE.

## 2.0 REQUIREMENTS

### 2.1 HARDWARE

THIS PROGRAM NEEDS THE FOLLOWING HARDWARE FOR PROPER EXECUTION

1. PDP 11/60 C. P. U.
2. 16 K OF MEMORY
3. WCS MODULE
4. CONSOLE TTY

### 2.2 PRE-REQUISITE PROGRAMS

THIS PROGRAM ASSUMES A GOOD CPU, MEMORY, CACHE AND CONSOLE TTY. HENCE APPROPRIATE DIAGNOSTIC PROGRAMS SHOULD BE RUN TO MAKE SURE THAT THE ABOVE HARDWARE IS GOOD.

### 2.3 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM WILL BE A QUICK-VERIFY PASS. SUBSEQUENT PASSES HAVE TEST ITERATIONS.

FIRST PASS TIME: 20. SEC (FAULT-FREE EXECUTION)  
SUBSEQUENT PASSES: 120. SEC



### 3.0 PROGRAM STRUCTURE AND DESCRIPTION

#### 3.1 DIAGNOSTIC PHILOSOPHY

THE PROGRAM USES A LOGICAL HIERCHY OF TESTING FOR THE WCS MODULE. IN THE BEGINNING SIMPLE TESTS ARE DONE TO CHECK THE PRIMITIVE LOGICAL FUNCTIONS. THESE TESTS HAVE BEEN SELECTED TO CHECK THE SMALLEST DISTINGUISHABLE PIECE OF HARDWARE. THUS PREVIOUSLY CHECKED HARDWARE IS USED IN SUBSEQUENT TESTS TO CHECK LARGER PORTIONS OF THE LOGIC. WHEN A FAULT IS DETECTED, IT CAN BE ASSOCIATED WITH THE PORTION OF HARDWARE BEING TESTED.

#### 3.2 PROGRAM STRUCTURE

THE PROGRAM IS ORGANIZED IN THE FOLLOWING MANNER:

1. COMMON DATA STORAGE CONSISTING OF:
  - A. TRAP-CATEHER
  - B. PARAMETER-TABLES AND TEMPORARY STORAGE LOCATIONS
  - C. ERROR ITEM TABLE
  - D. BIT DEFINITIONS
2. ERROR RECOVERY AND INITIALIZATION CONSISTING OF:
  - A. UNEXPECTED TRAP HANDLER (TO VECTOR 4)
  - B. UNEXPECTED PARITY ERROR TRAP HANDLER
  - C. INITIALIZATION CODE SETTING UP VARIOUS VECTORS AND PARAMETERS
3. TEST MODULES  
THIS SECTION CONTAINS THE INDIVIDUAL TESTS. EACH TEST IS PRECEDED BY A HEADER EXPLAINING THE PURPOSE OF THE TEST.
4. SUBROUTINES
5. ASCII MESSAGES
6. MICRO-CODE  
THIS SECTION CONTAINS THE MICRO-CODED TESTS, WHICH WILL BE LOADED INTO THE WCS PRIOR TO EXECUTION.



### 3.3 PROGRAM DESCRIPTION

UPON STARTING, THE PROGRAM IDENTIFIES ITSELF:

MAINDEC-11-DQKUA-A WCS DIAGNOSTIC

THE FIRST GROUP OF TESTS CHECK OUT THE INTERFACE OF THE WCS TO THE BASE MACHINE, VARIOUS REGISTERS ARE REFERENCED AND TESTED. READ/WRITE BITS OF THE STATUS REGISTER ARE CHECKED.

THEN VERY BASIC WRITE/READ FUNCTIONS OF THE WCS ARE PERFORMED USING THE WCS ADDRESS AND DATA REGISTER. EXPECTED BITS IN STATUS REGISTER ARE CHECKED.

HAVING CHECKED OUT THE WRITE/READ/SELECT LOGIC, THE HIGH AND LOW BYTE PARITY GENERATORS ARE TESTED. THE DIAGNOSTIC FUNCTION OF FORCING WRONG PARITY IS USED TO TEST THE PARITY ERROR DETECTION LOGIC, PARITY TRAP LOGIC, ETC.

AT THIS POINT THE BASIC OPERATIONS CAN BE PERFORMED AND EXTENSIVE CHECKING OF THE ADDRESS AND DATA PATHS IS DONE. TESTS OF THE WCS DATA MEMORY ARE INCLUDED.

UP TO THIS POINT, THE WCS HAS BEEN CHECKED FOR ITS ABILITY TO LOAD, STORE AND OUTPUT DATA. THE FIRST STEP IN USING THE WCS AS A CONTROL STORE IS TO TRANSFER CONTROL TO THE MICRO-CODE RESIDING IN IT. TESTS ARE DONE TO CHECK OUT ALL POSSIBLE WAYS OF TRANSFERRING CONTROL TO AND ENTERING THE WCS. PRIOR TO TRANSFERRING OF CONTROL, THE WCS IS PROPERLY LOADED WITH THE MICRO-CODE TEST WHICH WILL BE EXECUTED FROM THE WCS. A RETURN IS MADE TO THE BASE MACHINE AND IT IS CHECKED IF THE CONTROL WAS TRANSFERRED TO THE RIGHT POINT IN WCS.

FINALLY, VARIOUS BLOCK-TRANSFER FUNCTIONS OF THE WCS (LIKE LOAD GR'S, STORE GR'S ETC.) ARE CHECKED OUT.

AT THE END OF PASS OF THE PROGRAM, AN END OF PASS MESSAGE IS GIVEN:

END PASS N           : WHERE N IS THE PASS NUMBER

### 4.0 STARTING PROCEDURE



#### 4.1 PROGRAM LOADING

THE PROGRAM CAN BE LOADED FROM THE PAPER TAPE USING THE ABSOLUTE LOADER AND THE NORMAL LOADING PROCEDURE FOR ABSOLUTE FORMATTED BINARY TAPES.

IF AN XXDP DEVICE IS AVAILABLE, THE PROGRAM CAN BE LOADED USING THE STANDARD LOAD COMMANDS OF THE XXDP.

#### 4.2 STARTING ADDRESS

STARTING ADDRESS IS 200 (OCTAL).

#### 4.3 PROGRAM AND OPERATOR ACTION

AFTER LOADING THE PROGRAM INTO THE MEMORY, THE SWITCH REGISTER SHOULD BE CLEARED FOR NORMAL PROGRAM OPERATION. THEN LOAD ADDRESS 200. ON THE PDP 11/60, DEPRESS THE CONTROL AND START SWITCHES TOGETHER TO START THE PROGRAM.

IF ANY SWITCH OPTION IS TO BE SELECTED, THEN LOAD THE OCTAL EQUIVALENT OF THE SWITCH OPTION INTO THE SWITCH REGISTER. THIS CAN BE DONE BY FIRST KEYING IN THE OCTAL EQUIVALENT NUMBER AND THEN DEPRESSING "LSWR" SWITCH (LOAD SWITCH REGISTER). THEN THE PROGRAM CAN BE STARTED.

SWITCH NO.	OCTAL EQUIVALENT
0	000001
1	000002
2	000004
3	000010
4	000020
5	000040
6	000100
7	000200
8	000400
9	001000
10	002000
11	004000
12	010000
13	020000
14	040000
15	100000



ONCE STARTED, THE PROGRAM WILL CONTINUE TO RUN UNLESS HALTED BY THE USER. DEPRESS "CONTROL" AND "HALT" SWITCHES TOGETHER TO HALT THE PROGRAM.

AN ERROR FREE PASS OF THE PROGRAM WILL APPEAR AS:

```
MAINDEC-11-DQKUA-A WCS DIAGNOSTIC
END PASS 1
END PASS 2
:
```

IF AUTOMATIC RECOVERY AND RESTART OF THE PROGRAM IS DESIRED ON POWER-FAIL, THEN THE THREE POSITION SWITCH BOOT/RUN/HALT SHOULD BE LEFT IN THE "RUN" POSITION.

#### 5.0 OPERATIONAL SWITCH SETTINGS

IN ORDER TO PROVIDE FLEXIBILITY OF OPERATION WHILE TROUBLE-SHOOTING THE FOLLOWING OPERATIONAL SWITCHES ARE PROVIDED. IT SHOULD BE NOTED THAT THE PDP 11/60 HAS KEYPAD SWITCHES AND NON-MECHANICAL SWITCH REGISTER. IN ORDER TO SELECT A CERTAIN SWITCH OPTION, THE OCTAL EQUIVALENT OF THE SWITCH SHOULD BE KEYED IN AND THEN THE "LSWR" SWITCH DEPRESSED. COMBINATION OF SWITCH OPTIONS MAY BE SELECTED BY JUST "OR"ING TOGETHER THE INDIVIDUAL OCTAL EQUIVALENTS. THUS, IF SWITCH 15, 14 ARE TO BE SELECTED, THEN LOAD THE SWITCH REGISTER WITH OCTAL 140000.

SW <15> = 1    OCTAL 100000    HALT ON ERROR

THE PROGRAM WILL HALT WHEN AN ERROR OCCURS

SW <14> = 1    OCTAL 040000    LOOP ON TEST

IF THIS SWITCH IS SET WHEN AN ERROR OCCURS, THE PROGRAM WILL LOOP ON THE TEST HAVING THE ERROR.

SWITCH <13> = 1    OCTAL 020000    INHIBIT ERROR TYPEOUTS

SW <11> = 1    OCTAL 004000    INHIBIT ITERATIONS

THIS SWITCH INHIBITS TEST ITERATIONS. NOTE THAT THE FIRST PASS IS A QUICK-VERIFY PASS AND HENCE HAS NO TEXT ITERATIONS.

SW <10> = 1    OCTAL 002000    RING BELL ON ERROR

SW <9> = 1    OCTAL 001000    LOOP ON ERROR

WHEN AN ERROR OCCURS LOOPING IS DONE UPTO THE POINT OF ERROR (WHERE THE ERROR CALL IS LOCATED).



SW <8> = 1    OCTAL 000400    LOOP ON TEST IN SW <7: 0>

THE TEST SELECTED IN SW<7: 0> IS LOOPED ON.

SW <7: 0>

TEST NUMBER FOR LOOPING WHEN SW <8> = 1

#### 6.0 ERROR REPORTING

WHEN THE DIAGNOSTIC PROGRAM DETECTS A FAULT, AN ERROR IS REPORTED IN A STANDARD FORMAT AS FOLLOWING:

ERROR MESSAGE  
DATA HEADER  
ERROR DATA

THE ERROR MESSAGE DESCRIBES THE NATURE OF THE PROBLEM. THE DATA HEADER DEFINES THE VARIOUS ITEMS PRINTED OUT AS A PART OF THE ERROR DATA, RELEVANT TO THE ERROR. THE ERROR MESSAGE ALWAYS CONTAINS THE "PC", THE PROGRAM LOCATION WHERE THE ERROR CALL WAS MADE! THE ERROR "PC" IS THE CROSS-REFERENCE TO THE LISTINGS. IF THE USER WANTS MORE INFORMATION ON THE ERROR OR THE DESCRIPTION OF THE TEST THAT WAS BEING DONE, HE SHOULD REFER TO THE ERROR "PC" IN THE LISTING. WHEREVER NECESSARY, AN EXTENSIVE EXPLANATION IS GIVEN BESIDE THE ERROR CALL AND A PROBABLE FAULT DIAGNOSIS IS INCLUDED.

AT THE BEGINNING OF EACH TEST, A DESCRIPTION OF THE TEST IS GIVEN ALONG WITH THE LOGICAL FUNCTIONS OR PORTIONS IT IS TESTING. THE USER IS ADVISED TO REFERENCE THIS, IN CASE HE NEEDS MORE INFORMATION ON THE NATURE OF THE PROBLEM.

#### 7.0 PROGRAM RECOVERY



### 7.1 POWER-FAIL RECOVERY

IF A POWER-FAIL OCCURS WHILE RUNNING, THE PROGRAM SAVES THE NEEDED INFORMATION TO RESTART. ON SUBSEQUENT POWER-UP, THE PROGRAM PRINTS OUT "POWER" INDICATING THERE WAS A POWER FAILURE AND RESTARTS AUTOMATICALLY. IT SHOULD BE NOTED THAT THE BOOT/RUN/HALT SWITCH SHOULD BE IN THE "RUN" POSITION FOR AUTOMATIC RESTART AFTER POWER-FAIL.

### 7.2 SPURIOUS ERRORS

THE PROGRAM IS CAPABLE OF HANDLING TWO TYPES OF SPURIOUS ERRORS AND RECOVERING FROM THEM. CPU ERRORS RESULTING IN TRAPS TO VECTOR 4 ARE SERVICED BY THE CPU ERROR HANDLER. THE "PC" AT THE TIME OF THE TRAP IS PRINTED OUT, THE CONTENTS OF THE CPU ERROR REGISTER ARE ALSO GIVEN. THE CAUSE OF THE TRAP CAN BE FOUND OUT BY INSPECTING THE CPU ERROR REGISTER CONTENTS. RETURN IS MADE TO THE TEST NEXT TO THE ONE IN WHICH THE CPU ERROR OCCURED. A MESSAGE INDICATING THAT THE TEST WAS CORRECTED IS ALSO REPORTED.

PARITY ERRORS RESULTING IN TRAPS TO 114 ARE SERVICED BY THE UNEXPECTED PARITY ERROR HANDLER. THE SPURIOUS PARITY ERROR COULD HAVE OCCURED IN THE CACHE MEMORY, MAIN MEMORY OR THE WCS. IF THERE WAS A CACHE PARITY ERROR A SIMPLE RETURN (FROM INTERRUPT) IS MADE TO CONTINUE WITH THE TEST THAT WAS IN PROGRESS AT THE TIME OF THE ERROR. HOWEVER, IF THE PARITY ERROR WAS FROM THE MEMORY OR WCS THE PC AT THE TIME OF THE PARITY ERROR IS REPORTED AND CONTROL IS TRANSFERRED TO THE TEST NEXT TO THE ONE IN WHICH THE PARITY ERROR OCCURED. A MESSAGE INDICATING THAT THE TEST WAS ABORTED IS ALSO REPORTED.

### 8.0 XXDP AND ACT/APT COMPATABILITY

THE WCS DIAGNOSTIC PROGRAM CAN BE RUN UNDER THE XXDP SYSTEM. NORMAL LOADING AND STARTING PROCEDURES SHOULD BE FOLLOWED. THE PROGRAM IS COMPATIBLE WITH ACT AND APT SYSTEMS.



## 9.0 SUBROUTINE ABSTRACTS

### 9.1 SPURIOUS ERROR HANDLERS

THERE ARE TWO ROUTINES WHICH ARE CALLED BY UNEXPECTED TRAPS TO EITHER VECTOR 4 (CPU ERROR) OR VECTOR 114 (PARITY ERROR). THE CPU ERROR HANDLER "BADTMO" TYPES OUT THE PC AT THE TIME OF TRAP, AND SKIPS TO THE NEXT TEST. THE PARITY ERROR HANDLER "BADPAR" CHECKS IF THE PARITY ERROR WAS FROM THE MAIN MEMORY, CACHE MEMORY OR WCS. IF THE PARITY ERROR WAS FROM THE MAIN MEMORY OR WCS, THE PC AT THE TIME OF TRAP IS REPORTED AND CONTROL IS PASSED TO THE NEXT TEST.

### 9.2 SCOPE HANDLER

THIS ROUTINE IS CALLED (IRA AN IOT INSTRUCTION) AT THE BEGINNING OF ALL TESTS. IT CONTROLS THE OPERATIONAL FUNCTIONS OF LOOPING ON TEST, ITERATION, ETC.

### 9.3 ERROR HANDLER

THIS ROUTINE IS CALLED (ORA AN EMT) TO TYPE OUT AN ERROR REPORT. IT CONTROLS THE OPERATIONAL FUNCTIONS OF HALTING ON ERROR, INHIBITING ERROR PRINTOUT, LOOPING ON ERROR, BELL ON ERROR, ETC.

### 9.4 TRAPCATCHER

THIS CONSISTS OF A ".+2", "HALT" SEQUENCE REPEATED IN THE LOWER PART OF THE MEMORY (VECTOR AREA) FOR THE PURPOSE OF CATCHING ANY SPURIOUS TRAPS. SUCH A TRAP WILL RESULT IN A HALT AT THE TRAP VECTOR PLUS TWO.



#### 9.5 POWER DOWN, POWER UP ROUTINE

THIS ROUTINE IS CALLED WHEN AN UNEXPECTED POWER DOWN OCCURS WHILE THE PROGRAM IS RUNNING. WHEN THE POWER IS RETURNED THE PROGRAM WILL RESTART AFTER TYPING OUT "POWER".

#### 9.6 LDWCS - WCS LOADER

THIS ROUTINE IS CALLED TO LOAD A SECTION OF MICRO-CODE FROM THE MEMORY INTO THE WCS. AFTER LOADING THE WCS THE LOADER CHECKS IF THE MICRO-CODE WAS LOADED CORRECTLY.

#### 9.7 MISCELLANEOUS ROUTINES

TYPE - TO TYPE OUT AN ASCII STRING  
TYPDC - TO TYPE OUT AN OCTAL NUMBER  
TYPDS - TO CONVERT BINARY TO DECIMAL AND TYPE OUT THE NUMBER  
TRAP - TRAP DECODER  
EOF - END OF PASS ROUTINE

EVERY ROUTINE THAT HAS BEEN USED IN THE PROGRAM, IS PRECEDED BY A BRIEF EXPLANATION OF WHAT THE ROUTINE DOES AND CALLING CONVENTIONS.



16	BASIC DEFINITIONS
127	TRAP CATCHER
136	STARTING ADDRESS(ES)
139	APT PARAMETER BLOCK
161	ACT11 HOOKS
171	COMMON TAGS
233	APT MAILBOX-ETABLE
301	ERROR POINTER TABLE
1287	PROGRAM STARTS HERE
1361	T1 CHECK THAT WCS REGISTERS CAN BE ADDRESSED
1402	T2 CHECK WCS STATUS REGISTER, !D BIT, WCS PRESENT BIT
1457	T3 CHECK STATUS REGISTER, SET 'WWP' & 'PARDIS' BITS
1523	T4 CHECK 'WRITE' AND 'READ' OF LOCATION 0 IN WCS, EVEN PATTERN
1630	T5 CHECK 'WRITE' AND 'READ' OF LOCATION 0 IN WCS, ODD PATTERN
1722	T6 CHECK WRITE PULSES WPO, WP1, WP2 USING WRITE/READ (LOC. 0,2000,4000)
1837	T7 CHECK PARITY GENERATOR FOR HI BYTE DATA
1924	T10 CHECK PARITY GENERATOR FOR LO BYTE DATA
2019	T11 CHECK THAT WRONG PARITY CAN BE GENERATED, WHEN WWP IS SET
2117	T12 CHECK TRAP ON PARITY ERROR, TRAP INHIBITING, AND PARITY ERROR CLEARING LOGIC
2351	T13 CHECK THAT BAD PARITY 'PAR0' CAN FORCE A TRAP
2461	T14 CHECK THAT BAD PARITY 'PAR1' CAN FORCE A TRAP
2570	T15 CHECK THAT BAD PARITY 'PAR2' CAN FORCE A TRAP
2678	T16 CHECK WCS DATA PATH: DBUF (15:0), BUS DIN (15:0), LOC 0
2755	T17 CHECK WCS DATA PATH: BUS DIN (31:16), LOC2000
2837	T20 CHECK WCS DATA PATH: BUS DIN (47:32), LOC 4000
2920	T21 CHECK THE THREE HI-BYTE PARITY CHECKERS
3096	T22 CHECK THE THREE LO-BYTE PARITY CHECKERS
3270	T23 CHECK ADDRESSING FOR DATA ARRAY (11:0), ADDRESS(9:0)
3425	T24 CHECK ADDRESSING FOR DATA ARRAY (23:12), ADDRESS(7:0)
3575	T25 CHECK ADDRESSING FOR DATA ARRAY (23:12), ADDRESS (9:8)
3723	T26 CHECK ADDRESSING FOR DATA ARRAY (35:24), ADDRESS (7:0)
3872	T27 CHECK ADDRESSING FOR DATA ARRAY (35:24), ADDRESS (9:8)
4020	T30 CHECK ADDRESSING FOR DATA ARRAY (47:36), ADDRESS (9:0)
4169	T31 CHECK ADDRESS REGISTER - ADREG(11:0)
4206	T32 TEST DATA ARRAY-MARCHING 1'S AND 0'S
4291	T33 TEST DATA ARRAY MARCHING 0'S AND 1'S
4376	T34 TEST PARITY BIT STORAGE CHIPS-MARCHING 1'S AND 0'S
4503	T35 TEST PARITY BIT STORAGE CHIPS, MARCHING 0'S AND 1'S
4629	T36 CHECK WCS WATCH-DOG TIMER
4670	T37 CHECK THE WCS INITIALIZATION BY BM MICRO-CODE
4738	---TEST OF WCS XFC OPCODE DISPATCH---
4745	T40 CHECK "OTHER USER" DISPATCH TO 6001, XFC=076000-076077
4844	T41 CHECK "OTHER USER" DISPATCH TO 6011, XFC=076100-076177
4943	T42 CHECK "OTHER USER" DISPATCH TO 6012, XFC=076200-076277
5037	T43 CHECK "OTHER USER" DISPATCH TO 6013, XFC=076300-076377
5131	T44 CHECK "OTHER USER" DISPATCH TO 6014, XFC=076400-076477
5230	T45 CHECK "OTHER USER" DISPATCH TO 6015, XFC=076500-076577
5329	T46 CHECK "USER" DISPATCH TO 6002, XFC=076700-076777
5425	T47 CHECK "ODD PC" DISPATCH TO WCS AT 6004
5518	T50 CHECK "MICRO-BREAK" DISPATCH TO WCS AT 6000
5614	T51 CHECK "FLAG(7) WCS SERVICE" DISPATCH TO WCS AT 6005
5704	T52 CHECK "RESERVED INSTRUCTION" DISPATCH TO 6003 IN WCS
5800	T53 CHECK THAT XFC TRAPS OUT AS RESVD-INST WHEN WCS IS DISABLED
5898	---CHECK OF TMS ROM FUNCTIONS---
5905	T54 CHECK BLOCK MOVE OF GR'S TO WCS
6042	T55 CHECK BLOCK MOVE FROM WCS TO GR'S



6208	T56	CHECK BLOCK MOVE OF FLOATING POINT REGISTERS TO WCS
6323	T57	CHECK BLOCK MOVE FROM WCS TO WFP REGISTERS
6487	T60	CHECK BLOCK MOVE OF C-SCRATCH PAD REGISTERS TO WCS
6605	T61	CHECK BLOCK MOVE FROM WCS TO C-SCRATCH PAD, CSP
6761	T62	CHECK BLOCK MOVE OF USER-SCRATCH PAD REGISTERS TO WCS
6873	T63	CHECK BLOCK MOVE FROM WCS TO USER-SCRATCH PAD, USP
7021	T64	CHECK BLOCK STORE OF LOAD-WRITE-2 TO WCS
7137	T65	CHECK BLOCK LOAD OF LOAD-READ-2 FROM WCS
7252	T66	CHECK BLOCK STORE OF SET-STORE, WRITE-2 TO WCS
7368	T67	CHECK BLOCK LOAD OF SET-LOAD, INC-READ-2 FROM WCS
7483	T70	CHECK BLOCK STORE OF WRITE-INDIRECT TO WCS
7601	T71	CHECK BLOCK LOAD OF READ-INDIRECT FROM WCS
7719	T72	CHECK BLOCK STORE OF WFP. AC 0: 1 TO WCS
7846	T73	CHECK BLOCK STORE OF WFP. AC 2: 3 TO WCS
7972	T74	CHECK BLOCK STORE OF WFP. AC 4: 5 TO WCS
8098	T75	CHECK BLOCK LOAD OF WFP. AC 0: 1 FROM WCS
8221	T76	CHECK BLOCK LOAD OF WFP. AC 2: 3 FROM WCS
8344	T77	CHECK BLOCK LOAD OF WFP. AC 4: 5 FROM WCS
8470	T100	CHECK BLOCK STORE OF ASPAD 00: 37 TO WCS
8647	T101	CHECK BLOCK STORE OF BSPAD 00: 37 TO WCS
8821	T102	CHECK BLOCK STORE OF CSPAD 17: 00 TO WCS
8976	T103	CHECK BLOCK LOAD OF ASPAD 00: 37 FROM WCS
9189	T104	CHECK BLOCK LOAD OF BSPAD 00: 37 FROM WCS
9406	T105	CHECK BLOCK LOAD OF CSPAD 17: 00 FROM WCS
9584	T106	CHECK BLOCK STORE OF A&BSP TEMPS TO WCS
9737	T107	CHECK BLOCK LOAD OF A&BSP TEMPS FROM WCS
9909	T110	... END ...
9922		END OF PASS ROUTINE
10415		SCOPE HANDLER ROUTINE
10479		ERROR HANDLER ROUTINE
10531		ERROR MESSAGE TIMEOUT ROUTINE
10578		BINARY TO OCTAL (ASCII) AND TYPE
10655		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10722		TYPE ROUTINE
10801		APT COMMUNICATIONS ROUTINE
10858		TRAP DECODER
10881		TRAP TABLE
10897		POWER DOWN AND UP ROUTINES
10947		MESSAGES
11051		ERROR MESSAGES
11984		ERROR DATA HEADERS
12151		ERROR DATA VECTORS
12168		MICRO-CODE



. ENABL ABS,AMA

. TITLE PDP-11/60 WCS DIAGNOSTIC  
; \*COPYRIGHT (C) 1977  
; \*DIGITAL EQUIPMENT CORP.  
; \*MAYNARD, MASS. 01754  
; \*  
; \*PROGRAM BY JIM KAPADIA  
; \*  
; \*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
; \*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.  
; \*

. SBTTL BASIC DEFINITIONS

; \*INITIAL ADDRESS OF THE STACK POINTER \*\*\* 1100 \*\*\*

001100

STACK= 1100

. EQUIV EMT,ERROR ; ; BASIC DEFINITION OF ERROR CALL

. EQUIV IOT,SCOPE ; ; BASIC DEFINITION OF SCOPE CALL

; \*MISCELLANEOUS DEFINITIONS

000011

HT= 11 ; ; CODE FOR HORIZONTAL TAB

000012

LF= 12 ; ; CODE FOR LINE FEED

000015

CR= 15 ; ; CODE FOR CARRIAGE RETURN

000200

CRLF= 200 ; ; CODE FOR CARRIAGE RETURN-LINE FEED

177776

PS= 177776 ; ; PROCESSOR STATUS WORD

177774

. EQUIV PS,PSW

177774

STKLMT= 177774 ; ; STACK LIMIT REGISTER

177772

PIRQ= 177772 ; ; PROGRAM INTERRUPT REQUEST REGISTER

177570

DSWR= 177570 ; ; HARDWARE SWITCH REGISTER

177570

DDISP= 177570 ; ; HARDWARE DISPLAY REGISTER

; \*GENERAL PURPOSE REGISTER DEFINITIONS

000000

R0= %0 ; ; GENERAL REGISTER

000001

R1= %1 ; ; GENERAL REGISTER

000002

R2= %2 ; ; GENERAL REGISTER

000003

R3= %3 ; ; GENERAL REGISTER

000004

R4= %4 ; ; GENERAL REGISTER

000005

R5= %5 ; ; GENERAL REGISTER

000006

R6= %6 ; ; GENERAL REGISTER

000007

R7= %7 ; ; GENERAL REGISTER

000006

SP= %6 ; ; STACK POINTER

000007

PC= %7 ; ; PROGRAM COUNTER

; \*PRIORITY LEVEL DEFINITIONS

000000

PR0= 0 ; ; PRIORITY LEVEL 0

000040

PR1= 40 ; ; PRIORITY LEVEL 1

000100

PR2= 100 ; ; PRIORITY LEVEL 2

000140

PR3= 140 ; ; PRIORITY LEVEL 3

000200

PR4= 200 ; ; PRIORITY LEVEL 4

000240

PR5= 240 ; ; PRIORITY LEVEL 5

000300

PR6= 300 ; ; PRIORITY LEVEL 6

000340

PR7= 340 ; ; PRIORITY LEVEL 7

; \*"SWITCH REGISTER" SWITCH DEFINITIONS

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



57	100000	SW15=	100000
58	040000	SW14=	40000
59	020000	SW13=	20000
60	010000	SW12=	10000
61	004000	SW11=	4000
62	002000	SW10=	2000
63	001000	SW09=	1000
64	000400	SW08=	400
65	000200	SW07=	200
66	000100	SW06=	100
67	000040	SW05=	40
68	000020	SW04=	20
69	000010	SW03=	10
70	000004	SW02=	4
71	000002	SW01=	2
72	000001	SW00=	1
73		.EQUIV	SW09, SW9
74		.EQUIV	SW08, SW8
75		.EQUIV	SW07, SW7
76		.EQUIV	SW06, SW6
77		.EQUIV	SW05, SW5
78		.EQUIV	SW04, SW4
79		.EQUIV	SW03, SW3
80		.EQUIV	SW02, SW2
81		.EQUIV	SW01, SW1
82		.EQUIV	SW00, SW0

:\*DATA BIT DEFINITIONS (BIT00 TO BIT15)

85	100000	BIT15=	100000
86	040000	BIT14=	40000
87	020000	BIT13=	20000
88	010000	BIT12=	10000
89	004000	BIT11=	4000
90	002000	BIT10=	2000
91	001000	BIT09=	1000
92	000400	BIT08=	400
93	000200	BIT07=	200
94	000100	BIT06=	100
95	000040	BIT05=	40
96	000020	BIT04=	20
97	000010	BIT03=	10
98	000004	BIT02=	4
99	000002	BIT01=	2
100	000001	BIT00=	1
101		.EQUIV	BIT09, BIT9
102		.EQUIV	BIT08, BIT8
103		.EQUIV	BIT07, BIT7
104		.EQUIV	BIT06, BIT6
105		.EQUIV	BIT05, BIT5
106		.EQUIV	BIT04, BIT4
107		.EQUIV	BIT03, BIT3
108		.EQUIV	BIT02, BIT2
109		.EQUIV	BIT01, BIT1
110		.EQUIV	BIT00, BIT0

:\*BASIC "CPU" TRAP VECTOR ADDRESSES

112



```
113      000004      ERRVEC= 4          ;; TIME OUT AND OTHER ERRORS
114      000010      RESVEC= 10         ;; RESERVED AND ILLEGAL INSTRUCTIONS
115      000014      TBITVEC=14        ;; "T" BIT
116      000014      TRTVEC= 14         ;; TRACE TRAP
117      000014      BPTVEC= 14         ;; BREAKPOINT TRAP (BPT)
118      000020      IOTVEC= 20         ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
119      000024      PWRVEC= 24         ;; POWER FAIL
120      000030      EMTVEC= 30         ;; EMULATOR TRAP (EMT) **ERROR**
121      000034      TRAPVEC=34        ;; "TRAP" TRAP
122      000060      TKVEC= 60          ;; TTY KEYBOARD VECTOR
123      000064      TPVEC= 64          ;; TTY PRINTER VECTOR
124      000240      PIRQVEC=240        ;; PROGRAM INTERRUPT REQUEST VECTOR
125
126      .SBTTL TRAP CATCHER
127
128      .=0
129      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ". +2,HALT"
130      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
131      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
132      .=174
133      000174      000000      DISPREG: .WORD 0          ;; SOFTWARE DISPLAY REGISTER
134      000176      000000      SWREG: .WORD 0           ;; SOFTWARE SWITCH REGISTER
135      .SBTTL STARTING ADDRESS(ES)
136      000200      000137      003450      JMP @#START ;; JUMP TO STARTING ADDRESS OF PROGRAM
137
138      .SBTTL APT PARAMETER BLOCK
139
140      ;*****
141      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
142      ;*****
143      000204      .SX=.          ;; SAVE CURRENT LOCATION
144      000024      .=24          ;; SET POWER FAIL TO POINT TO START OF PROGRAM
145      000024      000200      200          ;; FOR APT START UP
146      000044      .=44          ;; POINT TO APT INDIRECT ADDRESS PNTR.
147      000044      000204      $APTHDR    ;; POINT TO APT HEADER BLOCK
148      000204      . .SX        ;; RESET LOCATION COUNTER
149      ;*****
150      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
151      ;INTERFACE SPEC.
152
153      000204      $APTHD:
154      000204      000000      $HIBTS: .WORD 0          ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
155      000206      001236      $MBADR: .WORD $MAIL      ;; ADDRESS OF APT MAILBOX (BITS 0-15)
156      000210      000036      $TSTM: .WORD 30.         ;; RUN TIM OF LONGEST TEST
157      000212      000024      $PASTM: .WORD 20.        ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
158      000214      000000      $UNITM: .WORD 0          ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
159      000216      000052      .WORD $ETEND-$MAIL/2    ;; LENGTH MAILBOX-ETABLE (WORDS)
160      .SBTTL ACT11 HOOKS
161
162      ;*****
163      ;HOOKS REQUIRED BY ACT11
164      000220      $SVPC=.          ;; SAVE PC
165      000046      .=46
166      000046      036514      $ENDAD          ;; 1)SET LOC. 46 TO ADDRESS OF SENDAD IN .SEOP
167      000052      .=52
168      000052      000000      .WORD 0          ;; 2)SET LOC. 52 TO ZERO
```



PDP-11/60 WCS DIAGNOSTIC  
DQKURR.P11 25-OCT-77 20:03

MACY11 30(1046) 25-OCT-77 20:12 PAGE 4  
ACT11 HOOKS

E 2

SEQ 0006  
SEQ 0017

169

000220

=\$SVPC

:: RESTORE PC



```

170          .SBTTL COMMON TAGS
171
172          ;*****
173          ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
174          ;*USED IN THE PROGRAM.
175
176          001100          =1100
177          001100          $CMTAG:          ;; START OF COMMON TAGS
178          001100          000000          .WORD          0          ;; CONTAINS THE TEST NUMBER
179          001102          000          $STNM:          .BYTE          0          ;; CONTAINS ERROR FLAG
180          001103          000          $ERFLG:          .BYTE          0          ;; CONTAINS SUBTEST ITERATION COUNT
181          001104          000000          $ICNT:          .WORD          0          ;; CONTAINS SCOPE LOOP ADDRESS
182          001106          000000          $LPADR:          .WORD          0          ;; CONTAINS SCOPE RETURN FOR ERRORS
183          001110          000000          $LPERR:          .WORD          0          ;; CONTAINS TOTAL ERRORS DETECTED
184          001112          000000          $ERTTL:          .WORD          0          ;; CONTAINS ITEM CONTROL BYTE
185          001114          000          $ITEMB:          .BYTE          0          ;; CONTAINS MAX. ERRORS PER TEST
186          001115          001          $ERMAX:          .BYTE          1          ;; CONTAINS PC OF LAST ERROR INSTRUCTION
187          001116          000000          $ERRPC:          .WORD          0          ;; CONTAINS ADDRESS OF 'GOOD' DATA
188          001120          000000          $GDADR:          .WORD          0          ;; CONTAINS ADDRESS OF 'BAD' DATA
189          001122          000000          $BDADR:          .WORD          0          ;; CONTAINS 'GOOD' DATA
190          001124          000000          $GDDAT:          .WORD          0          ;; CONTAINS 'BAD' DATA
191          001126          000000          $BDDAT:          .WORD          0          ;; RESERVED--NOT TO BE USED
192          001130          000000          .WORD          0
193          001132          000000          .WORD          0
194          001134          000          $AUTOB:          .BYTE          0          ;; AUTOMATIC MODE INDICATOR
195          001135          000          $INTAG:          .BYTE          0          ;; INTERRUPT MODE INDICATOR
196          001136          000000          .WORD          0
197          001140          177570          $SWR:          .WORD          DSWR          ;; ADDRESS OF SWITCH REGISTER
198          001142          177570          $DISPLAY:          .WORD          DDISP          ;; ADDRESS OF DISPLAY REGISTER
199          001144          177560          $TKS:          177560          ;; TTY KBD STATUS
200          001146          177562          $TKB:          177562          ;; TTY KBD BUFFER
201          001150          177564          $TPS:          177564          ;; TTY PRINTER STATUS REG. ADDRESS
202          001152          177566          $TPB:          177566          ;; TTY PRINTER BUFFER REG. ADDRESS
203          001154          000          $NULL:          .BYTE          0          ;; CONTAINS NULL CHARACTER FOR FILLS
204          001155          002          $FILLS:          .BYTE          2          ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
205          001156          012          $FILLC:          .BYTE          12          ;; INSERT FILL CHARS. AFTER A "LINE FEED"
206          001157          000          $TPFLG:          .BYTE          0          ;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
207          001160          000000          $REGAD:          .WORD          0          ;; CONTAINS THE ADDRESS FROM
208          ;; WHICH ($REG0) WAS OBTAINED
209          001162          000000          $REG0:          .WORD          0          ;; CONTAINS (($REGAD)+0)
210          001164          000000          $REG1:          .WORD          0          ;; CONTAINS (($REGAD)+2)
211          001166          000000          $REG2:          .WORD          0          ;; CONTAINS (($REGAD)+4)
212          001170          000000          $REG3:          .WORD          0          ;; CONTAINS (($REGAD)+6)
213          001172          000000          $REG4:          .WORD          0          ;; CONTAINS (($REGAD)+10)
214          001174          000000          $REG5:          .WORD          0          ;; CONTAINS (($REGAD)+12)
215          001176          000000          $REG6:          .WORD          0          ;; CONTAINS (($REGAD)+14)
216          001200          000000          $REG7:          .WORD          0          ;; CONTAINS (($REGAD)+16)
217          001202          000000          $TMP0:          .WORD          0          ;; USER DEFINED
218          001204          000000          $TMP1:          .WORD          0          ;; USER DEFINED
219          001206          000000          $TMP2:          .WORD          0          ;; USER DEFINED
220          001210          000000          $TMP3:          .WORD          0          ;; USER DEFINED
221          001212          000000          $TMP4:          .WORD          0          ;; USER DEFINED
222          001214          000000          $TMP5:          .WORD          0          ;; USER DEFINED
223          001216          000000          $TMP6:          .WORD          0          ;; USER DEFINED
224          001220          000000          $TMP7:          .WORD          0          ;; USER DEFINED
225          001222          000000          $TIMES:          0          ;; MAX. NUMBER OF ITERATIONS

```



```
226 001224 000000 $ESCAPE: 0 ;: ESCAPE ON ERROR ADDRESS
227 001226 177607 000377 $BELL: . ASCII <207><377><377> ;: CODE FOR BELL
228 001232 077 $QUES: . ASCII /?/ ;: QUESTION MARK
229 001233 015 $CRLF: . ASCII <15> ;: CARRIAGE RETURN
230 001234 000012 $LF: . ASCII <12> ;: LINE FEED
231 ;: *****
232 .SBTTL APT MAILBOX-ETABLE
233 ;: *****
234 ;: *****
235 .EVEN
236 001236 $MAIL: ;: APT MAILBOX
237 001236 000000 $MSGTY: . WORD AMSGTY ;: MESSAGE TYPE CODE
238 001240 000000 $FATAL: . WORD AFATAL ;: FATAL ERROR NUMBER
239 001242 000000 $TESTN: . WORD ATESTN ;: TEST NUMBER
240 001244 000000 $PASS: . WORD APASS ;: PASS COUNT
241 001246 000000 $DEVCT: . WORD ADEVCT ;: DEVICE COUNT
242 001250 000000 $UNIT: . WORD AUNIT ;: I/O UNIT NUMBER
243 001252 000000 $MSGAD: . WORD AMSGAD ;: MESSAGE ADDRESS
244 001254 000000 $MSGLG: . WORD AMSGLG ;: MESSAGE LENGTH
245 001256 $ETABLE: ;: APT ENVIRONMENT TABLE
246 001256 000 $ENV: . BYTE AENV ;: ENVIRONMENT BYTE
247 001257 000 $ENVM: . BYTE AENVM ;: ENVIRONMENT MODE BITS
248 001260 000000 $SWREG: . WORD ASWREG ;: APT SWITCH REGISTER
249 001262 000000 $USWR: . WORD AUSWR ;: USER SWITCHES
250 001264 000000 $CPUOP: . WORD ACPUOP ;: CPU TYPE, OPTIONS
251 ;: *
252 ;: * 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
253 ;: * 11/70=06, PDQ=07, Q=10
254 ;: * BIT 10=REAL TIME CLOCK
255 ;: * BIT 9=FLOATING POINT PROCESSOR
256 ;: * BIT 8=MEMORY MANAGEMENT
257 001266 000 $MAMS1: . BYTE AMAMS1 ;: HIGH ADDRESS, M. S. BYTE
258 001267 000 $MTYP1: . BYTE AMTYP1 ;: MEM. TYPE, BLK#1
259 ;: * MEM. TYPE BYTE -- (HIGH BYTE)
260 ;: * 900 NSEC CORE=001
261 ;: * 300 NSEC BIPOLAR=002
262 ;: * 500 NSEC MOS=003
263 001270 000000 $MADR1: . WORD AMADR1 ;: HIGH ADDRESS, BLK#1
264 ;: * MEM. LAST ADDR. =3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
265 001272 000 $MAMS2: . BYTE AMAMS2 ;: HIGH ADDRESS, M. S. BYTE
266 001273 000 $MTYP2: . BYTE AMTYP2 ;: MEM. TYPE, BLK#2
267 001274 000000 $MADR2: . WORD AMADR2 ;: MEM. LAST ADDRESS, BLK#2
268 001276 000 $MAMS3: . BYTE AMAMS3 ;: HIGH ADDRESS, M. S. BYTE
269 001277 000 $MTYP3: . BYTE AMTYP3 ;: MEM. TYPE, BLK#3
270 001300 000000 $MADR3: . WORD AMADR3 ;: MEM. LAST ADDRESS, BLK#3
271 001302 000 $MAMS4: . BYTE AMAMS4 ;: HIGH ADDRESS, M. S. BYTE
272 001303 000 $MTYP4: . BYTE AMTYP4 ;: MEM. TYPE, BLK#4
273 001304 000000 $MADR4: . WORD AMADR4 ;: MEM. LAST ADDRESS, BLK#4
274 001306 000000 $VECT1: . WORD AVECT1 ;: INTERRUPT VECTOR#1, BUS PRIORITY#1
275 001310 000000 $VECT2: . WORD AVECT2 ;: INTERRUPT VECTOR#2, BUS PRIORITY#2
276 001312 000000 $BASE: . WORD ABASE ;: BASE ADDRESS OF EQUIPMENT UNDER TEST
277 001314 000000 $DEVN: . WORD ADEVN ;: DEVICE MAP
278 001316 000000 $CDW1: . WORD ACDW1 ;: CONTROLLER DESCRIPTION WORD#1
279 001320 000000 $CDW2: . WORD ACDW2 ;: CONTROLLER DESCRIPTION WORD#2
280 001322 000000 $DDW0: . WORD ADDW0 ;: DEVICE DESCRIPTOR WORD#0
281 001324 000000 $DDW1: . WORD ADDW1 ;: DEVICE DESCRIPTOR WORD#1
```



282	001326	000000	\$DDW2:	. WORD	ADDW2	:: DEVICE	DESCRIPTOR	WORD#2
283	001330	000000	\$DDW3:	. WORD	ADDW3	:: DEVICE	DESCRIPTOR	WORD#3
284	001332	000000	\$DDW4:	. WORD	ADDW4	:: DEVICE	DESCRIPTOR	WORD#4
285	001334	000000	\$DDW5:	. WORD	ADDW5	:: DEVICE	DESCRIPTOR	WORD#5
286	001336	000000	\$DDW6:	. WORD	ADDW6	:: DEVICE	DESCRIPTOR	WORD#6
287	001340	000000	\$DDW7:	. WORD	ADDW7	:: DEVICE	DESCRIPTOR	WORD#7
288	001342	000000	\$DDW8:	. WORD	ADDW8	:: DEVICE	DESCRIPTOR	WORD#8
289	001344	000000	\$DDW9:	. WORD	ADDW9	:: DEVICE	DESCRIPTOR	WORD#9
290	001346	000000	\$DDW10:	. WORD	ADDW10	:: DEVICE	DESCRIPTOR	WORD#10
291	001350	000000	\$DDW11:	. WORD	ADDW11	:: DEVICE	DESCRIPTOR	WORD#11
292	001352	000000	\$DDW12:	. WORD	ADDW12	:: DEVICE	DESCRIPTOR	WORD#12
293	001354	000000	\$DDW13:	. WORD	ADDW13	:: DEVICE	DESCRIPTOR	WORD#13
294	001356	000000	\$DDW14:	. WORD	ADDW14	:: DEVICE	DESCRIPTOR	WORD#14
295	001360	000000	\$DDW15:	. WORD	ADDW15	:: DEVICE	DESCRIPTOR	WORD#15
296								
297								
298	001360		SETEND:					
299								



```
300 .SBTTL ERROR POINTER TABLE
301
302 ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
303 ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
304 ;*LOCATION $ITEMB THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
305 ;*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
306 ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
307
308 ;* EM ;:POINTS TO THE ERROR MESSAGE
309 ;* DH ;:POINTS TO THE DATA HEADER
310 ;* DT ;:POINTS TO THE DATA
311 ;* DF ;:POINTS TO THE DATA FORMAT
312
313
314 001362 $ERRTB:
315
316 ;ERROR TABLE
317
318 ;ITEM 1
319
320 001362 043240 EM1 ;TIMEOUT ON REFERENCING WCS REGISTER
321 001364 055536 DH1 ;PC REG ADRS
322 001366 057320 DT1 ;$ERRPC $REGO
323 001370 000000 0
324
325 ;ITEM 2
326
327 001372 043304 EM2 ;'WCS PRESENT' BIT NOT SET IN WHAMI
328 001374 055557 DH2 ;PC WHAMI
329 001376 057320 DT1 ;$ERRPC $REGO
330 001400 000000 0
331
332 ;ITEM 3
333
334 001402 043347 EM3 ;WCS ID CODE (BIT 14) NOT SET
335 001404 055575 DH3 ;PC STATUS
336 001406 057320 DT1 ;$ERRPC $REGO
337 001410 000000 0
338
339 ;ITEM 4
340
341 001412 043402 EM4 ;'PARITY DISABLE' BIT NOT CLEAR
342 001414 055575 DH3 ;PC STATUS
343 001416 057320 DT1 ;$ERRPC $REGO
344 001420 000000 0
345
346 ;ITEM 5
347 001422 043441 EM5 ;'WRITE WRONG PARITY' BIT NOT CLEAR
348 001424 055575 DH3 ;PC STATUS
349 001426 057320 DT1 ;$ERRPC $REGO
350 001430 000000 0
351
352 ;ITEM 6
353
354 001432 043504 EM6 ;'PARITY DISABLE' BIT NOT SET
355 001434 055575 DH3 ;PC STATUS
```



356	001436	057320	DT1	;SERRPC \$REG0
357	001440	000000	0	
358				
359			; ITEM 7	
360				
361	001442	043541	EM7	; 'PARITY ERROR' NOT CLEAR
362	001444	055575	DH3	; PC STATUS
363	001446	057320	DT1	;SERRPC \$REG0
364	001450	000000	0	
365				
366			; ITEM 10	
367				
368	001452	043572	EM10	; 'WRITE WRONG PARITY' BIT NOT SET
369	001454	055575	DH3	; PC STATUS
370	001456	057320	DT1	;SERRPC \$REG0
371	001460	000000	0	
372				
373			; ITEM 11	
374				
375	001462	043633	EM11	; STATUS ERROR ON WRITE TO WCS LOCATION
376	001464	055614	DH11	; PC EXPCT RECVD LOC PATRN
377	001466	057326	DT11	;SERRPC \$REG0 \$REG1 \$REG2 \$REG3
378	001470	000000	0	
379				
380			; ITEM 12	
381				
382	001472	043701	EM12	; DATA ERROR ON READ FROM WCS LOCATION
383	001474	055663	DH12	; PC LOC EXPCT RECVD
384	001476	057342	DT12	;SERRPC \$REG0 \$REG1 \$REG2
385	001500	000000	0	
386				
387			; ITEM 13	
388				
389	001502	043746	EM13	; STATUS ERROR ON READ FROM WCS LOCATION
390	001504	055614	DH11	; PC EXPCT RECVD LOC PATRN
391	001506	057326	DT11	;SERRPC \$REG0 \$REG1 \$REG2 \$REG3
392	001510	000000	0	
393				
394			; ITEM 14	
395				
396	001512	044015	EM14	; TRAP AT LOCATION
397	001514	055722	DH14	; PC TRPLOC
398	001516	057320	DT1	;SERRPC \$REG0
399	001520	000000	0	
400				
401			; ITEM 15	
402				
403	001522	044036	EM15	; INCORRECT PARITY BIT GENERATED FOR PATTERN
404	001524	055741	DH15	; PC PATRN EXPCTD PAR
405	001526	057354	DT15	;SERRPC \$REG0 \$REG1
406	001530	000000	0	
407				
408			; ITEM 16	
409				
410	001532	044111	EM16	; INCORRECT PARITY BIT GENERATED -BIT7- (WITH WWP) FOR PA TTERN

Line	PC	STATUS	PATR	REG	Description
411	001534	056000	DH17		
412	001536	057354	DT15		
413	001540	000000	0		
414					
415					: ITEM 17
416					
417	001542	044204	EM17		: UNEXPECTED PARITY TRAP ON READING BAD PARITY THOUGH 'PARDIS' WAS SET
418	001544	056000	DH17		
419	001546	057354	DT15		
420	001550	000000	0		
421					
422					: ITEM 20
423					
424	001552	044311	EM20		: 'PARITY ERROR' DID NOT SET ON PARITY ERROR TRAP
425	001554	055575	DH3		
426	001556	057320	DT1		
427	001560	000000	0		
428					
429					: ITEM 21
430					
431	001562	044371	EM21		: INCORRECT PARITY BIT STORED
432	001564	056027	DH21		
433	001566	057342	DT12		
434	001570	000000	0		
435					
436					: ITEM 22
437					
438	001572	044425	EM22		: NO PARITY ERROR TRAP ON READING WRONG PARITY
439	001574	056027	DH21		
440	001576	057342	DT12		
441	001600	000000	0		
442					
443					: ITEM 23
444					
445	001602	044502	EM23		: 'PARITY ERROR' BIT NOT SET ON READING BAD PARITY
446	001604	055575	DH3		
447	001606	057320	DT1		
448	001610	000000	0		
449					
450					: ITEM 24
451					
452	001612	044563	EM24		: SETTING 'PARDIS' DID NOT CLEAR 'PAREPR'
453	001614	055575	DH3		
454	001616	057320	DT1		
455	001620	000000	0		
456					
457					: ITEM 25
458					
459	001622	044633	EM25		: BAD PARITY NOT DETECTED BY CHECKER
460	001624	056027	DH21		
461	001626	057342	DT12		
462	001630	000000	0		
463					
464					: ITEM 26
465					



466	001632	044676	EM26	; PARITY ERROR ON READING WCS
467	001634	056075	DH26	; PC LOC EXPCT STATUS
468	001636	057342	DT12	; SERRPC \$REGO \$REG1 \$REG2
469	001640	000000	0	
470				
471			; ITEM 27	
472				
473	001642	044732	EM27	; TRAP AT LOCATION, ON DOING READ
474	001644	055722	DH14	; PC TRPLOC
475	001646	057320	DT1	; SERRPC \$REGO
476	001650	000000	0	
477				
478			; ITEM 30	
479				
480	001652	044772	EM30	; CORRECT XFC CODE WAS NOT SAVED ON "OTHER USER" DISPATCH OF XFC
481	001654	056134	DH30	; PC EXPCTD XFC RECVD
482	001656	057354	DT15	; SERRPC \$REGO \$REG1
483	001660	000000	0	
484				
485			; ITEM 31	
486				
487	001662	045074	EM31	; WATCH-DOG TIMED OUT ON "OTHER USER" DISPATCH OF XFC
488	001664	056200	DH31	; PC XFC CODE
489	001666	057320	DT1	; SERRPC \$REGO
490	001670	000000	0	
491				
492			; ITEM 32	
493				
494	001672	045163	EM32	; CORRECT XFC CODE WAS NOT SAVED ON "USER" DISPATCH OF XFC
495	001674	056134	DH30	; PC EXPCTD XFC RECVD
496	001676	057354	DT15	; SERRPC \$REGO \$REG1
497	001700	000000	0	
498				
499			; ITEM 33	
500				
501	001702	045257	EM33	; WATCH-DOG TIMED OUT ON "USER" DISPATCH OF XFC
502	001704	056200	DH31	; PC XFC CODE
503	001706	057320	DT1	; SERRPC \$REGO
504	001710	000000	0	
505				
506			; ITEM 34	
507				
508	001712	045340	EM34	; FLAG NOT =1 UPON ENTERING WCS VIA "ODD PC" DISPATCH
509	001714	056221	DH34	; PC R3-FLAG
510	001716	057320	DT1	; SERRPC \$REGO
511	001720	000000	0	
512				
513			; ITEM 35	
514				
515	001722	045425	EM35	; WATCH-DOG TIMED OUT ON "ODD PC" DISPATCH TO WCS
516	001724	056066	DH23	; PC
517	001726	057364	DT23	; SERRPC
518	001730	000000	0	
519				
520			; ITEM 36	
521				

522	001732	045510	EM36	; FLAG NOT =1 UPON ENTERING WCS VIA "MICROBREAK" DISPATCH			
523	001734	056221	DH34	; PC R3-FLAG			
524	001736	057320	DT1	; SERRPC \$REG0			
525	001740	000000	0				
526							
527			; ITEM	37			
528							
529	001742	045602	EM37	; WATCH-DOG TIMED OUT ON "MICRO-BREAK" DISPATCH TO WCS			
530	001744	056066	DH23	; PC			
531	001746	057364	DT23	; SERRPC			
532	001750	000000	0				
533							
534			; ITEM	40			
535							
536	001752	045672	EM40	; CORRECT DATA NOT STORED ON BLOCK MOVE OF GR'S TO WCS			
537	001754	056241	DH40	; PC WCS LOC EXPCT RECVD GR#			
538	001756	057326	DT40	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3			
539	001760	000000	0				
540							
541			; ITEM	41			
542							
543	001762	045757	EM41	; WATCH-DOG TIMED OUT ON STORING GR'S IN WCS			
544	001764	056200	DH31	; PC XFC CODE			
545	001766	057320	DT1	; SERRPC \$REG0			
546	001770	000000	0				
547							
548			; ITEM	42			
549							
550	001772	046032	EM42	; CORRECT DATA NOT STORED ON LOAD OF GR'S FROM WCS			
551	001774	056310	DH42	; PC GR# WCS-LOC EXPCT RECVD			
552	001776	057326	DT40	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3			
553	002000	000000	0				
554							
555			; ITEM	43			
556							
557	002002	046113	EM43	; WATCH-DOG TIMED OUT ON LOADING OF GR'S FROM WCS			
558	002004	056200	DH31	; PC XFC CODE			
559	002006	057320	DT1	; SERRPC \$REG0			
560	002010	000000	0				
561							
562			; ITEM	44			
563							
564	002012	046173	EM44	; CORRECT RESEVED INST. CODE NOT SAVED ON "RESVD INSTR" DISPATCH TO WCS			
565	002014	056357	DH44	; PC EXPCT RECVD			
566	002016	057354	DT15	; SERRPC \$REG0 \$REG1			
567	002020	000000	0				
568							
569			; ITEM	45			
570							
571	002022	046305	EM45	; WATCH-DOG TIMED OUT ON "RESVD INST" DISPATCH TO WCS			
572	002024	056066	DH23	; PC			
573	002026	057364	DT23	; SERRPC			
574	002030	000000	0				
575							
576			; ITEM	46			
577							



578	002032	046374	EM46	;CORRECT DATA WAS NOT STORED ON BLOCK MOVE OF FP REGISTERS TO WCS				
579	002034	056406	DH46	;PC	WCS LOC	EXPT	RECVD	FP REG#
580	002036	057326	DT40	;SERRPC	\$REG0	\$REG1	\$REG2	\$REG3
581	002040	000000	0					
582								
583			; ITEM	47				
584								
585	002042	046476	EM47	;WATCH-DOG TIMED OUT ON BLOCK MOVE OF FP REGISTERS TO WCS				
586	002044	056066	DH23	;PC				
587	002046	057364	DT23	;SERRPC				
588	002050	000000	0					
589								
590			; ITEM	50				
591								
592	002052	046570	EM50	;CORRECT DATA WAS NOT STORED ON BLOCK MOVE TO WFP REGISTERS FROM WCS				
593	002054	056455	DH50	;PC	WFP NO.	EXPT	RECVD	WCS LOC
594	002056	057326	DT40	;SERRPC	\$REG0	\$REG1	\$REG2	\$REG3
595	002060	000000	0					
596								
597			; ITEM	51				
598								
599	002062	046674	EM51	;WATCH DOG TIMED OUT ON BLOCK MOVE FROM WCS TO CSP				
600	002064	056066	DH23	;PC				
601	002066	057364	DT23	;SERRPC				
602	002070	000000	0					
603								
604			; ITEM	52				
605								
606	002072	046756	EM52	;CORRECT DATA WAS NOT STORED ON BLOCK MOVE OF CSP REGISTERS TO WCS				
607	002074	056524	DH52	;PC	WCS LOC	EXPT	RECVD	CSP NO.
608	002076	057326	DT40	;SERRPC	\$REG0	\$REG1	\$REG2	\$REG3
609	002100	000000	0					
610								
611			; ITEM	53				
612								
613	002102	047060	EM53	;WATCH-DOG TIMED OUT ON BLOCK MOVE OF CSP REGISTERS TO WCS				
614	002104	056066	DH23	;PC				
615	002106	057364	DT23	;SERRPC				
616	002110	000000	0					
617								
618			; ITEM	54				
619								
620	002112	047152	EM54	;CORRECT DATA WAS NOT STORED ON BLOCK MOVE FROM WCS TO CSP				
621	002114	056573	DH54	;PC	CSP NO.	EXPT	RECVD	WCS LOC
622	002116	057326	DT40	;SERRPC	\$REG0	\$REG1	\$REG2	\$REG3
623	002120	000000	0					
624								
625			; ITEM	55				
626								
627	002122	047244	EM55	;WATCH-DOG TIMED OUT ON BLOCK MOVE OF CSP REGISTERS TO WCS				
628	002124	056066	DH23	;PC				
629	002126	057364	DT23	;SERRPC				
630	002130	000000	0					
631								
632			; ITEM	56				
633								

634	002132	047336	EM56	:CORRECT DATA WAS NOT STORED ON BLOCK MOVE OF USER SP REGISTERS TO WCS
635	002134	056642	DH56	:PC WCS LOC EXPCT RECVD USP-MED-RD-CODE
636	002136	057326	DT40	:SERRPC \$REG0 \$REG1 \$REG2 \$REG3
637	002140	000000	0	
638				
639			:ITEM	57
640				
641	002142	047441	EM57	:WATCH DOG TIMED OUT ON BLOCK MOVE OF USP REGISTERS TO WCS
642	002144	056066	DH23	:PC
643	002146	057364	DT23	:SERRPC
644	002150	000000	0	
645				
646			:ITEM	60
647				
648	002152	047534	EM60	:CORRECT DATA WAS NOT LOADED ON BLOCK MOVE FROM WCS TO USER SP
649	002154	056642	DH56	:PC WCS LOC EXPCT RECVD USP-MED-RD-CODE
650	002156	057326	DT40	:SERRPC \$REG0 \$REG1 \$REG2 \$REG3
651	002160	000000	0	
652				
653			:ITEM	61
654				
655	002162	047641	EM61	:WATCH DOG TIMED OUT ON BLOCK MOVE FROM WCSS TO USER SP
656	002164	056066	DH23	:PC
657	002166	057364	DT23	:SERRPC
658	002170	000000	0	
659				
660			:ITEM	62
661				
662	002172	047736	EM62	:ODD-PC TRAP INSTEAD OF ODD-PC DISPATCH TO WCS
663	002174	056707	DH62	:PC ODD-PC-AT
664	002176	057320	DT1	:SERRPC \$REG0
665	002200	000000	0	
666				
667			:ITEM	63
668				
669	002202	050014	EM63	:RESERVED INSTR. TRAP INSTEAD OF RESERVED INST. DISPATCH TO WCS
670	002204	056066	DH23	:PC
671	002206	057364	DT23	:SERRPC
672	002210	000000	0	
673				
674			:ITEM	64
675				
676	002212	050110	EM64	:WCS WATCH-DOG TIMER DEFECTIVE-DID NOT TIME OUT
677	002214	056066	DH23	:PC
678	002216	057364	DT23	:SERRPC
679	002220	000000	0	
680				
681			:ITEM	65
682				
683	002222	050155	EM65	:WCS-WATCH-DOG TIMER DEFECTIVE-TIME NOT IN
684			:ACCEPTED RANGE	
685	002224	056731	DH65	:PC TIME
686	002226	057320	DT1	:SERRPC \$REG0
687	002230	000000	0	
688				
689			:ITEM	66



690					
691	002232	050233	EM66	;XFC OFCODE DID NOT TRAP AS RESERVED INSTR.	
692				;WHEN WCS WAS DISABLED	
693	002234	056200	DH31	;PC XFC CODE	
694	002236	057320	DT1	;SERRPC \$REGO	
695	002240	000000	0		
696					
697			; ITEM	67	
698					
699	002242	050332	EM67	;WCS INIT DID NOT CLEAR BAD PARITY IN WCS LOCATION	
700	002244	056746	DH67	;PC STATUS LOC	
701	002246	057354	DT15	;SERRPC \$REGO \$REG1	
702	002250	000000	0		
703					
704			; ITEM	70	
705					
706	002252	050410	EM70	;MICRO-WORD LOADED WRONGLY IN WCS AT LOC	
707	002254	055663	DH12	;PC LOC EXPCT RECVD	
708	002256	057342	DT12	;SERRPC \$REGO \$REG1 \$REG2	
709	002260	000000	0		
710					
711			; ITEM	71	
712					
713	002262	050460	EM71	;WRONG PARITY BIT-PAR0- (BIT1 IN STAT REG) READ FROM WCS LOC	
714	002264	056775	DH71	;PC WCSLOC STAT REG	
715	002266	057354	DT15	;SERRPC \$REGO \$REG1	
716	002270	000000	0		
717					
718			; ITEM	72	
719					
720	002272	050552	EM72	;WRONG PARITY BIT-PAR1- (BIT2 IN STAT REG) READ FROM WCS LOC	
721	002274	056775	DH71	;PC WCSLOC STAT REG	
722	002276	057354	DT15	;SERRPC \$REGO \$REG1	
723	002300	000000	0		
724					
725			; ITEM	73	
726					
727	002302	050644	EM73	;WRONG PARITY BIT-PAR2- (BIT3 IN STAT REG) READ FROM WCS LOC	
728	002304	056775	DH71	;PC WCSLOC STAT REG	
729	002306	057354	DT15	;SERRPC \$REGO \$REG1	
730	002310	000000	0		
731					
732			; ITEM	74	
733					
734	002312	050736	EM74	;FLAG NOT =2 UPON ENTERING WCS VIA "FLAG<7>-SVC" DISPATCH	
735	002314	056221	DH34	;PC R3-FLAG	
736	002316	057320	DT1	;SERRPC \$REGO	
737	002320	000000	0		
738					
739			; ITEM	75	
740					
741	002322	051030	EM75	;WATCH-DOG TIMED OUT ON "FLAG<7>-SVC" DISPATCH TO WCS	
742	002324	056066	DH23	;PC	
743	002326	057364	DT23	;SERRPC	
744	002330	000000	0		
745					

746			: ITEM 76
747			
748	002332	051120	EM76 : INCORRECT DATA STORED DURING 'LOAD WRITE 2' TMS FUNCTION
749	002334	057024	DH76 : PC GPR EXP'D-DATA RCV'D-DATA WCS-ADDR
750	002336	057326	DT76 : \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
751	002340	000000	0
752			
753			: ITEM 77
754			
755	002342	051211	EM77 : WATCH-DOG TIMEOUT DURING 'LOAD WRITE 2' TMS FUNCTION
756	002344	056066	DH23 : PC
757	002346	057364	DT23 : \$ERRPC
758	002350	000000	0
759			
760			: ITEM 100
761			
762	002352	051276	EM100 : INCORRECT DATA STORED DURING 'LOAD READ 2' TMS FUNCTION
763	002354	057073	DH100 : PC GPR EXP'D-DATA RCV'D-DATA
764	002356	057342	DT100 : \$ERRPC \$REG0 \$REG1 \$REG2
765	002360	000000	0
766			
767			: ITEM 101
768			
769	002362	051366	EM101 : WATCH-DOG TIMEOUT DURING 'LOAD READ 2' TMS FUNCTION
770	002364	056066	DH23 : PC
771	002366	057364	DT23 : \$ERRPC
772	002370	000000	0
773			
774			: ITEM 102
775			
776	002372	051452	EM102 : INCORRECT DATA STORED DURING 'SET STORE / WRITE 2' TMS FUNCTION
777	002374	057024	DH76 : PC GPR EXP'D-DATA RCV'D-DATA WCS-ADDR
778	002376	057326	DT76 : \$ERRPC \$REG0 \$REG1 \$REG2 \$REG3
779	002400	000000	0
780			
781			: ITEM 103
782			
783	002402	051552	EM103 : WATCH-DOG TIMEOUT DURING 'SET STORE / WRITE 2' TMS FUNCTION
784	002404	056066	DH23 : PC
785	002406	057364	DT23 : \$ERRPC
786	002410	000000	0
787			
788			: ITEM 104
789			
790	002412	051646	EM104 : INCORRECT DATA STORED DURING 'SET LOAD / INC READ 2' TMS FUNCTION
791	002414	057073	DH100 : PC GPR EXP'D-DATA RCV'D-DATA
792	002416	057342	DT100 : \$ERRPC \$REG0 \$REG1 \$REG2
793	002420	000000	0
794			
795			: ITEM 105
796			
797	002422	051750	EM105 : WATCH-DOG TIMEOUT DURING 'SET LOAD / INC READ 2' TMS FUNCTION
798	002424	056066	DH23 : PC
799	002426	057364	DT23 : \$ERRPC
800	002430	000000	0
801			





858			; ITEM	116	
859			EM116	; INCORRECT DATA STORED DURING 'STORE WFP<4:5>' TMS FUNCTION	
860	002532	052754	DH112	; PC MED WCSLOC E-DATA R-DATA	
861	002534	057250	DT112	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3	
862	002536	057326	0		
863	002540	000000			
864			; ITEM	117	
865			EM117	; WATCH-DOG TIMEOUT DURING 'STORE WFP<4:5>' TMS FUNCTION	
866			DH23	; PC	
867	002542	053047	DT23	; SERRPC	
868	002544	056066	0		
869	002546	057364			
870	002550	000000			
871			; ITEM	120	
872			EM120	; INCORRECT DATA STORED DURING 'LOAD WFP<0:1>' TMS FUNCTION	
873			DH120	; PC MED E-DATA R-DATA	
874	002552	053136	DT120	; SERRPC \$REG0 \$REG1 \$REG2	
875	002554	057211	0		
876	002556	057342			
877	002560	000000			
878			; ITEM	121	
879			EM121	; WATCH-DOG TIMEOUT DURING 'LOAD WFP<0:1>' TMS FUNCTION	
880			DH23	; PC	
881	002562	053230	DT23	; SERRPC	
882	002564	056066	0		
883	002566	057364			
884	002570	000000			
885			; ITEM	122	
886			EM122	; INCORRECT DATA STORED DURING 'LOAD WFP<2:3>' TMS FUNCTION	
887			DH120	; PC MED E-DATA R-DATA	
888	002572	053316	DT120	; SERRPC \$REG0 \$REG1 \$REG2	
889	002574	057211	0		
890	002576	057342			
891	002600	000000			
892			; ITEM	123	
893			EM123	; WATCH-DOG TIMEOUT DURING 'LOAD WFP<2:3>' TMS FUNCTION	
894			DH23	; PC	
895	002602	053410	DT23	; SERRPC	
896	002604	056066	0		
897	002606	057364			
898	002610	000000			
899			; ITEM	124	
900			EM124	; INCORRECT DATA STORED DURING 'LOAD WFP<4:5>' TMS FUNCTION	
901			DH120	; PC MED E-DATA R-DATA	
902	002612	053476	DT120	; SERRPC \$REG0 \$REG1 \$REG2	
903	002614	057211	0		
904	002616	057342			
905	002620	000000			
906			; ITEM	125	
907			EM125	; WATCH-DOG TIMEOUT DURING 'LOAD WFP<4:5>' TMS FUNCTION	
908			DH23	; PC	
909	002622	053570	DT23	; SERRPC	
910	002624	056066	0		
911	002626	057364			
912	002630	000000			
913					



914			; ITEM	126	
915					
916	002632	053656	EM126	; INCORRECT DATA STORED DURING 'STORE ASPAD 00:37 ' TMS FUNCTION	
917	002634	057250	DH126	; PC MED WCSLOC E-DATA R-DATA	
918	002636	057326	DT11	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3	
919	002640	000000	0		
920					
921			; ITEM	127	
922					
923	002642	053755	EM127	; WATCH-DOG TIMEOUT DURING 'STORE ASPAD 00:37 ' TMS FUNCTION	
924	002644	056066	DH23	; PC	
925	002646	057364	DT23	; SERRPC	
926	002650	000000	0		
927					
928			; ITEM	130	
929					
930	002652	054050	EM130	; INCORRECT DATA STORED DURING 'STORE BSPAD 00:37 ' TMS FUNCTION	
931	002654	057250	DH130	; PC MED WCSLOC E-DATA R-DATA	
932	002656	057326	DT11	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3	
933	002660	000000	0		
934					
935			; ITEM	131	
936					
937	002662	054147	EM131	; WATCH-DOG TIMEOUT DURING 'STORE BSPAD 00:37 ' TMS FUNCTION	
938	002664	056066	DH23	; PC	
939	002666	057364	DT23	; SERRPC	
940	002670	000000	0		
941					
942			; ITEM	132	
943					
944	002672	054242	EM132	; INCORRECT DATA STORED DURING 'STORE CSPAD 17:00 ' TMS FUNCTION	
945	002674	057250	DH132	; PC MED WCSLOC E-DATA R-DATA	
946	002676	057326	DT11	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3	
947	002700	000000	0		
948					
949			; ITEM	133	
950					
951	002702	054341	EM133	; WATCH-DOG TIMEOUT DURING 'STORE CSPAD 17:00 ' TMS FUNCTION	
952	002704	056066	DH23	; PC	
953	002706	057364	DT23	; SERRPC	
954	002710	000000	0		
955					
956			; ITEM	134	
957					
958	002712	054434	EM134	; INCORRECT DATA LOADED DURING 'LOAD ASPAD 00:37 ' TMS FUNCTION	
959	002714	057250	DH126	; PC MED WCSLOC E-DATA R-DATA	
960	002716	057326	DT11	; SERRPC \$REG0 \$REG1 \$REG2 \$REG3	
961	002720	000000	0		
962					
963			; ITEM	135	
964					
965	002722	054532	EM135	; WATCH-DOG TIMEOUT DURING 'LOAD ASPAD 00:37 ' TMS FUNCTION	
966	002724	056066	DH23	; PC	
967	002726	057364	DT23	; SERRPC	
968	002730	000000	0		
969					

```
970          ; ITEM 136
971
972 002732 054624 EM136 ; INCORRECT DATA LOADED DURING 'LOAD BSPAD 00:37 ' TMS FUNCTION
973 002734 057250 DH130 ; PC MED WCSLOC E-DATA R-DATA
974 002736 057326 DT11 ; SERRPC $REG0 $REG1 $REG2 $REG3
975 002740 000000 0
976
977          ; ITEM 137
978
979 002742 054722 EM137 ; WATCH-DOG TIMEOUT DURING 'LOAD BSPAD 00:37 ' TMS FUNCTION
980 002744 056066 DH23 ; PC
981 002746 057364 DT23 ; SERRPC
982 002750 000000 0
983
984          ; ITEM 140
985
986 002752 055014 EM140 ; INCORRECT DATA LOADED DURING 'LOAD CSPAD 17:00 ' TMS FUNCTION
987 002754 057250 DH132 ; PC MED WCSLOC E-DATA R-DATA
988 002756 057326 DT11 ; SERRPC $REG0 $REG1 $REG2 $REG3
989 002760 000000 0
990
991          ; ITEM 141
992
993 002762 055112 EM141 ; WATCH-DOG TIMEOUT DURING 'LOAD CSPAD 17:00 ' TMS FUNCTION
994 002764 056066 DH23 ; PC
995 002766 057364 DT23 ; SERRPC
996 002770 000000 0
997
998          ; ITEM 142
999
1000 002772 055204 EM142 ; INCORRECT DATA LOADED DURING 'LOAD TEMP-S' TMS FUNCTION
1001 002774 057250 DH132 ; PC MED WCSLOC E-DATA R-DATA
1002 002776 057326 DT11 ; SERRPC $REG0 $REG1 $REG2 $REG3
1003 003000 000000 0
1004
1005          ; ITEM 143
1006
1007 003002 055274 EM143 ; WATCH-DOG TIMEOUT DURING 'LOAD TEMP-S' TMS FUNCTION
1008 003004 056066 DH23 ; PC
1009 003006 057364 DT23 ; SERRPC
1010 003010 000000 0
1011
1012          ; ITEM 144
1013
1014 003012 055360 EM144 ; INCORRECT DATA STORED DURING 'STORE TEMP-S' TMS FUNCTION
1015 003014 057250 DH132 ; PC MED WCSLOC E-DATA R-DATA
1016 003016 057326 DT11 ; SERRPC $REG0 $REG1 $REG2 $REG3
1017 003020 000000 0
1018
1019          ; ITEM 145
1020
1021 003022 055451 EM145 ; WATCH-DOG TIMEOUT DURING 'STORE TEMP-S' TMS FUNCTION
1022 003024 056066 DH23 ; PC
1023 003026 057364 DT23 ; SERRPC
1024 003030 000000 0
```



Line	Code	Value	Definition
1025			EVEN
1026			DEFINITIONS
1027			
1028		100000	PARERR=BIT15
1029		000200	PARGEN=BIT7
1030		000100	WRTEN=BIT6
1031		000040	WWP=BIT5
1032		000020	PARDIS=BIT4
1033		000010	PAR2=BIT3
1034		000004	PAR1=BIT2
1035		000002	PAR0=BIT1
1036		000001	WCSPRS=BIT0
1037		040000	ID=BIT14
1038			
1039		076600	MED=76600
1040		000022	RDWHAM1=22
1041		000222	WRWHAM1=222
1042		000305	WRLWHAM1=305
1043			
1044		177766	CPUERR=177766
1045		177770	MBKREG=177770
1046		000344	WRFLAG=344
1047		000144	RDFLAG=144
1048		000071	SWB01=71
1049		000210	RESINS=210
1050		001000	MAINT=BIT9
1051		006001	OTHDIS=6001
1052		006005	F7SVC=6005
1053		006011	OTH11=6011
1054		006012	OTH12=6012
1055		006013	OTH13=6013
1056		006014	OTH14=6014
1057		006015	OTH15=6015
1058		006003	URESIN=6003
1059		006002	UDISP=6002
1060		006004	ODDPC=6004
1061		006000	UBRK=6000
1062		076700	XFCUDIS=076700
1063		125252	UWDEND=125252
1064			
1065			
1066	003032	177744	MEMERR: 177744
1067	003034	177540	WCSST: 177540
1068	003036	177542	WCSAR: 177542
1069	003040	177544	WCSOR: 177544
1070			
1071	003042	000000	ATMPO: WORD 0
1072	003044	000000	ATMP1: WORD 0
1073	003046	000000	TMPO: WORD 0
1074	003050	000000	TMP1: WORD 0
1075	003052	000000	COUNT: WORD 0
1076	003054	000000	COUNT0: WORD 0
1077	003056	000000	NXTST: WORD 0
1078	003060	000	LOCHIP: BYTE 0
1079	003061	000	HICHIP: BYTE 0
1080	003062	000	LOCNT: BYTE 0

MAINTAINANCE EX/DEP

1081	003063	000	HICNT:	. BYTE	0
1082	003064	000	ERCNT0:	. BYTE	0
1083	003065	000	ERCNT1:	. BYTE	0
1084					
1085	003066	042264	PMSG2:	. WORD	MSG2
1086	003070	042271		. WORD	MSG3
1087	003072	042276		. WORD	MSG4
1088	003074	042303		. WORD	MSG5
1089	003076	042322		. WORD	MSG6
1090	003100	042345		. WORD	MSG7
1091	003102	042362		. WORD	MSG8
1092	003104	042403		. WORD	MSG80
1093	003106	042410		. WORD	MSG81
1094	003110	042415		. WORD	MSG82
1095	003112	042423		. WORD	MSG83
1096	003114	042431		. WORD	MSG84
1097	003116	042437		. WORD	MSG85
1098	003120	042445		. WORD	MSG86
1099	003122	042453		. WORD	MSG9

: THIS TABLE CONTAINS THE ENTRY POINTS FOR ALL THE EIGHT GROUPS  
: OF XFC-OP-CODES. NOTE THAT IN THE SEVENTH GROUP, THE FIRST  
: OP-CODE (076600) IS A VALID "MED" OP-CODE. ALL THE REST  
: OP-CODES SHOULD TRAP AS A RESERVED INSTRUCTION WHEN  
: WCS IS NOT ENABLED.

1108	003124	006001	WCSENT:	6001	: ENTRY POINT FOR 0760XX
1109	003126	006011		6011	: FOR 0761XX
1110	003130	006012		6012	: FOR 0762XX
1111	003132	006013		6013	: FOR 0763XX
1112	003134	006014		6014	: FOR 0764XX
1113	003136	006015		6015	: FOR 0765XX
1114	003140	000000		0	: FOR 0766XX, ALL THESE OPCODES TRAP AS
1115					: RESERVED INSTRUCTIONS.
1116	003142	006002		6002	: FOR 0767XX

: THIS TABLE CONTAINS THE READ MED CODES FOR WFP  
: ACCUMULATORS, GROUPED AS <0:1>, <2:3>, <4:5>

1122	003144		FP05XX:		
1123					
1124	003144	070	FP01XX:	. BYTE	70, 30, 50, 10
1125	003147	010			
1126	003150	071		. BYTE	71, 31, 51, 11
1127	003153	011			
1128					
1129	003154	072	FP23XX:	. BYTE	72, 32, 52, 12
1130	003157	012			
1131	003160	073		. BYTE	73, 33, 53, 13
1132	003163	013			
1133					
1134	003164	074	FP45XX:	. BYTE	74, 34, 54, 14
1135	003167	014			
1136	003170	075		. BYTE	75, 35, 55, 15



1137 003173 015

1138

1139

1140

1141

1142

1143

1144 003174 275

1145 003175 235

1146 003176 255

1147 003177 215

1148 003200 274

1149 003201 234

1150 003202 254

1151 003203 214

1152 003204 273

1153 003205 233

1154 003206 253

1155 003207 213

1156 003210 272

1157 003211 232

1158 003212 252

1159 003213 212

1160 003214 271

1161 003215 231

1162 003216 251

1163 003217 211

1164 003220 270

1165 003221 230

1166 003222 250

1167 003223 210

1168

1169

1170

1171

1172 003224 005

1173 003225 015

1174 003226 025

1175 003227 035

1176 003230 004

1177 003231 014

1178 003232 024

1179 003233 034

1180 003234 003

1181 003235 013

1182 003236 023

1183 003237 033

1184 003240 002

1185 003241 012

1186 003242 022

1187 003243 032

1188 003244 001

1189 003245 011

1190 003246 021

1191 003247 031

1192 003250 000

; THIS TABLE CONTAINS THE WRITE-MED-CODES FOR THE WARM  
; FLOATING POINT (WFP) REGISTERS. SIX REGISTERS EACH  
; IN BOTTOM HALF OF ASPLO, ASPHI, BSPLO, BSPHI

FAC05W:	BYTE	275	; FAC0	5
	BYTE	235	; FAC1	5
	BYTE	255	; FAC2	5
	BYTE	215	; FAC3	5
	BYTE	274	; FAC0	4
	BYTE	234	; FAC1	4
	BYTE	254	; FAC2	4
	BYTE	214	; FAC3	4
	BYTE	273	; FAC0	3
	BYTE	233	; FAC1	3
	BYTE	253	; FAC2	3
	BYTE	213	; FAC3	3
	BYTE	272	; FAC0	2
	BYTE	232	; FAC1	2
	BYTE	252	; FAC2	2
	BYTE	212	; FAC3	2
	BYTE	271	; FAC0	1
	BYTE	231	; FAC1	1
	BYTE	251	; FAC2	1
	BYTE	211	; FAC3	1
	BYTE	270	; FAC0	0
	BYTE	230	; FAC1	0
	BYTE	250	; FAC2	0
	BYTE	210	; FAC3	0

; THIS TABLE CONTAINS THE DATA PATTERNS THAT ARE  
; WRITTEN IN THE WFP REGISTERS IN BASE MACHINE

PFAC05:	BYTE	05
	BYTE	15
	BYTE	25
	BYTE	35
	BYTE	04
	BYTE	14
	BYTE	24
	BYTE	34
	BYTE	03
	BYTE	13
	BYTE	23
	BYTE	33
	BYTE	02
	BYTE	12
	BYTE	22
	BYTE	32
	BYTE	01
	BYTE	11
	BYTE	21
	BYTE	31
	BYTE	00

1193 003251 010  
1194 003252 020  
1195 003253 030

. BYTE 10  
. BYTE 20  
. BYTE 30

1196  
1197  
1198  
1199  
1200

; THIS TABLE CONTAINS THE WRITE-MED-CODES FOR THE C-SCRATCH  
; PAD REGISTERS (CSP). NOTE THAT CSP 12 , CSP 13 , CSP 14 ,  
; CSP 15 ARE NOT INCLUDED.

1201 003254 300  
1202 003255 301  
1203 003256 302  
1204 003257 303  
1205 003260 304  
1206 003261 305  
1207 003262 306  
1208 003263 307  
1209 003264 310  
1210 003265 311  
1211 003266 312  
1212 003267 313

CSP00W: . BYTE 300 ; CSP 0  
. BYTE 301 ; CSP 1  
. BYTE 302 ; CSP 2  
. BYTE 303 ; CSP 3  
. BYTE 304 ; CSP 4  
. BYTE 305 ; CSP 5  
. BYTE 306 ; CSP 6  
. BYTE 307 ; CSP 7  
. BYTE 310 ; CSP 8  
. BYTE 311 ; CSP 9  
. BYTE 312 ; CSP 10  
. BYTE 313 ; CSP 11

1214 003270 000220  
1215 003272 000260  
1216 003274 000261

WCSA0W: . WORD 220  
. WORD 260  
. WORD 261

1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225

. EVEN

1226  
1227  
1228  
1229  
1230  
1231  
1232

; BADTMO  
; THIS IS CPU ERROR TRAP HANDLER AND SERVICES UNEXPECTED TIME-OUT  
; TRAPS TO 4. THE ADDRESS (PC) AT WHICH THE TIME-OUT OCCURED IS  
; TYPED OUT. THE TEST IN WHICH THE TIME-OUT OCCURED IS ABORTED  
; (MESSAGE TYPED) AND A RETURN IS MADE TO THE BEGINING OF THE  
; NEXT TEST (THRU 'NXTST').

1233  
1234 003276 011637 037034  
1235 003302 104401 042552  
1236 003306 013746 177766  
1237 003312 104402  
1238 003314 104401 043233  
1239 003320 013746 037034  
1240 003324 162716 000002  
1241 003330 104402  
1242 003332 104401 042652  
1243 003336 013716 003056

BADTMO: MOV (SP), TRAPC  
TYPE , MSG13  
MOV @#CPUERR, -(SP) ; GET CPU ERROR REGISTER  
TYPOC ; AND TYPE IT OUT  
TYPE , MSG31  
MOV TRAPC, -(SP)  
SUB #2, (SP)  
TYPOC  
TYPE , MSG17  
MOV NXTST, (SP) ; ON RETURN GO TH THE NEXT TEST  
; ENABLE ERROR LOG

1244  
1245 003342 000002  
1246  
1247  
1248

RTI



```

1249 ;BADPAR
1250 ;THIS IS A PARITY TRAP HANDLER AND SERVICES UNEXPECTED
1251 ;PARITY ERROR TRAPS TO 114. THE TRAP COULD BE BECAUSE OF
1252 ;A CACHE PARITY ERROR, MAIN MEMORY PARITY ERROR OR WCS
1253 ;PARITY ERROR. IF THE TRAP WAS DUE TO A CACHE PARITY ERROR
1254 ;THEN A SIMPLE RETURN IS DONE (NO REPORT). IF THE TRAP
1255 ;WAS DUE TO WCS OR MAIN MEMORY PARITY ERROR IT IS SO
1256 ;REPORTED AND A RETURN IS MADE TO THE BEGINNING OF THE
1257 ;NEXT TEST (THRU 'NXTST'). A MESSAGE, 'TEST ABORTED',
1258 ;IS ALSO PRINTED OUT. THE ADDRESS (PC) AT WHICH THE TRAP OCCURRED
1259 ;IS ALSO REPORTED.
1260
1261 003344 011637 037034 BADPAR: MOV (SP),TRAPC ;SAVE OLD PC FROM THE STACK
1262 003350 032777 000340 177454 BIT #BIT5+BIT6+BIT7,@MEMERR ;PARITY ERROR FROM CACHE?
1263 003356 001031 BNE 25 ;YES
1264 003360 032777 100000 177444 BIT #BIT15,@MEMERR ;PARITY ERROR FROM MEMORY?
1265 003366 001007 BNE 35 ;YES
1266 003370 032777 100000 177436 BIT #PARERR,@WCSST ;PARITY ERROR FROM WCS?
1267 003376 001421 BEQ 25 ;NO
1268 003400 104401 043134 TYPE ,MSG29 ;UNEXPECTED WCS PARITY ERROR
1269 003404 000402 BR 45
1270
1271 003406 104401 043172 35: TYPE ,MSG30 ;UNEXPECTED MEMORY PARITY ERROR
1272 003412 104401 043233 45: TYPE ,MSG31 ;PC=
1273 003416 013746 037034 MOV TRAPC,-(SP)
1274 003422 162716 000002 SUB #2,(SP)
1275 003426 104402 TYPOC
1276 003430 104401 042652 TYPE ,MSG17 ;TEST ABORTED
1277 003434 013716 003056 MOV NXTST,(SP) ;RETURN TO BEGINNING OF NEXT TEST
1278 ;ENABLE ERROR LOG
1279 003440 000002 RTI ;RETURN
1280
1281 003442 013716 037034 25: MOV TRAPC,(SP) ;PROCEED WITH NORMAL RETURN
1282 003446 000002 RTI
1283
1284
1285

```

```

1286 .SBTTL PROGRAM STARTS HERE
1287
1288
1289 003450 012706 001100 START: ;; CLEAR THE COMMON TAGS ($CMTAG) AREA
1290 003454 005026 MOV #CMTAG,R6 ;; FIRST LOCATION TO BE CLEARED
1291 003456 022706 001140 CLR (R6)+ ;; CLEAR MEMORY LOCATION
1292 003462 001374 CMP #SWR,R6 ;; DONE?
1293 003464 012706 001100 BNE -6 ;; LOOP BACK IF NO
1294 MOV #STACK,SP ;; SETUP THE STACK POINTER
1295 003470 012737 040004 000020 ;; INITIALIZE A FEW VECTORS
1296 003476 012737 000340 000022 MOV $$SCOPE,@#IOTVEC ;; IOT VECTOR FOR SCOPE ROUTINE
1297 003504 012737 040262 000030 MOV #340,@#IOTVEC+2 ;; LEVEL 7
1298 003512 012737 000340 000032 MOV $ERROR,@#EMTVEC ;; EMT VECTOR FOR ERROR ROUTINE
1299 003520 012737 042014 000034 MOV #340,@#EMTVEC+2 ;; LEVEL 7
1300 003526 012737 000340 000036 MOV $TRAP,@#TRAPVEC ;; TRAP VECTOR FOR TRAP CALLS
1301 003534 012737 042066 000024 MOV #340,@#TRAPVEC+2 ;; LEVEL 7
1302 003542 012737 000340 000026 MOV $PWDRN,@#PWRVEC ;; POWER FAILURE VECTOR
1303 003550 013737 036462 036454 MOV #340,@#PWRVEC+2 ;; LEVEL 7
1304 003556 005037 001224 SENDCT,$EOPCT ;; SETUP END-OF-PROGRAM COUNTER
1305 003562 112737 000001 001115 CLR $ESCAPE ;; CLEAR THE ESCAPE ON ERROR ADDRESS
1306 003570 005037 001222 MOVB #1,$ERMAX ;; ALLOW ONE ERROR PER TEST
1307 003574 012737 003574 001106 CLR $TIMES ;; INITIALIZE NUMBER OF ITERATIONS
1308 003602 012737 003602 001110 MOV #,$LPADR ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
1309 MOV #,$LPERR ;; SETUP THE ERROR LOOP ADDRESS
1310 ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1311 003610 013746 000004 ;; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1312 003614 012737 003650 000004 MOV @#ERRVEC,-(SP) ;; SAVE ERROR VECTOR
1313 003622 012737 177570 001140 MOV #64$,@#ERRVEC ;; SET UP ERROR VECTOR
1314 003630 012737 177570 001142 MOV #DSWR,SWR ;; SETUP FOR A HARDWARE SWICH REGISTER
1315 003636 022777 177777 175274 MOV #DDISP,DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
1316 003644 001012 CMP #-1,@SWR ;; TRY TO REFERENCE HARDWARE SWR
1317 BNE 66$ ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
1318 BR 65$ ;; AND THE HARDWARE SWR IS NOT = -1
1319 003650 012716 003656 64$: MOV #65$,(SP) ;; BRANCH IF NO TIMEOUT
1320 003654 000002 RTI ;; SET UP FOR TRAP RETURN
1321 003656 012737 000176 001140 65$: MOV #SWREG,SWR ;; POINT TO SOFTWARE SWR
1322 003664 012737 000174 001142 MOV #DISPREG,DISPLAY
1323 003672 012637 000004 66$: MOV (SP)+,@#ERRVEC ;; RESTORE ERROR VECTOR
1324
1325 003676 005037 001244 CLR $PASS ;; CLEAR PASS COUNT
1326 003702 122737 000001 001256 CMPB #APTENV,$ENV ;; TEST USER RUNNING UNDER APT
1327 003710 001003 BNE 67$ ;; NO, USE NON-APT SWITCH
1328 003712 012737 001260 001140 MOV #SWREG,SWR ;; YES, USE APT SWITCH REGISTER
1329 003720 67$:
1330
1331 ;; PROGRAM HEADER
1332 003720 104401 003726 TYPE ,69$ ;; TYPE ASCIZ STRING
1333 003724 000430 BR 68$ ;; GET OVER THE ASCIZ
1334 ;; 69$: .ASCIZ <15><12>"MAINDEC-11-DQKUA-AO. PDP-11/60 WCS DIAGNOSTIC"
1335 004006 68$:
1336
1337 ;; POWER FAIL RESTART ENTERS HERE
1338
1339 004006 012706 001100 RESTART: MOV #CMTAG,R6 ;; CLEAR THE COMMON TAGS ($CMTAG) AREA
1340 004012 005026 CLR (R6)+ ;; FIRST LOCATION TO BE CLEARED
1341 004014 022706 001140 CMP #SWR,R6 ;; CLEAR MEMORY LOCATION
;; DONE?

```



```
1342 004020 001374          BNE      .-6          ;; LOOP BACK IF NO
1343 004022 012706 001100    MOV      #STACK, SP  ;; SETUP THE STACK POINTER
1344 004026 005037 001224    CLR      $ESCAPE     ;; CLEAR THE ESCAPE ON ERROR ADDRESS
1345 004032 112737 000001 001115  MOVB     #1, $ERMAX   ;; ALLOW ONE ERROR PER TEST
1346 004040 012737 004040 001106  MOV      #. , $LPADR  ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
1347 004046 012737 004046 001110  MOV      #. , $LPERR  ;; SETUP THE ERROR LOOP ADDRESS
1348
1349                          ; ---NEW PASS ENTERS HERE---
1350 004054 012706 001100    NEWPAS: MOV     #STACK, SP ; SETUP THE STACK POINTER
1351 004060 012737 003276 000004    MOV     #BADTMO, @#4
1352 004066 012737 003344 000114    MOV     #BADPAR, @#114
1353 004074 076600          MED
1354 004076 000022          RDWHAM!
1355 004100 052700 000001    BIS     #BIT0, R0    ; ENABLE ERROR LOG IF NOT
1356 004104 076600          MED                    ; ERROR LOG CONTINUES MODE
1357 004106 000222          WRWHAM!
1358
```

```

1359
1360 ;*****
1361 ;*TEST 1 CHECK THAT WCS REGISTERS CAN BE ADDRESSED
1362 ;THIS TEST CHECKS THAT THE THREE REGISTERS OF WCS CAN BE
1363 ;ADDRESSED CORRECTLY WITHOUT TIMING OUT. THE REGISTERS ARE:
1364 ;WCS STATUS REGISTER (777740)
1365 ;WCS ADDRESS REGISTER (777542)
1366 ;WCS DATA REGISTER (777544)
1367 ;ON THE 11/60, THESE ARE INTERNAL (PROCESSOR) ADDRESSES.
1368 ;*****
1369 004110 000004 TST1: SCOPE
1370 004112 012737 004224 003056 MOV #TST2,NXTST ;STARTING ADDRESS OF NEXT TEST
1371
1372 004120 012700 003034 MOV #WCSST,R0 ;INITIALIZE ADRES TO BE REFERENCED
1373 004124 012701 177775 MOV #-3,R1 ;INITIALIZE COUNT
1374 004130 012737 004136 001110 MOV #1$,SLPERR ;LOOP ON ERROR TO THIS POINT
1375 004136 005002 1$: CLR R2
1376 004140 012737 004202 000004 4$: MOV #2$,@#4 ;SET UP TIME-OUT VECTOR
1377 004146 005770 000000 TST @ (R0) ;REFERENCE THE WCS REGISTER
1378 004152 005720 3$: TST (R0)+ ;POINT TO THE NEXT WCS ADRES
1379 004154 005201 INC R1
1380 004156 001370 BNE 4$ ;DONE?
1381 004160 012737 003276 000004 MOV #BADTMO,@#4 ;REESTABLISH TIME OUT VECTOR
1382 004166 005702 TST R2 ;ANY REGISTER TIMED OUT?
1383 004170 001415 BEQ TST2 ;GO TO THE NEXT TEST
1384 004172 104401 043112 TYPE ,MSG28 ;PROGRAM ABORTED
1385 004176 000137 036426 JMP $EOP ;GO TO END OF PASS
1386
1387 004202 022616 2$: CMP (SP)+,(SP) ;RESTORE THE STACK
1388 004204 012737 003276 000004 MOV #BADTMO,@#4 ;RE-ESTABLISH NORMAL TIME-OUT
1389 ;VECTOR
1390 INC R2 ;FLAG THIS TIMEOUT
1391 004214 011037 001162 MOV (R0),SREGO ;SAVE THE REGISTER-ADRES
1392 ;WHICH TIMED OUT
1393 004220 104001 ERROR 1 ;TIMEOUT OCCURRED ON REFERENCING
1394 ;A WCS REGISTER. ADRES OF
1395 ;THAT REGISTER IS GIVEN IN
1396 ;THE PRINTOUT
1397 004222 000753 BR 3$ ;GO BACK & CHECK THE REST
1398
1399
1400 ;*****
1401 ;*TEST 2 CHECK WCS STATUS REGISTER, ID BIT, WCS PRESENT BIT
1402 ;THIS TEST CHECKS THAT WCS PRESENT BIT IS SET IN LOG WHAMI
1403 ;REGISTER. THEN IT CLEARS THE STATUS REGISTER AND CHECKS IF
1404 ;THE 'WRITE ENABLE' SIGNAL WAS GENERATED. THEN IT CHECKS
1405 ;IF THE 'PARDIS' (PARITY DISABLE) AND THE 'WWP'
1406 ;(WRITE WRONG PARITY) BITS WERE CLEARED. THIS INVOLVES
1407 ;SELECTION OF THE WCS HOT BOX, GENERATING THE 'WRITE
1408 ;ENABLE' SIGNAL AND WRITING A '0' INTO THE
1409 ;'PARDIS' AND 'WWP' F/FS.
1410 ;WRITE INTO STATUS REGISTER INVOLVES:
1411 ;GENERATION OF 'STATUS(1)L'
1412 ;HBMUXSEL0, HBMUXSEL1
1413 ;WRITTEN, WRITE PULSE
1414 ;READ OF STATUS REGISTER INVOLVES THE SELECTION OF THE

```



```
1415 ;CORRECT 'BUS DIN' MUX (PORT A).
1416 ;*****
1417 004224 000004 TST2: SCOPE
1418 004226 012737 004342 003056 MOV #TST3,NXTST ;STARTING ADDRESS OF NEXT TEST
1419 004234 076600 MED ;READ LOGF WHAMI
1420 004236 000022 RDWHAMI
1421 004240 032700 000040 BIT #BIT5,RO ;WCS PRESENT BIT SET?
1422 004244 001003 BNE 15 ;YES
1423 004246 010037 001162 MOV RO,$REGO ;SAVE LOG WHAMI
1424 004252 104002 ERROR 2 ;WCS PRESENT BIT WAS NOT SET
1425 ; IN WHAMI REGISTER
1426
1427 004254 052700 000200 15: BIS #BIT7,RO ;ENABLE WCS
1428 004260 076600 MED
1429 004262 000222 WRWHAMI
1430
1431 004264 013700 003034 25: MOV WCSST,RO
1432 004270 005010 CLR (RO)
1433 004272 032710 040000 BIT #ID,(RO) ;ID CODE SET?
1434 004276 001003 BNE 35 ;YES
1435 004300 011037 001162 MOV (RO),$REGO
1436 004304 104003 ERROR 3 ;WCS ID CODE (BIT 14) NOT SET
1437 ; IN THE STATUS REGISTER
1438
1439 004306 005077 176522 35: CLR @WCSST
1440 004312 032710 000020 BIT #PARDIS,(RO) ;CHECK IF 'PARITY DISABLE' BIT IS
1441 004316 001403 BEQ 45 ;CLEARED
1442 ; 'PARITY DISABLE' BIT DID NOT
1443 004320 011037 001162 MOV (RO),$REGO ;GET CLEARED ON CLEARING THE
1444 004324 104004 ERROR 4 ;STATUS REGISTER.
1445
1446 004326 032710 000040 45: BIT #WWP,(RO) ;CHECK IF 'WRITE WRONG PARITY' BIT
1447 004332 001403 BEQ TST3 ;IS CLEARED
1448 004334 011037 001162 MOV (RO),$REGO ;SAVE STATUS REGISTER
1449 004340 104005 ERROR 5 ;'WRITE WRONG PARITY' BIT DID
1450 ;NOT GET CLEARED ON CLEARING
1451 ;THE STATUS REGISTER.
1452
1453
1454 ;*****
1455 ;*TEST 3 CHECK STATUS REGISTER, SET 'WWP' & 'PARDIS' BITS
1456 ;THIS TEST DOES A WRITE OF THE STATUS REGISTER AND
1457 ;CHECKS IF THE 'WRITE WRONG PARITY' AND 'PARITY DISABLE'
1458 ;BITS CAN BE SET AND CLEARED. SETTING OF 'PARDIS' BIT
1459 ;CLEARS THE 'PARITY ERROR' (PARERR) BIT, IT IS CHECKED.
1460 ;*****
1461 004342 000004 TST3: SCOPE
1462 004344 012737 004532 003056 MOV #TST4,NXTST ;STARTING ADDRESS OF NEXT TEST
1463
1464 004352 013700 003034 MOV WCSST,RO ;POINTER TO STATUS REGISTER
1465 004356 005010 CLR (RO) ;CLEAR STATUS REGISTER
1466 004360 012737 004366 001110 MOV #10,$LPERR ;LOOP ON ERROR TO THIS POINT
1467 004366 012710 000020 105: MOV #PARDIS,(RO) ;SET 'PARDIS' BIT IN STATUS REGISTER
1468
1469 004372 032710 000020 15: BIT #PARDIS,(RO) ;'PARDIS' BIT SET?
1470 004376 001003 BNE 25 ;OK, IF YES
```

```
1471 004400 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1472 004404 104006              ERROR    6                ;'PARITY DISABLE' BIT COULD
1473                                ;NOT BE SET ON DOING A
1474                                ;WRITE TO THE STATUS REGISTER.
1475
1476 004406 032710 000040      25:    BIT      #WWP, (R0)    ;CHECK 'WWP' BIT SHOULD BE CLEAR
1477 004412 001403              BEQ      35                ;OK
1478 004414 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1479 004420 104005              ERROR    5                ;'WWP' GOT WRITTEN WHEN TRYING
1480                                ;TO WRITE 'PARDIS' BIT IN STATUS
1481                                ;REGISTER.
1482 004422 032710 100000      35:    BIT      #PARERR, (R0)  ;CHECK IF 'PARITY ERROR' BIT IS CLEAR
1483 004426 001403              BEQ      45                ;OK, IF YES
1484 004430 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1485 004434 104007              ERROR    7                ;'PARITY ERROR' BIT SHOULD
1486                                ;BE CLEARED WHEN 'PARITY
1487                                ;'DISABLE' WAS SET. 'PARERR'
1488                                ;WAS NOT CLEARED.
1489
1490 004436 042710 000020      45:    BIC      #PARDIS, (R0)  ;CLEAR 'PARDIS' BIT
1491 004442 032710 000020      BIT      #FARDIS, (R0)    ;CHECK IF IT IS CLEARED
1492 004446 001403              BEQ      55                ;OK
1493 004450 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1494 004454 104004              ERROR    4                ;'PARDIS' BIT COULD NOT BE
1495                                ;CLEARED
1496
1497 004456 012710 000040      55:    MOV      #WWP, (R0)      ;SET 'WWP' BIT IN STATUS REGISTER
1498 004462 032710 000040      BIT      #WWP, (R0)      ;DID IT SET?
1499 004466 001003              BNE      65                ;OK
1500 004470 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1501 004474 104010              ERROR    10               ;'WWP' BIT COULD NOT BE SET
1502                                ;IN STATUS REGISTER.
1503
1504 004476 032710 000020      65:    BIT      #PARDIS, (R0)    ;CHECK IF 'PARDIS' IS CLEAR
1505 004502 001403              BEQ      75                ;OK
1506 004504 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1507 004510 104004              ERROR    4                ;'PARDIS' NOT CLEAR. IT
1508                                ;GOT WRITTEN WHEN TRYING TO WRITE
1509                                ;'WWP' BIT.
1510
1511 004512 042710 000040      75:    BIC      #WWP, (R0)      ;CLEAR 'WWP' BIT
1512 004516 032710 000040      BIT      #WWP, (R0)      ;IS IT CLEARED?
1513 004522 001403              BEQ      TST4              ;OK
1514 004524 011037 001162      MOV      (R0), $REGO      ;SAVE STATUS REGISTER
1515 004530 104005              ERROR    5                ;'WWP' BIT COULD NOT BE
1516                                ;CLEARED
1517
1518
1519
1520      ;*****
1521      ;*TEST 4      CHECK 'WRITE' AND 'READ' OF LOCATION 0 IN WCS, EVEN PATTERN
1522      ;PATTERN '0' IS WRITTEN INTO LOCATION 0 OF WCS DATA ARRAY.
1523      ;IT IS CHECKED IF 'WRITEN' IS SET, 'PARGEN' IS CLEAR, 'WWP' IS
1524      ;CLEAR, 'PARDIS' IS STILL SET AND 'PARERR' IS CLEAR.
1525      ;THEN A READ FROM LOCATION 0 IS DONE. IT IS CHECKED IF
1526      ;'PARGEN' IS CLEAR, 'WWP' IS CLEAR, 'PARDIS' IS SET, AND 'PO'
```



```

1527 ;EVEN PARITY IS BEING GENERATED AND CHECKED.
1528 ;THIS IS THE FIRST TIME IN THE PROGRAM, A 'WRITE' AND 'READ'
1529 ;IS DONE TO AND FROM THE WCS DATA ARRAY. IN CASE OF A FATAL
1530 ;ERROR UNEXPECTED ACTION MAY TAKE PLACE.
1531 ;*****
1532 004532 000004 TST4: SCOPE
1533 004534 012737 005076 003056 MOV #TST5,NXTST ;STARTING ADDRESS OF NEXT TEST
1534
1535 004542 005037 003042 CLR ATMP0 ;COUNT FOR TIMEOUT TRAPS
1536 004546 005037 003044 CLR ATMP1 ;COUNT FOR ILLEGAL INSTRUCTION TRAPS
1537 ;IF ANY OF THESE TWO COUNTS GO TO
1538 ;2 THEN THE PROGRAM IS ABORTED.
1539 004552 012737 005006 000114 MOV #3$,@#114 ;SET UP PARITY ERROR TRAP VECTOR
1540 004560 012737 005014 000004 MOV #4$,@#4 ;SET UP TIME-OUT TRAP VECTOR
1541 004566 012737 005026 000010 MOV #5$,@#10 ;SET UP ILLEGAL AND RESERVED
1542 ;INSTRUCTION TRAP VECTOR
1543 004574 012777 000020 176232 MOV #PARDIS,@WCSST ;SET 'PARITY DISABLE' BIT
1544 004602 005077 176230 CLR @WCSAR ;ADDRESS LOCATION 0 OF WCS
1545 004606 005077 176226 CLR @WCSOR ;WRITE 0 INTO LOCATION 0 OF WCS
1546 004612 004737 036550 JSR PC,VECTRS ;RE-ESTABLISH TIMEOUT, PARITY
1547 ;ILLEGAL INSTRUCTION TRAPS
1548 004616 012737 040020 037252 MOV #40020,CESTAT ;EXPCTD STATUS REG
1549 004624 012737 000014 037256 MOV #PAR1+PAR2,CMASK ;MASK BITS FOR STATUS
1550 004632 012737 004656 037260 MOV #1$,CRETRN ;RETURN ADDRESS
1551 004640 004737 037152 JSR PC,CHKSTAT ;CHECK STATUS REGISTER CONTENTS
1552 004644 005037 001166 CLR $REG2 ;SAVE LOCATION INTO WHICH WRITTEN
1553 004650 005037 001170 CLR $REG3 ;SAVE PATTERN
1554 004654 104011 ERROR 11 ;STATUS REGISTER ERROR ON WRITING
1555 ;LOCATION 0 OF WCS WITH A 0
1556 ;PATTERNS. BITS WERE NOT AS EXPECTED
1557
1558 004656 012737 005006 000114 15: MOV #3$,@#114 ;PARITY TRAP VECTOR
1559 004664 012737 005014 000004 MOV #4$,@#4 ;TIMEOUT TRAP VECTOR
1560 004672 012737 005026 000010 MOV #5$,@#10 ;ILLEGAL INSTRUCTION TRAP
1561 004700 005077 176132 CLR @WCSAR ;ADDRESS LOCATION 0 OF WCS
1562 004704 017737 176130 001166 MOV @WCSOR,$REG2 ;READ LOCATION 0
1563 004712 004737 036550 JSR PC,VECTRS ;RE-ESTABLISH PARITY, TIMEOUT & ILLEGAL
1564 ;TRAP VECTORS
1565 004716 005737 001166 TST $REG2 ;IS IT 0?
1566 004722 001405 BEQ 25 ;OK, IF IT IS 0
1567 004724 005037 001164 CLR $REG1 ;SAVE EXPECTED PATTERN (DATA)
1568 004730 005037 001162 CLR $REG0 ;SAVE LOCATION THAT WAS READ
1569 004734 104012 ERROR 12 ;LOCATION 0 OF WCS WAS READ.
1570 ;IT SHOULD CONTAIN 0 THAT WAS
1571 ;WRITTEN BEFORE. IT DID NOT.
1572
1573 004736 012777 000000 176072 25: MOV #0,@WCSAR ;EXPECTED STATUS REGISTER
1574 004744 012737 040020 037252 MOV #40020,CESTAT ;EXPECTED STATUS REGISTER
1575 004752 012737 000014 037256 MOV #PAR1+PAR2,CMASK ;MASK BITS FOR STATUS
1576 004760 012737 005004 037260 MOV #10$,CRETRN ;RETURN ADDRESS FOR 'CHKSTAT'
1577 004766 004737 037152 JSR PC,CHKSTAT ;CHECK STATUS REGISTER CONTENTS
1578 004772 005037 001166 CLR $REG2 ;SAVE LOCATION READ
1579 004776 005037 001170 CLR $REG3 ;SAVE PATTERN READ
1580 005002 104013 ERROR 13 ;STATUS REGISTER ERROR ON READING
1581 ;LOCATION 0 OF WCS (0 WAS WRITTEN
1582 ;INTO IT BEFORE). BITS WERE NOT AS

```

```

1583                                     ; EXPECTED.
1584
1585 005004 000432                      105: BR      85
1586
1587                                     ; ENTER HERE IF A PARITY ERROR
1588                                     ; TRAP OR TIMEOUT OR ILLEGAL
1589                                     ; INSTRUCTION TRAP TAKES PLACE
1590                                     ; ON DOING A WRITE OR READ TO WCS
1591
1592 005006 104406 042520                  35:  TYPMSG ,MSG12      ; UNEXPECTED PARITY ERROR TRAP OCCURRED
1593 005012 000411                        BR      65
1594 005014 104406 042552                  45:  TYPMSG ,MSG13      ; UNEXPECTED TIMEOUT TRAP
1595 005020 005237 003042                  INC     ATMP0
1596 005024 000404                        BR      65
1597 005026 104406 042617                  55:  TYPMSG ,MSG14      ; ILLEGAL INSTRUCTION TRAP
1598 005032 005237 003044                  INC     ATMP1
1599
1600 005036 162776 000002 000000 65:  SUB     #2,@(SP)
1601 005044 017637 000000 001162          MOV     @(SP),SREG0
1602 005052 022626                        CMP     (SP)+,(SP)+
1603 005054 023727 001162 004656          CMP     SREG0,#15      ; TRAP ON DOING READ OR WRITE?
1604 005062 103002                        BHIS   75              ; ON READ
1605                                     ; AN UNEXPECTED TRAP (TIMEOUT,
1606                                     ; PARITY ERROR OR ILLEGAL INSTRUCTION)
1607                                     ; TOOK PLACE ON DOING
1608                                     ; WRITE TO WCS LOCATION
1609 005064 104014                        ERROR  14              ; 0. PC AT WHICH THE TRAP TOOK
1610                                     ; PLACE IS GIVEN IN THE ERROR
1611                                     ; MESSAGE (TRPLOC)
1612
1613 005066 000401                        BR      85
1614
1615 005070 104027                        75:  ERROR  27              ; AN UNEXPECTED TRAP (TIMEOUT,
1616                                     ; PARITY ERROR OR ILLEGAL INSTRUCTION)
1617                                     ; TOOK PLACE ON DOING A READ FROM
1618                                     ; WCS LOCATION 0. PC AT WHICH THE TRAP
1619                                     ; OCCURED IS GIVEN IN THE ERROR MESSAGE
1620
1621 005072 004737 036550                  85:  JSR     PC,VECTRS  ; REESTABLISH VECTORS AT
1622                                     ; 4, 114 AND 10
1623
1624
1625                                     ; *****
1626 :*TEST 5      CHECK 'WRITE' AND 'READ' OF LOCATION 0 IN WCS, ODD PATTERN
1627               ; THIS TEST DOES A WRITE OF AN ODD, ALTERNATE 0'S AND 1'S PATTERN
1628               ; INTO LOCATION 0 AND CHECKS IF THE APPROPRIATE BITS ARE SET
1629               ; IN THE STATUS REGISTER. THEN IT READS LOCATION 0 AND CHECKS
1630               ; IF THE CORRECT DATA WAS READ, AGAIN THE STATUS REGISTER
1631               ; IS CHECKED.
1632                                     ; *****
1633 005076 000004                        TST5: SCOPE
1634 005100 012737 005476 003056          MOV     #TST6,NXTST   ; STARTING ADDRESS OF NEXT TEST
1635 005106 012737 005354 000114          MOV     #35,@#114    ; SET UP PARITY ERROR TRAP VECTOR
1636 005114 012737 005362 000004          MOV     #45,@#4      ; SET UP TIMEOUT TRAP VECTOR
1637 005122 012737 005374 000010          MOV     #55,@#10     ; SET UP ILLEGAL & RESERVED INSTRUCTION TRAP
1638 005130 012777 000020 175676          MOV     #PARDIS,@WCSST ; SET 'PARITY DISABLE' BIT

```





```
1695 005432 104014 ERROR 14 ;AN UNEXPECTED TRAP OCCURRED ON
1696 ;DOING A WRITE
1697 ;OF WCS LOCATION 0. PC AT WHICH
1698 ;TRAP OCCURRED IS GIVEN IN
1699 ;ERROR MESSAGE (TRPLOC)
1700 005434 000401 BR 85
1701
1702 005436 104027 75: ERROR 27 ;AN UNEXPECTED TRAP OCCURED ON
1703 ;DOING A READ OF WCS LOCATION 0. PC
1704 ;AT WHICH TRAP OCCURED IS GIVEN IN
1705 ;ERROR MESSAGE (TRPLOC)
1706 005440 023727 003042 000002 85: CMP ATMP0,#2 ;ABORT THE PROGRAM?
1707 005446 001407 BEQ 95 ;YES
1708 005450 023727 003044 000002 CMP ATMP1,#2
1709 005456 001403 BEQ 95
1710 005460 004737 036550 JSR PC,VECTRS ;REESTABLISH VECTORS AT 4, 114
1711 ;AND 10
1712 005464 000404 BR TST6 ;EXIT
1713 005466 104401 043112 95: TYPE ,MSG28 ;ABORT THE PROGRAM
1714 005472 000137 036426 JMP $EOP ;GO TO END OF PASS
1715
```

```
1716 ;*****
1717 ;*TEST 6 CHECK WRITE PULSES WPO, WP1, WP2 USING WRITE/READ (LOC. 0, 2000, 4000)
1718 ;THIS TEST CHECKS OUT THAT THE THREE WRITE PULSES - WPO
1719 ;WP1 AND WP2 GET GENERATED CORRECTLY. LOCATIONS 0, 2000, 4000
1720 ;OF THE WCS ARE WRITTEN WITH THREE UNIQUE PATTERNS AND
1721 ;ARE READ BACK AND THE CONTENTS CHECKED FOR CORRECT
1722 ;DATA. THE STATUS REGISTER IS CHECKED EACH TIME AFTER THE
1723 ;WRITE TO SEE IF THE BITS (PO, P1, P2, PARGEN ETC) ARE
1724 ;CORRECT. THE SEQUENCE OF CHECKING IS AS FOLLOWS:
1725
1726 ;WRITE LOC 0=0, LOC2000=177777, LOC4000=52524
1727 ;READ STATUS=330 (PO=0, P1=0, P2=1, PARDIS=PARGEN=WRTEN=1, REST=0)
1728 ;READ LOCO=0 ,LOC2000=177777, LOC4000=52524
1729
1730 ;WRITE LOCO=177777, LOC2000=52524, LOC4000=0
1731 ;READ STATUS=324 (PO=P2=0, P1=1, PARDIS=PARGEN=WRTEN=1, REST=0)
1732 ;READ LOCO=177777, LOC2000=52524, LOC4000=0
1733
1734 ;WRITE LOCO=52524, LOC2000=0, LOC4000=177777
1735 ;READ STATUS=322 (PO=1, P1=P2=0, PARDIS=PARGEN=WRTEN=1, REST=0)
1736 ;READ LOCO=52524, LOC2000=0, LOC4000=177777
1737
```

```
1738 005476 000004 TST6: SCOPE
1739 005500 012737 006052 003056 MOV #TST7,NXTST ;STARTING ADDRESS OF NEXT TEST
1740 005506 012700 006036 MOV #PATO,R0 ;INITIALIZE THE PATTERNS
1741 005512 012720 000000 MOV #J,(R0)+ ;TO BE USED. NOTE, YOU
1742 005516 012720 177777 MOV #177777,(R0)+ ;CAN CHANGE THE THREE PATTERNS
1743 005522 012720 052524 MOV #52524,(R0)+ ;BEING WRITTEN BY CHANGING THESE
1744 005526 012777 000020 175300 MOV #PARDIS,@WCSST ;DISABLE PARITY ERROR TRAP
1745 005534 012703 006044 MOV #STWRD0,R3 ;INITIALIZE POINTER TO STATUS WORD
1746 005540 012737 005556 001110 MOV #10$,SLPERR
1747 005546 013704 003036 MOV WCSAR,R4 ;R4 CONTAINS ADDRESS OF WCSAR
1748 005552 013705 003040 MOV WCSDR,R5 ;R5 CONTAINS ADDRESS OF WCSDR
1749 005556 012700 006036 10$: MOV #PATO,R0 ;INITIALIZE POINTER TO PATTERN
1750 005562 012714 000000 MOV #0,(R4) ;ADDRESS LOC 0 OF WCS
```





```

1807 006016 014012      MOV      -(R0),(R2)      ;THE CONTENTS OF THE THREE ENTRIES.
1808 006020 014042      MOV      -(R0),-(R2)    ;(PATO)--->(PAT1),(PAT1)--->(PAT2),
1809 006022 010110      MOV      R1,(R0)        ;(PAT2)--->(PATO)
1810 006024 005723      TST      (R3)+          ;INCREMENT POINTER TO THE NEXT
1811                                     ;ENTRY IN STATUS REGISTER
1812                                     ;TABLE
1813 006026 022710 000000  CMP      #0,(R0)        ;DONE ALL THREE TIMES? NOTE
1814 006032 001251      BNE      10$            ;#C MODE IS USED TO ALLOW THE
1815                                     ;USER TO CHANGE THE PATTERN IN
1816                                     ;THE PATTERN TABLE.
1817
1818
1819 006034 000406      BR       TST7           ;;
1820
1821 006036 000000      PATO:   .WORD 0          ;PATTERN TABLE
1822 006040 177777      .WORD 177777          ;THESE PATTERNS ARE USED 3 TIMES
1823 006042 052524      .WORD 52524           ;SHIFTED CYCLICALLY EACH TIME
1824
1825 006044 040230      STWRDO: .WORD 40230    ;STATUS REGISTER TABLE
1826 006046 040222      .WORD 40222           ;EXPECTED STATUS REGISTER CONTENTS
1827 006050 040224      .WORD 40224           ;FOR THE THREE CASES.
1828
1829
1830 ;*****
1831 ;*TEST 7 CHECK PARITY GENERATOR FOR HI BYTE DATA
1832 ;THIS TEST CHECKS OUT THE PARITY GENERATOR CHIP CORRESPONDING
1833 ;TO THE HI-BYTE. IN THIS TEST HI BYTE REFERS TO DATA BITS (15,14,13,12,11,10,1,
1834 ;LO BYTE REFERS TO DATA BITS (9,8,7,6,5,4,3,2).
1835 ;A PATTERN IS WRITTEN INTO THE ADDRESS REGISTER AND THE PARITY BIT
1836 ;GENERATED FOR THAT PATTERN (PARGEN) IS READ FROM
1837 ;THE STATUS REGISTER AND CHECKED IF IT IS CORRECT.
1838 ;SEQUENCE OF TESTING:
1839 ;DATA PATTERN USED FOR LOW BYTE IS KEPT CONSTANT AT
1840 ;377 (EVEN PATTERN) AND A COUNT PATTERN 0-377 IS
1841 ;USED FOR THE HI-BYTE. THE PARITY BIT IS CHECKED
1842 ;FOR CORRECTNESS. IF A WRONG PARITY WAS GENERATED AN ERROR
1843 ;MESSAGE IS REPORTED. IF THERE ARE MORE THAN
1844 ;FIVE ERRORS THE SUB-TEST IS ABORTED. THE FAULT
1845 ;COULD BE IN THE PARITY GENERATOR CHIPS, GATING
1846 ;CHIP OR THE ASSOCIATED DATA PATH.
1847
1848 ;THE ENTIRE SUB-TEST IS REPEATED WITH A CONSTANT ODD PATTERN OF
1849 ;376 IN THE LOBYTE AND COUNT PATTERN 0-377 FOR THE
1850 ;HI-BYTE.
1851 ;*****
1852 006052 000004      TST7:   SCOPE
1853 006054 012737 006256 003056  MOV      #TST10,NXTST  ;STARTING ADDRESS OF NEXT TEST
1854 006062 005037 003046  CLR      TMPO           ;IF TMPO=0, USE EVEN PATTERN ALL 1'S FOR LOBYTE
1855                                     ;IF TMPO=2, USE ODD PATTERN 376 FOR LOBYTE
1856                                     ;FOR BOTH CASES VARY HI-BYTE FROM 0 TO 377
1857 006066 012777 000020 174740  MOV      #PARDIS,@WCSST ;SET PARITY ERROR DISABLE BIT
1858 006074 013704 003046 20$:   MOV      TMPO,R4
1859 006100 016400 006252  MOV      PARPAT(R4),R0 ;INITIALIZE THE PATTERN
1860                                     ;LO BYTE=ALL 1'S, HI BYTE=0 TO 377
1861 006104 012737 006120 001110  MOV      #15,$LPERR    ;LOOP TO THIS POINT ON ERROR

```



```
1862 006112 012737 177400 003054      MOV    #-400,COUNT0 ; INITIALIZE COUNT FOR NUMBER OF PATTERNS
1863 006120 112737 177773 003064 15:    MOVB   #-5,ERCNT0  ; DO NOT TYPE MORE THAN 5 ERRORS
1864 006126 010004                25:    MOV    R0,R4      ; COPY THE PATTERN
1865
1866 006130 004737 036734                JSR    PC,XFRPGN   ; MAP INPUT PATTERN TO CHIP COMPATIBLE PATTERN
1867 006134 010477 174676                MOV    R4,@WCSAR  ; LOAD THE PATTERN INTO ADDRESS REGISTER
1868 006140 017705 174670                MOV    @WCSST,R5  ; READ STATUS REGISTER AND 'PARGEN'
1869                                ; BIT-THE PARITY BIT
1870 006144 042705 177577                BIC    #177777-PARGEN,R5 ; MASK ALL BUT PARITY BIT
1871
1872
1873 006150 004737 036676                JSR    PC,GENPAR   ; GENERATE PARITY BIT (EVEN) FOR
1874                                ; THE PATTERN IN R4, RETURN WITH BIT IN R2
1875 006154 020502                CMP    R5,R2      ; CORRECT PARITY BIT GENERATED?
1876 006156 001006                BNE    4$         ; NO. REPORT ERROR
1877 006160 062700 000400 35:    ADD    #400,R0    ; CREATE NEXT PATTERN, INCREMENT HI-BYTE
1878 006164 005237 003054                INC    COUNT0     ; GO BACK AND CHECK NEXT PATTERN
1879 006170 001356                BNE    2$         ;
1880 006172 000417                BR     13$        ;
1881                                ; CORRECT PARITY BIT WAS NOT
1882                                ; GENERATED FOR PATTERN HELD IN R4
1883 006174 105737 003064 45:    TSTB   ERCNT0     ; REPORT 5 ERRORS MAXIMUM, OF THIS KIND
1884 006200 001414                BEQ    13$        ; ABORT THE SUB-TEST
1885 006202 105237 003064                INCB   ERCNT0
1886 006206 010203                MOV    R2,P3     ; GET EXPECTED PARITY BIT AND
1887 006210 000241                CLC                    ; ROTATE IT TO BIT 0 POSITION
1888 006212 106103                ROLB   R3
1889 006214 106103                ROLB   R3
1890 006216 010437 001162                MOV    R4,$REG0   ; SAVE PATTERN FOR WHICH PARITY WAS GENERATED
1891 006222 010337 001164                MOV    R3,$REG1   ; SAVE EXPECTED PARITY BIT FOR THAT PATTERN
1892 006226 104015                ERROR  15         ; INCORRECT PARITY BIT WAS GENERATED
1893                                ; FOR THE PATTERN GIVEN IN ERROR
1894                                ; MESSAGE. NORMALLY, EVEN PARITY
1895                                ; IS GENERATED.
1896
1897
1898
1899 006230 000753 55:    BR     3$
1900
1901
1902
1903 006232 005737 003046 13$:    TST    TMO
1904                                ; DONE CHECKING OF HI-BYTE PARITY
1905                                ; GENERATOR USING ODD PATTERN IN
1906                                ; LO BYTE?
1907 006236 001007                BNE    TST10     ; YES, EXIT
1908 006240 062737 000002 003046        ADD    #2,TMO
1909 006246 000137 006074                JMP    20$
1910                                ; NO. CHECK HI-BYTE PARITY GENERATOR
1911                                ; USING ODD PATTERN (376) FOR
1912 006252 000377 377                FARPAT: .WORD    ; LO BYTE
1913 006254 000376 376                .WORD    ; LO-BYTE WILL BE THESE EVEN AND ODD
1914                                ; PATTERNS WHILE HI-BYTE IS BEING
1915                                ; CHECKED WITH ALL PATTERNS 0-377
1916
1917                                ; *****
1917                                ; *TEST 10 CHECK PARITY GENERATOR FOR LO BYTE DATA
```

```

1918 ; THIS TEST CHECKS OUT THE PARITY GENERATOR CHIP CORRESPONDING
1919 ; TO THE LO-BYTE (AND ASSOCIATED CIRCUIT). IN THIS TEST LO BYTE
1920 ; REFERS TO DATA BITS (9,8,7,6,5,4,3,2).
1921 ; HI BYTE REFERS TO DATA BITS (15,14,13,12,11,10,1,0).
1922 ; A PATTERN IS WRITTEN INTO THE ADDRESS REGISTER AND THE
1923 ; PARITY BIT GENERATED FOR THAT PATTERN (PARGEN) IS READ
1924 ; FROM THE STATUS REGISTER AND CHECKED IF IT IS CORRECT.
1925 ; SEQUENCE OF TESTING:
1926 ; DATA PATH USED FOR HI-BYTE IS KEPT CONSTANT AT 377
1927 ; (EVEN PATTERN) AND A COUNT PATTERN 0-377 IS USED
1928 ; FOR THE LO-BYTE. THE PARITY BIT IS CHECKED FOR
1929 ; CORRECTNESS. IF A WRONG PARITY WAS GENERATED AN
1930 ; AN ERROR MESSAGE IS REPORTED. THE FAULT COULD BE
1931 ; IN THE PARITY GENERATOR CHIPS, GATING CHIP OR THE ASSOCIATED
1932 ; DATA PATH.
1933
1934 ; NOT MORE THAN 5 ERROR MESSAGES OF THE SAME KIND
1935 ; (ABOUT INCORRECT PARITY) ARE REPORTED. IF MORE THAN
1936 ; FIVE ERRORS OCCUR THE SUB-TEST IS ABORTED.
1937
1938 ; THE ENTIRE SUB-TEST IS REPEATED WITH A CONSTANT
1939 ; ODD PATTERN OF 376 IN THE HIGH BYTE AND A
1940 ; COUNT PATTERN 0-377 FOR THE LO-BYTE.
1941 ; *****
1942 006256 000004 TST10: SCOPE
1943 006260 012737 006444 003056 MOV #TST11,NXTST ; STARTING ADDRESS OF NEXT TEST
1944 006266 005037 003046 CLR TMPO ; IF TMPO=0, USE EVEN PATTERN (377) FOR
1945 ; HI BYTE.
1946 ; IF TMPO=2, USE ODD PATTERN (376)
1947 ; FOR HI BYTE.
1948 ; FOR BOTH CASES LO-BYTE PATTERN
1949 ; CHANGES FROM 0 TO 377.
1950 006272 012777 000020 174534 MOV #PARDIS,@WCSST ; SET PARITY ERROR DISABLE BIT
1951 006300 013704 003046 20$: MOV TMPO,R4
1952 006304 016400 006252 MOV PARPAT(R4),R0 ; INITIALIZE THE PATTERN
1953 006310 000300 SWAB R0
1954 006312 012737 006320 001110 MOV #15,$LPERR ; LOOP TO THIS POINT ON ERROR
1955 006320 112737 177773 003064 1$: MOVB #-5,ERCNT0 ; DO NOT TYPE MORE THAN 5 ERRORS
1956 006326 010004 2$: MOV R0,R4
1957
1958
1959 006330 004737 036734 JSR PC,XFRPGN ; MAP INPUT PATTERN TO CHIP COMPATIBLE PATTERN
1960 006334 010477 174476 MOV R4,@WCSAR ; LOAD PATTERN INTO ADDRESS REGISTER
1961 006340 017705 174470 MOV @WCSST,R5 ; READ STATUS REGISTER AND
1962 ; 'PARGEN'-PARITY BIT.
1963 006344 042705 177577 BIC #177777-PARGEN,R5 ; MASK OUT ALL BUT PARITY BIT
1964
1965
1966
1967 006350 004737 036676 JSR PC,GENPAR ; GENERATE PARITY BIT (EVEN) FOR
1968 ; THE PATTERN IN R4. RETURN WITH BIT IN R2
1969 006354 020502 CMP R5,R2 ; CORRECT PARITY BIT GENERATED?
1970 006356 001003 BNE 4$ ; NO, REPORT ERROR
1971 006360 105200 3$: INCB R0 ; CREATE NEXT PATTERN. INCREMENT
1972 ; LO-BYTE
1973 006362 001361 BNE 2$ ; GO BACK IF NOT DONE

```



```

1974 006364 000417 BR 13$ ;CORRECT PARITY BIT WAS NOT GENERATED
1975 ;FOR PATTERN HELD IN R4
1976 ;REPORT 5 ERRORS MAXIMUM, OF THIS KIND
1977 006366 105737 003064 4$: TSTB ERCNT0 ;ABORT SUB-TEST
1978 006372 001414 BEQ 13$
1979 006374 105237 003064 INCB ERCNT0
1980 006400 010203 MOV R2,R3 ;GET EXPECTED PARITY BIT AND
1981 006402 000241 CLC ;RCTATE IT TO BIT 0 POSITION.
1982 006404 106103 ROLB R3 ;***
1983 006406 106103 ROLB R3
1984 006410 010437 001162 MOV R4,$REG0 ;SAVE PATTERN FOR WHICH PARITY
1985 ;WAS GENERATED.
1986 006414 010337 001164 MOV R3,$REG1 ;SAVE EXPECTED PARITY BIT FOR
1987 ;THAT PATTERN.
1988 006420 104015 ERROR 15 ;INCORRECT PARITY BIT WAS
1989 ;GENERATED (AND READ) FOR
1990 ;THE PATTERN GIVEN IN ERROR
1991 ;MESSAGE. NORMALLY, EVEN
1992 ;PARITY IS GENERATED.
1993
1994
1995
1996 006422 000756 5$: BR 3$
1997
1998
1999
2000 006424 005737 003046 13$: TST TMPO ;DONE CHECKING OF LO-BYTE
2001 ;PARITY GENERATOR USING ODD
2002 ;PATTERN IN HI-BYTE?
2003 006430 001005 BNE TST11 ;:YES, EXIT
2004 006432 062737 000002 003046 ADD #2, TMPO
2005 006440 000137 006300 JMP 20$ ;****IF NOT, CHECK LO-BYTE PARITY
2006 ;GENERATOR USING ODD PATTERN
2007 ;(376) FOR HI-BYTE.
2008
2009

```

```

2010 ;:*****
2011 ;*TEST 11 CHECK THAT WRONG PARITY CAN BE GENERATED, WHEN WWP IS SET
2012 ;THIS TEST CHECKS THAT WHEN WWP (WRITE WRONG PARITY) BIT IS SET
2013 ;IN THE STATUS REGISTER, WRONG PARITY (ODD PARITY) IS GENERATED.
2014 ;NORMALLY, EVEN PARITY IS GENERATED. 'PARDIS' (DISABLE PARITY
2015 ;ERROR TRAP) BIT IS ALSO SET, SO THAT A PARITY ERROR TRAP
2016 ;MAY NOT OCCUR WHEN WRONG PARITY IS WRITTEN. IF STILL,
2017 ;PARITY ERROR TRAP (FROM WCS) OCCURS (IE PARERR BIT IS
2018 ;SET IN STATUS REGISTER), THEN THERE IS A FAULT IN THE
2019 ;TRAP INHIBITING LOGIC (PARDIS, PARERR, JAM F/F'S ETC).
2020 ;FOUR DIFFERENT PATTERNS AND THEIR COMPLEMENTS ARE USED FOR
2021 ;GENERATING WRONG PARITY AND CHECKING THE CHECKER.
2022 ;HI BYTE LO BYTE PATTERN
2023 ;ODD EVEN 125525
2024 ;EVEN EVEN 145252
2025 ;EVEN ODD 052524
2026 ;ODD ODD 032253
2027 ;THE PATTERN IS LOADED INTO WCS ADDRESS REGISTER AND THE
2028 ;PARITY BIT GENERATED FOR THAT PATTERN IS READ FROM
2029 ;THE STATUS REGISTER.

```

```

2030 ;*****
2031 006444 000004 TST11: SCOPE
2032 006446 012737 006632 003056 MOV #TST12,NXTST ;STARTING ADDRESS OF NEXT TEST
2033 006454 012777 000060 174352 MOV #PARDIS+WWP,@WCSST ;DISABLE PARITY ERROR TRAPS
2034 ;AND WRITE WRONG PARITY.
2035 006462 005001 CLR R1 ;INITIALIZE PARITY MASK FOR
2036 ;'PARGEN'
2037 006464 012704 000200 MOV #PARGEN,R4
2038 006470 012700 006622 MOV #WRPAT,R0 ;INITIALIZE POINTER TO THE FIRST PATTERN.
2039 006474 012703 177774 MOV #-4,R3 ;SET COUNT FOR 4 PATTERNS
2040 006500 012737 006516 001110 MOV #15,$LPERR ;LOOP TO THIS POINT ON ERROR.
2041
2042 006506 012737 006604 000114 MOV #35,@#114 ;ESTABLISH PARITY TRAP VECTOR
2043
2044 006514 005002 55: CLR R2
2045 006516 011077 174314 15: MOV (R0),@WCSAR ;WRITE THE PATTERN IN ADDRESS REGISTER
2046 006522 017705 174306 MOV @WCSST,R5 ;READ STATUS REGISTER & 'PARGEN' BIT
2047 006526 042705 177577 BIC #177777-PARGEN,R5 ;MASK OUT EVERYTHING BUT 'PARGEN' BIT.
2048 006532 020105 CMP R1,R5 ;CORRECT PARITY BIT GENERATED?
2049 006534 001406 BEQ 25 ;YES
2050 006536 011037 001164 MOV (R0),$REG1 ;GET PATTERN THAT GAVE ERROR
2051 006542 017737 174266 001162 MOV @WCSST,$REG0 ;SAVE STATUS
2052 006550 104016 ERROR 16 ;INCORRECT PARITY BIT WAS GENERATED
2053 ;FOR THE PATTERN GIVEN IN THE
2054 ;ERROR MESSAGE. NOTE, THAT WWP
2055 ;(WRITE WRONG PARITY) BIT WAS SET
2056 ;IN THE STATUS REGISTER. THUS
2057 ;INSTEAD OF GENERATING EVEN
2058 ;PARITY AS USUAL, ODD PARITY
2059 ;SHOULD HAVE BEEN GENERATED
2060 006552 005702 25: TST R2 ;USED THE COMPLIMENTARY PATTERN?
2061 006554 001003 BNE 45 ;YES
2062 006556 005202 INC R2
2063 006560 005110 COM (R0) ;IF NOT, USE IT
2064 006562 000755 BR 15 ;GO AND CHECK
2065
2066 006564 074401 45: XOR R4,R1 ;PARITY BIT MASK FOR THE
2067 ;NEXT PATTERN
2068 006566 005720 TST (R0)+ ;INCREMENT POINTER TO THE NEXT
2069 ;PATTERN
2070 006570 005203 INC R3
2071 006572 001350 BNE 55 ;IF NOT, GO BACK AND CHECK
2072 006574 012737 003344 000114 MOV #BADPAR,@#114
2073 006602 000413 BR TST12 ;EXIT
2074
2075 006604 35: ;ENTER HERE IF AN UNEXPECTED PARITY
2076 ;ERROR TRAP TO 114 OCCURRED
2077 ;***
2078 006604 017737 174224 001162 MOV @WCSST,$REG0 ;SAVE WCS STATUS REGISTER
2079 006612 011037 001164 MOV (R0),$REG1 ;SAVE PATTERN THAT WAS WRITTEN
2080 006616 104017 ERROR 17 ;UNEXPECTED PARITY ERROR TRAP
2081 ;TOOK PLACE.
2082 ;NOTE, WCS PARITY ERROR TRAPS
2083 ;HAVE BEEN DISABLED BY SETTING
2084 ;'PARDIS' BIT IN STATUS REGISTER.
2085 ;IF THIS ERROR OCCURS AND 'PAPEPR'

```



```

2086                                     ;BIT IS SET IN STATUS REGISTER,
2087                                     ;THEN THERE IS A FAULT IN THE
2088                                     ;PARITY-TRAP-DISABLING LOGIC
2089                                     ;(PARDIS, PARERR, JAM F/FS)
2090
2091 006620 000754                       BR      25      ;CONTINUE
2092
2093                                     ;THIS TABLE CONTAINS THE PATTERNS
2094                                     ;USED FOR WRITING WRONG PARITY
2095                                     ;AND CHECKING THE PARITY GENERATION
2096 006622 125525                       WRPAT: .WORD 125525 ;HI BYTE ODD, LO BYTE EVEN, PARITY =0
2097 006624 145252                       .WORD 145252 ;HI BYTE EVEN, LO BYTE EVEN, PARITY=1
2098 006626 052524                       .WORD 052524 ;HI BYTE EVEN, LO BYTE ODD, PARITY=0
2099 006630 032253                       .WORD 032253 ;HI BYTE ODD, LO BYTE ODD, PARITY=1
2100
2101                                     ;GENERATION OF PARITY BIT MASK (R1)
2102                                     ;RELIES ON THE FACT THAT THE 4
2103                                     ;PATTERNS GIVE PARITY BITS 0,1,0,1
2104                                     ;RESPECTIVELY.
2105
2106
2107                                     ;*****
2108 *TEST 12      CHECK TRAP ON PARITY ERROR, TRAP INHIBITING, AND PARITY ERROR CLEARING LOGIC
2109
2110                                     ;THIS TEST CHECK OUT THE TRAP-ON-PARITY-ERROR LOGIC,
2111                                     ;THE PARITY-ERROR-TRAP-INHIBITING LOGIC, AND PARITY-ERROR
2112                                     ;CLEARING LOGIC.
2113                                     ;PARITY ERROR TRAPS ARE INHIBITED (SETTING 'PARDIS') AND
2114                                     ;A PATTERN IS WRITTEN INTO LOCATION 0. WWP IS SET SO WRONG (ODD)
2115                                     ;PARITY IS GENERATED AND STORED. LOCATION 0 IS READ, IF A
2116                                     ;TRAP OCCURS, THE FAULT (IN THE TRAP-INHIBITING LOGIC-PARDIS) IS
2117                                     ;REPORTED.
2118                                     ;'PARDIS' IS CLEARED AND LOCATION 0 IS READ AGAIN, FORCING
2119                                     ;A PARITY ERROR TRAP THIS TIME. IF A TRAP DID NOT
2120                                     ;OCCUR THE FAULT IS REPORTED.
2121                                     ;IF THE TRAP OCCURRED 'PARERR' BIT (15) IS CHECKED TO SEE
2122                                     ;IF IT WAS SET. IT IS ALSO CHECKED WHETHER THE CORRECT PARITY
2123                                     ;BIT WAS STORED. THEN, 'PARDIS' BIT IS SET AND IT IS CHECKED
2124                                     ;IF 'PARERR' BIT GOT CLEARED.
2125                                     ;IF A PARITY ERROR TRAP DID NOT OCCUR ON READING WRONG
2126                                     ;PARITY 'PARERR' IS CHECKED TO SEE IF THE PARITY ERROR
2127                                     ;WAS AT LEAST DETECTED. IF 'PARERR' IS NOT SET, THEN
2128                                     ;'PO' PARITY BIT STORED FOR LOCATION 0 IS CHECKED TO SEE
2129                                     ;IF IT WAS STORED CORRECTLY. IF IT WAS NOT THEN IT IS
2130                                     ;CHECKED IF THE PARITY GENERATOR CAN GENERATE CORRECT
2131                                     ;PARITY BIT FOR THAT PATTERN.
2132                                     ;THE ABOVE PROCEDURE SHOULD PROVIDE OPTIMUM FAULT RESOLUTION.
2133                                     ;THE ENTIRE TEST IS REPEATED WITH A SECOND PATTERN.
2134 006632 000004                       TST12: SCOPE
2135 006634 012737 007552 003056         MOV     #TST13,NXTST ;STARTING ADDRESS OF NEXT TEST
2136
2137 006642 012700 007542                 MOV     #TPATRN,R0  ;INITIALIZE PATTERN
2138 006646 012701 007546                 MOV     #TSTAT,R1  ;INITIALIZE STATUS REGISTER CONTENTS
2139 006652 012702 000200                 MOV     #PARGEN,R2 ;INITIALIZE PARITY GENERATED AND
2140 006656 012703 000002                 MOV     #PARO,R3   ;PARITY STORED BITS

```

CHECK TRAP ON PARITY ERROR, TRAP INHIBITING, AND PARITY ERROR CLEARING LO

```

2141
2142 006662 012777 000060 174144 15:  MOV    #PARDIS+WMP, @WCSST ;WRITE WRONG PARITY, DISABLE TRAPS
2143 006670 012737 007426 000114      MOV    #14$, @#114      ;UNEXPECTED TRAP SERVICE
2144 006676 012737 006662 001110      MOV    #15$, $LPERR    ;LOOP-BACK POINT
2145
2146 006704 012777 000000 174124 25:  MOV    #0, @WCSAR      ;WRITE PATTERN INTO LOCATION 0
2147 006712 011077 174122      MOV    (R0), @WCSDR    ;OF WCS
2148
2149 006716 011137 037252      MOV    (R1), CESTAT    ;CHECK IF THE CONTENTS OF STATUS REGISTER
2150 006722 012737 000014 037256      MOV    #PAR1+PAR2, CMASK;ARE OK
2151 006730 012737 006756 037260      MOV    #3$, CRETRN    ;RETURN ADDRESS FOR SUBROUTINE
2152 006736 004737 037152      JSR    PC, CHKSTAT
2153 006742 012737 000000 001166      MOV    #0, $REG2      ;IF ERROR, SAVE LOCATION ADDRESS
2154 006750 011037 001170      MOV    (R0), $REG3    ;SAVE PATTERN WRITTEN IN THAT LOC
2155 006754 104011      ERROR  11             ;A PATTERN WAS WRITTEN INTO LOCATION
2156                                     ;0 OF WCS, THEN STATUS REGISTER
2157                                     ;WAS CHECKED TO SEE IF CORRECT BITS
2158                                     ;WERE SET, THEY WERE NOT. ERROR MESSAGE
2159                                     ;GIVES THE INFO.
2160
2161 006756 012777 000000 174052 35:  MOV    #0, @WCSAR      ;ADDRESS LOC 0 AND READ IT
2162 006764 005777 174050      TST    @WCSDR          ;SINCE 'PARDIS' IS SET A TRAP
2163 006770 000240      NOP                    ;SHOULD NOT OCCUR. FAULT, IF IT DOES.
2164
2165 006772 012737 007002 001110      MOV    #4$, $LPERR    ;LOOP ON ERROR POINT
2166 007000 000403      BR     5$
2167
2168 007002 052777 000020 174024 45:  BIS    #PARDIS, @WCSST ;IF LOOPING OR ERROR, CLEAR UP 'PARERR'
2169                                     ;BY SETTING 'PARDIS'
2170
2171 007010 012737 007044 000114 55:  MOV    #6$, @#114     ;PARITY ERROR TRAP VECTOR
2172 007016 042777 000020 174010      BIC    #PARDIS, @WCSST ;CLEAR 'PARDIS' SO THAT PARITY TRAP CAN
2173                                     ;BE FORCED
2174 007024 012777 000000 174004      MOV    #0, @WCSAR      ;READ LOC 0 (BAD PARITY IS THERE) AND
2175 007032 005777 174002      TST    @WCSDR          ;FORCE PARITY ERROR TRAP
2176 007036 000240      NOP                    ;PARITY ERROR TRAP SHOULD OCCUR
2177 007040 000240      NOP                    ;BECAUSE LOC WITH BAD PARITY WAS READ
2178
2179 007042 000450      BR     9$             ;ERROR, IF PARITY TRAP DID NOT OCCUR
2180
2181                                     ;LOCATION WITH BAD PARITY WAS READ
2182                                     ;AND TRAP OCCURRED. ENTER THIS
2183                                     ;CODE UPON TRAP.
2184                                     ;FOR CERTAIN FAULTS IN TRAP LOGIC, THE
2185                                     ;MACHINE CAN HANG
2186 007044 022626 000000 000114 65:  CMP    (SP)+, (SP)+   ;POP THE STACK
2187 007046 012737 007460 000114      MOV    #TRP3, @#114   ;UNEXPECTED TRAP SERVICE
2188 007054 005777 173754      TST    @WCSST         ;DID 'PARERR' SET?
2189 007060 100404 000000 000000      BMI    7$             ;YES
2190 007062 017737 173746 001162      MOV    @WCSST, $REG0  ;NO, SAVE STATUS REGISTER
2191 007070 104020      ERROR  20             ;'PARERR' BIT(15) DID NOT SET IN
2192                                     ;STATUS REGISTER ON FORCING A
2193                                     ;PARITY ERROR TRAP.
2194 007072 012777 000000 173736 75:  MOV    #0, @WCSAR      ;READ THE PARITY BIT STORED FOR
2195 007100 017705 173730      MOV    @WCSST, R5     ;PATTERN IN LOC 0
2196 007104 042705 177775      BIC    #177775, R5

```









CHECK TRAP ON PARITY ERROR, TRAP INHIBITING, AND PARITY ERROR CLEARING LOGIC

```

2309                                     ;WAS FORCED BY READING BAD PARITY)
2310 007460 011637 037034 TRP3: MOV (SP),TRAPC ;SAVE PC WHERE TRAP OCCURRED
2311 007464 022626 CMP (SP)+,(SP)+ ;POP STACK
2312 007466 004737 037000 JSR PC,TRPMSG ;GO TYPE OUT PC WHERE TRAP
2313                                     ;OCCURRED
2314 007472 000405 BR EXIT1
2315
2316 007474 TRP2:                                     ;THIS TRAP OCCURRED WHILE CHECKING
2317                                     ;WAS BEING DONE ON THE FAULT
2318                                     ;CONDITION -- NO TRAP ON READING
2319                                     ;BAD PARITY.
2320 007474 011637 037034 MOV (SP),TRAPC ;SAVE PC WHERE TRAP OCCURRED.
2321 007500 022626 CMP (SP)+,(SP)+ ;POP STACK
2322 007502 004737 037000 JSR PC,TRPMSG ;GO TYPE OUT PC WHERE TRAP
2323                                     ;OCCURRED
2324 007506 104406 042652 EXIT1: TYPMSG ,MSG17 ;INDICATE THAT THIS TEST IS
2325                                     ;ABORTED.
2326 007512 012777 000020 173314 EXIT2: MOV #PARDIS,@WCSST ;DISABLE TRAPS
2327 007520 012777 000000 173310 MOV #0,@WCSAR ;WRITE BACK CORRECT PARITY
2328 007526 005077 173306 CLR @WCSDR ;IN LOCATION 0
2329 007532 012737 003344 000114 MOV #BADPAR,@#114
2330 007540 000404 BR TST13 ;EXIT
2331
2332 007542 000000 TPATRN: .WORD 0
2333 007544 052524 .WORD 52524
2334
2335 007546 040262 TSTAT: 40262
2336 007550 040260 40260
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364

```

```

*****
*TEST 13 CHECK THAT BAD PARITY 'PARO' CAN FORCE A TRAP
;THIS TEST CHECKS THAT A PARITY ERROR IN THE 16-BIT FIRST SEGMENT
;(CORRESPONDING TO PARO AND WPO) OF THE 48-BIT MICRO-WORD
;WILL RESULT IN A TRAP TO 114, IF 'PARDIS' IS CLEAR.
;BAD PARITY (ODD) IS WRITTEN INTO LOCATION 0 OF THE WCS.
;THE STATUS REGISTER IS CHECKED TO SEE IF THE CORRECT PARITY
;BIT (PARO) WAS STORED.
;THEN 'PARDIS' IS CLEARED AND LOCATION 0 IS READ SO AS TO
;FORCE A PARITY ERROR TRAP. IF THE PARITY ERROR TRAP DOES
;NOT TAKE PLACE THE FAULT IS REPORTED. THEN THE PATTERN
;WHICH WAS READ FROM LOCATION 0 IS CHECKED TO SEE
;IF IT WAS READ CORRECTLY. IF THE PATTERN WAS NOT
;CORRECT THEN THE FAULT IS PROBABLY IN THE DATA STORAGE
;LOGIC (CORRESPONDING TO WPO). IF THE PATTERN WAS READ
;CORRECTLY AND THE PREVIOUS TEST PASSED (INDICATING THAT
;A PARITY ERROR TRAP CAN TAKE PLACE) THEN THE FAULT IS
;IN PARITY CHECKING AND ROUTING LOGIC (CORRESPONDING TO PARO).
;THE ENTIRE TEST IS REPEATED FOR THE FOLLOWING PATTERNS.

;PATTERN HI BYTE LO BYTE PARO OUTPUT OF 'AND-OR' GATE SHOULD BE
;125525 ODD EVEN LO(0) LO(0)
;145252 EVEN EVEN HI(1) LO(0)
;052524 EVEN ODD LO(0) LO(0)

```

```

2365 ;032253 ODD ODD HI(1) LO(0)
2366 ;*****
2367 007552 000004 TST13: SCOPE
2368 007554 012737 010076 003056 MOV #TST14,NXTST ;STARTING ADDRESS OF NEXT TEST
2369 007562 012700 006622 MOV #WRPAT,R0 ;INITIALIZE THE PATTERN TO BE USED
2370 007566 012701 040260 MOV #40260,R1 ;INITIALIZE EXPECTED STATUS
2371 007572 012702 177774 MOV #-4,R2
2372
2373 007576 012737 007604 001110 15: MOV #25,SLPERR ;LOOP ON ERROR POINT
2374
2375 007604 012777 000060 173222 25: MOV #PARDIS+WWP,@WCSST ;DISABLE TRAPS, WRITE WRONG PARITY
2376 007612 012737 010024 000114 MOV #65,@#114 ;UNEXPECTED TRAP SERVICE
2377
2378 007620 012777 000000 173210 MOV #0,@WCSAR ;WRITE PATTERN WITH WRONG PARITY (ODD)
2379 007626 011077 173206 MOV (R0),@WCSDR ;INTO LOCATION 0
2380
2381 007632 010137 037252 MOV R1,CESTAT ;CHECK IF THE BITS IN STATUS
2382 007636 012737 000014 037256 MOV #PAR1+PAR2,CMASK;REGISTER ARE SET CORRECTLY
2383 007644 012737 007672 037260 MOV #35,CRETRN
2384 007652 004737 037152 JSR PC,CHKSTAT
2385 007656 012737 000000 001166 MOV #0,$REG2 ;SAVE LOCATION THAT WAS WRITTEN
2386 007664 011037 001170 MOV (R0),$REG3 ;SAVE PATTERN THAT WAS WRITTEN
2387 007670 104011 ERROR 11
2388
2389 007672 012737 010044 000114 35: MOV #75,@#114 ;SET TRAP VECTOR FOR PARITY ERROR
2390 007700 042777 000020 173126 BIC #PARDIS,@WCSST ;CLEAR 'PARDIS' SO THAT PARITY ERROR
2391 ;TRAP CAN OCCUR.
2392 007706 012777 000000 173122 MOV #0,@WCSAR ;FORCE A PARITY ERROR TRAP BY
2393 007714 017705 173120 MOV @WCSDR,R5 ;READING LOCATION 0 (BAD PARITY)
2394 007720 000240 NOP ;PARITY ERROR TRAP SHOULD OCCUR BECAUSE
2395 007722 000240 NOP ;LOCATION 0 CONTAINS BAD PARITY
2396
2397 ;IF A TRAP DID NOT OCCUR, REPORT
2398 ;THE FAULT
2399
2400 007724 012737 010024 000114 MOV #65,@#114 ;UNEXPECTED TRAP SERVICE
2401 007732 012737 000000 001162 MOV #0,$REG0 ;SAVE LOCATION WHERE READ WAS DONE
2402 007740 011037 001164 MOV (R0),$REG1 ;SAVE EXPECTED PATTERN
2403 007744 017737 173064 001166 MOV @WCSST,$REG2 ;SAVE STATUS
2404 007752 104022 ERROR 22 ;PARITY ERROR TRAP DID NOT
2405 ;OCCUR WHEN LOCATION 0 CONTAINING
2406 ;BAD PARITY WAS READ
2407
2408 007754 012737 000004 001110 MOV #4,SLPERR ;LOOP ON ERROR POINT
2409
2410 007762 020510 45: CMP R5,(R0) ;DATA READ WAS CORRECT?
2411 007764 001410 BEQ 53 ;YES
2412 007766 012737 000000 001162 MOV #0,$REG0 ;SAVE LOCATION WHERE READ WAS DONE
2413 007774 011037 001164 MOV (R0),$REG1 ;SAVE EXPECTED DATA
2414 010000 010537 001166 MOV R5,$REG2 ;SAVE DATA RECEIVED
2415 010004 104012 ERROR 12 ;DATA ERROR ON DOING A READ
2416 ;FROM LOCATION 0. ALSO, PARITY ERROR
2417 ;TRAP DID NOT OCCUR ON READING.
2418
2419 010006 005720 55: TST (R0)+ ;INCREMENT POINTER TO NEXT PATTERN
2420 ;TO BE USED

```



```

2421 010010 012705 000002      MOV    #PAR0,R5
2422 010014 074501      XOR    R5,R1      ;GENERATE EXPECTED STATUS FOR NEXT
2423                          ;PATTERN
2424 010016 005202      INC    R2          ;DONE?
2425 010020 001266      BNE   1$
2426
2427 010022 000425      BR    9$          ;EXIT
2428
2429 010024 011637 037034      6$:  MOV    (SP),TRAPC ;AN UNEXPECTED TRAP TO 114
2430 010030 C22626      CMP    (SP)+,(SP)+ ;OCCURRED. REPORT PC AT WHICH
2431 010032 004737 037000      JSR   PC,TRPMSG  ;THE TRAP OCCURRED
2432 010036 104406 042652      TYPMSG ,MSG17    ;REPORT THIS TEST ABORTED
2433 010042 000401      BR    8$
2434
2435 010044 022626      7$:  CMP    (SP)+,(SP)+ ;ENTER HERE, IF A VALID TRAP TO
2436                          ;114 OCCURRED ON READING LOC
2437                          ;0 HAVING BAD PARITY
2438
2439 010046 012777 000020 172760      8$:  MOV    #PARDIS,@WCSST ;INHIBIT FURTHER TRAPS
2440 010054 012777 000000 172754      MOV    #0,@WCSAR   ;WRITE BACK GOOD PARITY INTO
2441 010062 005077 172752      CLR   @WCSDR      ;LOCATION 0
2442 010066 012737 003344 000114      MOV    #BADPAR,@#114 ;ESTABLISH UNEXPECTED PARITY TRAP
2443                          ;SERVICE (114) FOR THE PROGRAM.
2444
2445 010074 000744      BR    5$
2446 010076      9$:
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475 010076 000004      TST14: SCOPE
2476 010100 012737 010422 003056      MOV    #TST15,NXTST ;STARTING ADDRESS OF NEXT TEST

```

; \*\*\*\*\*  
; \*TEST 14 CHECK THAT BAD PARITY 'PAR1' CAN FORCE A TRAP  
; THIS TEST CHECKS THAT A PARITY ERROR IN THE 16-BIT MIDDLE SEGMENT  
; CORRESPONDING TO PAR1 AND WP1) OF THE 48-BIT MICRO-WORD  
; WILL RESULT IN A TRAP TO 114, IF 'PARDIS' IS CLEAR.  
; BAD PARITY (ODD) IS WRITTEN INTO LOCATION 2000 OF THE WCS.  
; THE STATUS REGISTER IS CHECKED TO SEE IF THE CORRECT PARITY  
; BIT (PAR1) WAS STORED.  
; THEN 'PARDIS' IS CLEARED AND LOCATION 2000 IS READ SO AS TO  
; FORCE A PARITY ERROR TRAP. IF THE PARITY ERROR TRAP DOES  
; NOT TAKE PLACE THE FAULT IS REPORTED. THEN THE PATTERN  
; WHICH WAS READ FROM LOCATION 2000 IS CHECKED TO SEE  
; IF IT WAS READ CORRECTLY. IF THE PATTERN WAS NOT  
; CORRECT THEN THE FAULT IS PROBABLY IN THE DATA STORAGE  
; LOGIC (CORRESPONDING TO WP1). IF THE PATTERN WAS READ  
; CORRECTLY AND THE PREVIOUS TEST PASSED (INDICATING THAT  
; A PARITY ERROR TRAP CAN TAKE PLACE) THEN THE FAULT IS  
; IN PARITY CHECKING AND ROUTING LOGIC (CORRESPONDING TO PAR1).  
; THE ENTIRE TEST IS REPEATED FOR THE FOLLOWING PATTERNS.  
; PATTERN HI BYTE LO BYTE PAR1 OUTPUT OF 'AND-OR' GATE SHOULD BE  
; 125525 ODD EVEN LO(0) LO(0)  
; 145252 EVEN EVEN HI(1) LO(0)  
; 052524 EVEN ODD LO(0) LO(0)  
; 032253 ODD ODD HI(1) LO(0)  
; \*\*\*\*\*

```

2477 010106 012700 006622      MOV      #WRPAT,R0      ; INITIALIZE THE PATTERN TO BE USED
2478 010112 012701 040060      MOV      #40060,R1     ; INITIALIZE EXPECTED STATUS
2479 010116 012702 177774      MOV      #-4,R2
2480
2481 010122 012737 010130 001110 15:  MOV      #25,$LPERR    ; LOOP ON ERROR POINT
2482
2483 010130 012777 000060 172676 25:  MOV      #PARDIS+WWP,@WCSST ; DISABLE TRAPS, WRITE WRONG PARITY
2484 010136 012737 010350 000114      MOV      #65,@#114     ; UNEXPECTED TRAP SERVICE
2485
2486 010144 012777 002000 172664      MOV      #2000,@WCSAR   ; WRITE PATTERN WITH WRONG PARITY (ODD)
2487 010152 011077 172662      MOV      (R0),@WCSDR   ; INTO LOCATION 2000
2488
2489 010156 010137 037252      MOV      R1,CESTAT     ; CHECK IF THE BITS IN STATUS
2490 010162 012737 000012 037256      MOV      #PAR0+PAR2,CMASK; REGISTER ARE SET CORRECTLY
2491 010170 012737 010216 037260      MOV      #35,CRETRN
2492 010176 004737 037152      JSR      PC,CHKSTAT
2493 010202 012737 002000 001166      MOV      #2000,$REG2   ; SAVE LOCATION THAT WAS WRITTEN
2494 010210 011037 001170      MOV      (R0),$REG3    ; SAVE PATTERN THAT WAS WRITTEN
2495 010214 104011      ERROR    11
2496
2497 010216 012737 010370 000114 35:  MOV      #75,@#114     ; SET TRAP VECTOR FOR PARITY ERROR
2498 010224 042777 000020 172602      BIC      #PARDIS,@WCSST ; CLEAR 'PARDIS' SO THAT PARITY ERROR
2499                                ; TRAP CAN OCCUR.
2500 010232 012777 002000 172576      MOV      #2000,@WCSAR   ; FORCE A PARITY ERROR TRAP BY
2501 010240 017705 172574      MOV      @WCSDR,R5     ; READING LOCATION 2000 (BAD PARITY)
2502 010244 000240      NOP                                ; PARITY ERROR TRAP SHOULD OCCUR BECAUSE
2503 010246 000240      NOP                                ; LOCATION 1 CONTAINS BAD PARITY
2504
2505                                ; IF A TRAP DID NOT OCCUR, REPORT
2506                                ; THE FAULT
2507
2508 010250 012737 010350 000114      MOV      #65,@#114     ; UNEXPECTED TRAP SERVICE
2509 010256 012737 002000 001162      MOV      #2000,$REG0   ; SAVE LOCATION WHERE READ WAS DONE
2510 010264 011037 001164      MOV      (R0),$REG1    ; SAVE EXPECTED PATTERN
2511 010270 017737 172540 001166      MOV      @WCSST,$REG2  ; SAVE STATUS
2512 010276 104022      ERROR    22           ; PARITY ERROR TRAP DID NOT
2513                                ; OCCUR WHEN LOCATION 2000 CONTAINING
2514                                ; BAD PARITY WAS READ
2515
2516 010300 012737 000004 001110      MOV      #4,$LPERR     ; LOOP ON ERROR POINT
2517
2518 010306 020510      45:  CMP      R5,(R0)       ; DATA READ WAS CORRECT?
2519 010310 001410      BEQ      55           ; YES
2520 010312 012737 002000 001162      MOV      #2000,$REG0   ; SAVE LOCATION WHERE READ WAS DONE
2521 010320 011037 001164      MOV      (R0),$REG1    ; SAVE EXPECTED DATA
2522 010324 010537 001166      MOV      R5,$REG2     ; SAVE DATA RECEIVED
2523 010330 104012      ERROR    12           ; DATA ERROR ON DOING A READ
2524                                ; FROM LOCATION 2000. ALSO, PARITY ERROR
2525                                ; TRAP DID NOT OCCUR ON READING.
2526
2527 010332 005720      55:  TST      (R0)+        ; INCREMENT POINTER TO NEXT PATTERN
2528                                ; TO BE USED
2529 010334 012705 000004      MOV      #PAR1,R5     ; GENERATE EXPECTED STATUS FOR NEXT
2530 010340 074501      XOR      R5,R1        ; PATTERN
2531
2532 010342 005202      INC      R2           ; DONE?

```



```

2533 010344 001266      BNE      15
2534
2535 010346 000425      BR       95      ;EXIT
2536
2537 010350 011637 037034 65:     MOV      (SP),TRAPC    ;AN UNEXPECTED TRAP TO 114
2538 010354 022626      CMP      (SP)+,(SP)+  ;OCCURRED. REPORT PC AT WHICH
2539 010356 004737 037000      JSR      PC,TRPMSG    ;THE TRAP OCCURRED
2540 010362 104406 042652      TYPMSG   ,MSG17      ;REPORT THIS TEST ABORTED
2541 010366 000401      BR       85
2542
2543 010370 022626      75:     CMP      (SP)+,(SP)+  ;ENTER HERE, IF A VALID TRAP TO
2544                                     ;114 OCCURRED ON READING LOC
2545                                     ;1 HAVING BAD PARITY
2546
2547 010372 012777 000020 172434 85:     MOV      #PARDIS,@WCSST ;INH BIT FURTHER TRAPS
2548 010400 012777 002000 172430      MOV      #2000,@WCSAR  ;WRITE BACK GOOD PARITY INTO
2549 010406 005077 172426      CLR      @WCSDR        ;LOCATION 2000
2550 010412 012737 003344 000114      MOV      #BADPAR,@#114 ;ESTABLISH UNEXPECTED PARITY TRAP
2551                                     ;SERVICE (114) FOR THE PROGRAM.
2552 010420 000744      BR       55
2553
2554 010422      95:
2555

```

```

*****
*TEST 15      CHECK THAT BAD PARITY 'PAR2' CAN FORCE A TRAP
; THIS TEST CHECKS THAT A PARITY ERROR IN THE 16-BIT THIRD SEGMENT
; (CORRESPONDING TO PAR2 AND WP2) OF THE 48-BIT MICRO-WORD
; WILL RESULT IN A TRAP TO 114, IF 'PARDIS' IS CLEAR.
; BAD PARITY (ODD) IS WRITTEN INTO LOCATION 4000 OF THE WCS.
; THE STATUS REGISTER IS CHECKED TO SEE IF THE CORRECT PARITY
; BIT (PAR2) WAS STORED.
; THEN 'PARDIS' IS CLEARED AND LOCATION 4000 IS READ SO AS TO
; FORCE A PARITY ERROR TRAP. IF THE PARITY ERROR TRAP DOES
; NOT TAKE PLACE THE FAULT IS REPORTED. THEN THE PATTERN
; WHICH WAS READ FROM LOCATION 4000 IS CHECKED TO SEE
; IF IT WAS READ CORRECTLY. IF THE PATTERN WAS NOT
; CORRECT THEN THE FAULT IS PROBABLY IN THE DATA STORAGE
; LOGIC (CORRESPONDING TO WP2). IF THE PATTERN WAS READ
; CORRECTLY AND THE PREVIOUS TEST PASSED (INDICATING THAT
; A PARITY ERROR TRAP CAN TAKE PLACE) THEN THE FAULT IS
; IN PARITY CHECKING AND ROUTING LOGIC (CORRESPONDING TO PAR2).
; THE ENTIRE TEST IS REPEATED FOR THE FOLLOWING PATTERNS.
; PATTERN  HI BYTE  LO BYTE  PAR2  OUTPUT OF 'AND-OR' GATE SHOULD BE
; 125525      ODD      EVEN      LO(0)    LO(0)
; 145252      EVEN     EVEN     HI(1)    LO(0)
; 052524      EVEN     ODD      LO(0)    LO(0)
; 032253      ODD      ODD      HI(1)    LO(0)
*****

```

```

2582 TST15: SCOPE
2583 010422 000004      MOV      #TST16,NXTST  ;STARTING ADDRESS OF NEXT TEST
2584 010424 012737 010746 003056      MOV      #WRPAT,R0     ;INITIALIZE THE PATTERN TO BE USED
2585 010432 012700 006622      MOV      #40060,R1    ;INITIALIZE EXPECTED STATUS
2586 010436 012701 040060
2587 010442 012702 177774
2588

```

```

2589 010446 012737 010454 001110 15:  MOV    #25,$LPERR      ;LOOP ON ERROR POINT
2590
2591 010454 012777 000060 172352 25:  MOV    #PARDIS+WWP,@WCSST ;DISABLE TRAPS, WRITE WRONG PARITY
2592 010462 012737 010674 000114      MOV    #65,@#114      ;UNEXPECTED TRAP SERVICE
2593
2594 010470 012777 004000 172340      MOV    #4000,@WCSAR   ;WRITE PATTERN WITH WRONG PARITY (ODD)
2595 010476 011077 172336      MOV    (R0),@WCSDR   ;INTO LOCATION 4000
2596
2597 010502 010137 037252      MOV    R1,CESTAT     ;CHECK IF THE BITS IN STATUS
2598 010506 012737 000006 037256      MOV    #PAR0+PAR1,CMASK;REGISTER ARE SET CORRECTLY
2599 010514 012737 010542 037260      MOV    #35,CRETRN
2600 010522 004737 037152      JSR    PC,CHKSTAT
2601 010526 012737 004000 001166      MOV    #4000,$REG2   ;SAVE LOCATION THAT WAS WRITTEN
2602 010534 011037 001170      MOV    (R0),$REG3   ;SAVE PATTERN THAT WAS WRITTEN
2603 010540 104011      ERROR  11          ;STATUS ERROR ON WRITE TO WCS LOCATION
2604
2605 010542 012737 010714 000114 35:  MOV    #75,@#114    ;SET TRAP VECTOR FOR PARITY ERROR
2606 010550 042777 000020 172256      BIC    #PARDIS,@WCSST ;CLEAR 'PARDIS' SO THAT PARITY ERROR
2607                                ;TRAP CAN OCCUR.
2608 010556 012777 004000 172252      MOV    #4000,@WCSAR ;FORCE A PARITY ERROR TRAP BY
2609 010564 017705 172250      MOV    @WCSDR,R5    ;READING LOCATION 4000 (BAD PARITY)
2610 010570 000240      NOP
2611 010572 000240      NOP
2612                                ;PARITY ERROR TRAP SHOULD OCCUR BECAUSE
2613                                ;LOCATION 4000 CONTAINS BAD PARITY
2614                                ;IF A TRAP DID NOT OCCUR, REPORT
2615                                ;THE FAULT
2616 010574 012737 010674 000114      MOV    #65,@#114    ;UNEXPECTED TRAP SERVICE
2617 010602 012737 004000 001162      MOV    #4000,$REG0  ;SAVE LOCATION WHERE READ WAS DONE
2618 010610 011037 001164      MOV    (R0),$REG1   ;SAVE EXPECTED PATTERN
2619 010614 017737 172214 001166      MOV    @WCSST,$REG2 ;SAVE STATUS
2620 010622 104022      ERROR  22          ;PARITY ERROR TRAP DID NOT
2621                                ;OCCUR WHEN LOCATION 4000 CONTAINING
2622                                ;BAD PARITY WAS READ
2623
2624 010624 012737 000004 001110      MOV    #4,$LPERR    ;LOOP ON ERROR POINT
2625
2626 010632 020510      45:  CMP    R5,(R0)      ;DATA READ WAS CORRECT?
2627 010634 001410      BEQ    55
2628 010636 012737 004000 001162      MOV    #4000,$REG0  ;SAVE LOCATION WHERE READ WAS DONE
2629 010644 011037 001164      MOV    (R0),$REG1   ;SAVE EXPECTED DATA
2630 010650 010537 001166      MOV    R5,$REG2    ;SAVE DATA RECEIVED
2631 010654 104012      ERROR  12          ;DATA ERROR ON DOING A READ
2632                                ;FROM LOCATION 4000. ALSO, PARITY ERROR
2633                                ;TRAP DID NOT OCCUR ON READING.
2634
2635 010656 005720      55:  TST    (R0)+        ;INCREMENT POINTER TO NEXT PATTERN
2636                                ;TO BE USED
2637 010660 012705 000010      MOV    #PAR2,R5
2638 010664 074501      XOR    R5,R1
2639                                ;GENERATE EXPECTED STATUS FOR NEXT
2640                                ;PATTERN
2641                                ;DONE?
2642 010666 005202      INC    R2
2643 010670 001266      BNE   15
2644 010672 000425      BR    95

```





```

2701                                     ;BE IN THE 'DB' REGISTER OR LINES <15:0>
2702                                     ;OR BUS DIN<15:0>, PORT B0
2703 011074 005203                       4$:   INC      R3
2704
2705 011076 006300                       5$:   ASL      R0           ;CREATE NEXT PATTERN TO BE USED
2706 011100 005704                       TST      R4           ;FLOAT A 1 OR 0?
2707 011102 001402                       BEQ      6$
2708 011104 052700 000001                 BIS      #BIT0,R0     ;FLOAT A 1
2709 011110 077527                       6$:   SOB      R5,3$     ;DONE ALL 16 PATTERNS?
2710 011112 012702 000001                 MOV      #BIT0,R2
2711 011116 074204                       XOR      R2,R4       ;DONE BOTH-FLOAT A 1 AND 0?
2712 011120 001436                       BEQ      8$           ;YES
2713 011122 012700 177776                 MOV      #177776,R0  ;INITIALIZE THE PATTERN
2714 011126 000742                       BR       3$           ;DO REST OF CHECKING
2715
2716                                     ; IF A PARITY ERROR OCCURRED ENTER HERE.
2717
2718 011130 032777 100000 171676 7$:   BIT      #PARERR,@WCSST ;WCS PARITY ERROR?
2719 011136 001422                       BEQ      9$           ;NO, CHECK WHICH ONE?
2720 011140 022626                       CMP      (SP)+,(SP)+ ;POP THE STACK
2721 011142 105237 003065                 INCB    ERCNT1
2722 011146 001753                       BEQ      5$
2723 011150 017737 171660 001166         MOV      @WCSST,$REG2 ;SAVE STATUS REGISTER
2724 011156 012777 000020 171650         MOV      #PARDIS,@WCSST ;CLEAR PARITY ERROR
2725 011164 005077 171644                 CLR     @WCSST
2726 011170 005037 001162                 CLR     $REG0       ;SAVE WCS ADDRESS
2727 011174 010037 001164                 MOV     R0,$REG1    ;SAVE EXPECTED PATTERN
2728 011200 104026                       ERROR   26          ;PARITY ERROR OCCURRED ON READING
2729                                     ;LOCATION 0 OF WCS. FAULT COULD
2730                                     ;BE IN LOCATION 0, WRITE PULSE
2731                                     ;LOGIC OR PARITY GENERATOR/CHECKER
2732 011202 000735                       BR       5$          ;RETURN, DO REST OF CHECKING
2733 011204 012737 003344 000114 9$:   MOV     #BADPAR,@#114 ;IF THE PARITY ERROR WAS NOT
2734 011212 000137 003344                 JMP     BADPAR      ;FROM WCS GO TO THE NORMAL
2735                                     ;PARITY ERROR HANDLER
2736 011216                                     8$:
2737 011216 012737 003344 000114         MOV     #BADPAR,@#114 ;NORMAL PARITY ERROR HANDLER
2738
2739

```

```

2740                                     ;*****
2741                                     ;*TEST 17 CHECK WCS DATA PATH: BUS DIN (31:16), LOC2000
2742                                     ;THIS TEST PRIMARILY CHECKS OUT THE DATA PATH SEGMENT <31:16>
2743                                     ;OF THE WCS; NAMELY INPUTS TO DATA ARRAY CHIPS <31:16>,
2744                                     ;DOUT <31:16>, 'BUS DIN' MUX (PORT C0) AND LOCATION 2000
2745                                     ; (TEST LOCATION). FAULTS IN THE DBUF PATH AND WRITE PULSE
2746                                     ;LOGIC WILL ALSO BE DETECTED.
2747                                     ;FIRST, A PATTERN OF 1 IS FLOATED THROUGH LOCATION 2000. THEN
2748                                     ;A 0 (WITH A BACKGROUND OF 1'S) IS FLOATED THROUGH. AFTER
2749                                     ;WRITING A PATTERN A READ IS DONE AND COMPARED. IF A
2750                                     ;PARITY ERROR OCCURS THE FAULT IS PROBABLY IN THE DATA
2751                                     ;ARRAY OR ASSOCIATED INPUT/OUTPUT LINES.
2752                                     ;IF A DATA ERROR OCCURS, THE FAULT IS PROBABLY IN THE
2753                                     ;'BUS DIN' MUX <31:16> OR ASSOCIATED LINES.
2754                                     ;NOTE, FAULTS IN DBUF REGISTER, WRITE PULSE LOGIC, PARITY
2755                                     ;LOGIC WILL ALSO BE DETECTED HERE. BUT THEY SHOULD HAVE
2756                                     ;BEEN ALREADY DETECTED IN PREVIOUS TESTS.

```



```

2757 ;*****
2758 011224 000004 TST17: SCOPE
2759 011226 012737 011514 003056 MOV #TST20,NXTST ;STARTING ADDRESS OF NEXT TEST
2760 011234 005077 171574 CLR @WCSST
2761 011240 005003 CLR R3
2762 011242 005004 CLR R4 ;IF R4=0 FLOAT A 1; R4=1 FLOAT A 0
2763 011244 012700 000001 MOV #1,R0 ;INITIALIZE THE PATTERN
2764 011250 012737 011412 000114 MOV #7$,@#114 ;PARITY ERROR TRAP VECTOR
2765 011256 012705 000020 15: MOV #16.,R5 ;COUNT
2766 011262 012701 002000 MOV #2000,R1 ;WCS ADDRESS TO BE USED (2000)
2767 011266 012737 011274 001110 MOV #2$, $LPERR ;LOOP BACK ADDRESS
2768 011274 112737 177773 003064 25: MOVB #-5,ERCNT0 ;REPORT 5 ERRORS MAXIMUM
2769 011302 112737 177773 003065 MOVB #-5,ERCNT1
2770 011310 010177 171522 MOV R1,@WCSAR ;ADDRESS THE WCS, LOC 2000
2771 011314 010077 171520 35: MOV R0,@WCSDR ;WRITE THE PATTERN
2772 011320 017702 171514 MOV @WCSDR,R2 ;READ IT BACK
2773 011324 020002 CMP R0,R2 ;OK?
2774 011326 001414 BEQ 5$ ;YES
2775 011330 105237 003064 INCB ERCNT0 ;DONT REPORT MORE THAN 5 ERRORS
2776 011334 001410 BEQ 4$
2777 011336 012737 002000 001162 MOV #2000,$REG0 ;SAVE WCS LOCATION
2778 011344 010037 001164 MOV R0,$REG1 ;SAVE EXPECTED PATTERN
2779 011350 010237 001166 MOV R2,$REG2 ;SAVE PATTERN READ
2780 011354 104012 ERROR 12 ;DATA ERROR OCCURRED ON DOING A
2781 ;READ FROM WCS LOCATION 2000. FAULT
2782 ;COULD BE IN 'BUS DIN' MUX <31:16>
2783 ; (PORT CO), LINES <31:16>, OR DB REGISTER
2784 011356 005203 45: INC R3 ;KEEP COUNT OF ERRORS
2785 011360 006300 55: ASL R0 ;CREATE NEXT PATTERN TO BE USED
2786 011362 005704 TST R4 ;FLOAT A 1 OR 0?
2787 011364 001402 BEQ 6$
2788 011366 052700 000001 BIS #BIT0,R0 ;FLOAT A 1
2789 011372 077530 65: SOB R5,3$ ;DONE ALL 16 PATTERNS?
2790 011374 012702 000001 MOV #BIT0,R2
2791 011400 074204 XOR R2,R4 ;DONE BOTH-FLOAT A 1 AND 0?
2792 011402 001441 BEQ 8$ ;YES
2793 011404 012700 177776 MOV #177776,R0 ;INITIALIZE THE PATTERN
2794 011410 000741 BR 3$ ;DO REST OF CHECKING
2795
2796 ; IF A PARITY ERROR OCCURRED ENTER HERE
2797
2798 011412 032777 100000 171414 75: BIT #PARERR,@WCSST ;WCS PARITY ERROR?
2799 011420 001425 BEQ 9$ ;NO. CHECK WHICH ONE
2800 011422 022626 CMP (SP)+,(SP)+ ;POP THE STACK
2801 011424 105237 003065 INCB ERCNT1 ;REPORT MAXIMUM 5 ERRORS
2802 011430 001002 BNE .+6
2803 011432 000137 011360 JMP 53
2804 011436 017737 171372 001166 MOV @WCSST,$REG2 ;SAVE STATUS REGISTER
2805 011444 012777 000020 171362 MOV #PARDIS,@WCSST ;CLEAR PARITY ERROR
2806 011452 005077 171356 CLR @WCSST
2807 011456 012737 002000 001162 MOV #2000,$REG0 ;SAVE WCS ADDRESS
2808 011464 010037 001164 MOV R0,$REG1 ;SAVE EXPECTED PATTERN
2809 011470 104026 ERROR 26 ;PARITY ERROR OCCURRED ON READING
2810 ;LOCATION 2000 OF WCS. FAULT COULD
2811 ;BE IN LOC 2000 OR THE INPUT/OUTPUT
2812 ;LINES.

```



```

2813 011472 000732          BR      55          ;RETURN, DO REST OF CHECKING
2814 011474 012737 003344 000114 95:   MOV      #BADPAR,@#114 ;IF THE PARITY ERROR WAS NOT FROM
2815 011502 000137 003344          JMP      BADPAR       ;WCS GO TO THE NORMAL
2816                                ;PARITY ERROR HANDLER
2817 011506                85:      ;EXIT
2818
2819 011506 012737 003344 000114          MOV      #BADPAR,@#114 ;NORMAL PARITY ERROR HANDLER
2820
2821                                ;*****
2822                                ;*TEST 20 CHECK WCS DATA PATH: BUS DIN (47:32), LOC 4000
2823                                ;THIS TEST PRIMARILY CHECKS OUT THE DATA PATH SEGMENT <47:31>
2824                                ;OF THE WCS; NAMELY INPUTS TO DATA ARRAY CHIPS <47:32>,
2825                                ;DOUT <47:32>, 'BUS DIN' MUX (PORT DO) AND LOCATION 4000
2826                                ;(TEST LOCATION). FAULTS IN THE DBUF PATH AND WRITE
2827                                ;PULSE LOGIC WILL ALSO BE DETECTED.
2828                                ;FIRST, A PATTERN OF 1 IS FLOATED THROUGH LOCATION 4000. THEN
2829                                ;A 0 (WITH A BACKGROUND OF 1'S) IS FLOATED THROUGH. AFTER
2830                                ;WRITING A PATTERN A READ IS DONE AND COMPARED. IF A
2831                                ;PARITY ERROR OCCURS THE FAULT IS PROBABLY IN THE DATA ARRAY
2832                                ;(LOCATION 4000) OR ASSOCIATED INPUT OUTPUT LINES.
2833                                ;IF A DATA ERROR OCCURS, THE FAULT IS PROBABLY IN THE
2834                                ;'BUS DIN' MUX <47:32> OR ASSOCIATED LINES. NOTE,
2835                                ;FAULTS IN DBUF REGISTER, WRITE PULSE LOGIC, PARITY
2836                                ;LOGIC WILL ALSO BE DETECTED HERE. BUT THEY SHOULD HAVE
2837                                ;BEEN ALREADY DETECTED IN THE PREVIOUS TESTS.
2838                                ;*****
2839 011514 000004          TST20: SCOPE
2840 011516 012737 012000 003056          MOV      #TST21,NXTST  ;STARTING ADDRESS OF NEXT TEST
2841 011524 005077 171304          CLR      @WCSST
2842 011530 005003          CLR      R3
2843 011532 005004          CLR      R4          ;IF R4=0 FLOAT A 1; R4=1 FLOAT A 0
2844 011534 012700 000001          MOV      #1,R0       ;INITIALIZE THE PATTERN
2845 011540 012737 011702 000114          MOV      #75,@#114   ;PARITY ERROR TRAP VECTOR
2846 011546 012705 000020 15:      MOV      #16,R5      ;COUNT
2847 011552 012701 004000          MOV      #4000,R1    ;WCS ADDRESS TO BE USED (4000)
2848 011556 012737 011564 001110          MOV      #25,$LPERR  ;LOOP BACK ADDRESS
2849 011564 112737 177773 003064 25:   MOV      #-5,ERCNT0  ;REPORT 5 ERRORS MAXIMUM
2850 011572 112737 177773 003065          MOV      #-5,ERCNT1
2851 011600 010177 171232          MOV      R1,@WCSAR   ;ADDRESS THE WCS, LOCATION 4000
2852 011604 010077 171230 35:      MOV      R0,@WCSDR   ;WRITE THE PATTERN
2853 011610 017702 171224          MOV      @WCSDR,R2   ;READ IT BACK
2854 011614 020002          CMP      R0,R2       ;OK?
2855 011616 071414          BEQ      55          ;YES
2856 011620 105237 003064          INCB    ERCNT0       ;DONT REPORT MORE THAN 5 ERRORS
2857 011624 001410          BEQ      45
2858 011626 012737 004000 001162          MOV      #4000,$REG0 ;SAVE WCS LOCATION
2859 011634 010037 001164          MOV      R0,$REG1   ;SAVE EXPECTED PATTERN
2860 011640 010237 001166          MOV      R2,$REG2   ;SAVE PATTERN READ
2861 011644 104012          ERROR   12          ;DATA ERROR OCCURRED ON DOING A
2862                                ;READ FROM WCS LOCATION 4000. FAULT
2863                                ;COULD BE IN 'BUS DIN' MUX <47:32>
2864                                ;(PORT DO), LINES <47:32>, OR DB REGISTER
2865 011646 005203          45:      INC      R3          ;KEEP COUNT OF ERRORS
2866
2867 011650 006300          55:      ASL      R0          ;CREATE NEXT PATTERN TO BE USED
2868 011652 005704          TST     R4          ;FLOAT A 1 OR 0?

```



```
2869 011654 001402 BEQ 65
2870 011656 052700 000001 BIS #BIT0,R0 ;FLOAT A 1
2871
2872 011662 077530 65: SOB R5,35 ;DONE ALL 16 PATTERNS?
2873 011664 012702 000001 MOV #BIT0,R2
2874 011670 074204 XOR R2,R4 ;DONE BOTH-FLOAT A 1 AND 0?
2875 011672 001437 BEQ 85 ;YES
2876 011674 012700 177776 MOV #177776,R0 ;INITIALIZE THE PATTERN
2877 011700 000741 BR 35 ;DO REST OF CHECKING
2878
2879 ; IF A PARITY ERROR OCCURRED ENTER HERE
2880
2881 011702 032777 100000 171124 75: BIT #PARERR,@WCSST ;WCS PARITY ERROR?
2882 011710 001757 BEQ 55 ;NO, CHECK WHICH ONE?
2883 011712 105237 003065 INCB ERCNT1 ;REPORT MAXIMUM 5 ERRORS
2884 011716 001420 BEQ 95
2885 011720 022626 CMP (SP)+,(SP)+ ;POP THE STACK
2886 011722 017737 171106 001166 MOV @WCSST,%REG2 ;SAVE STATUS REGISTER
2887 011730 012777 000020 171076 MOV #PARDIS,@WCSST ;CLEAR PARITY ERROR
2888 011736 005077 171072 CLR @WCSST
2889 011742 012737 004000 001162 MOV #4000,%REG0 ;SAVE WCS ADDRESS
2890 011750 010037 001164 MOV R0,%REG1 ;SAVE EXPECTED PATTERN
2891 011754 104026 ERROR 26 ;PARITY ERROR OCCURRED ON READING
2892 ;LOCATION 40000 OF WCS. FAULT COULD
2893 ;BE IN LOC 4000 OR THE INPUT/OUTPUT
2894 ;LINES.
2895 011756 000734 BR 55 ;RETURN, DO REST OF CHECKING
2896 011760 012737 003344 000114 95: MOV #BADPAR,@#114 ;IF THE PARITY ERROR WAS NOT FROM
2897 011766 000137 003344 JMP BADPAR ;WCS GO TO THE NORMAL PARITY
2898 ;ERROR HANDLER
2899
2900 011772 85: ;EXIT
2901 011772 012737 003344 000114 MOV #BADPAR,@#114 ;NORMAL PARITY ERROR HANDLER
2902
2903 ;*****
2904 ;*TEST 21 CHECK THE THREE HI-BYTE PARITY CHECKERS
2905 ;THIS TEST CHECKS THE THREE HI-BYTE PARITY CHECKERS, THAT THEY
2906 ;CAN DETECT BAD PARITY WHEN A WCS LOCATION IS READ.
2907 ;THE TEST CHECKS THE THREE HI-BYTE PARITY CHECKERS CORRESPONDING
2908 ;TO THE THREE SEGMENTS OF A 48 BIT MICRO-WORD.
2909 ;SEGMENT 0 CORRESPONDS TO 'PAR0' AND 'WPO', THE 16 BIT FIELD
2910 ;CORRESPONDING TO WCS LOCATIONS 0,1,2,3,... 1777(OCTAL)
2911 ;SEGMENT 1 CORRESPONDS TO 'PAR1' AND 'WP1', THE 16 BIT FIELD
2912 ;CORRESPONDING TO WCS LOCATIONS 2000,2001,2002,... 3777(OCTAL).
2913 ;SEGMENT 2 CORRESPONDS TO 'PAR2' AND 'WP2', THE 16-BIT FIELD
2914 ;CORRESPONDING TO WCS LOCATIONS 4000,4001,4002,... 5777(OCTAL).
2915 ;LOCATIONS 0,2000, 4000 ARE USED AS TEST LOCATIONS WHERE PATTERNS ARE STORED
2916 ;AND READ FROM, AND THE PARITY CHECKED. BAD PARITY IS GENERATED AND
2917 ;STORED USING 'WRITE WRONG PARITY' WHEN THE BAD PARITY IS
2918 ;READ, THE PARITY CHECKERS SHOULD DETECT IT AND CAUSE A TRAP.
2919 ;A COUNT PATTERN FROM 0 TO 377 IS USED TO TEST THE
2920 ;8-BIT PARITY CHECKERS (HI BYTE). MEANWHILE, THE LO-BYTE PATTERN
2921 ;IS KEPT CONSTANT AT EVEN (377) AND ODD (376). IF THE
2922 ;BAD PARITY IS NOT DETECTED FOR A PARTICULAR PATTERN AN ERROR
2923 ;MESSAGE IS REPORTED.
2924 ;THE ENTIRE TEST IS REPEATED FOR EACH OF THE THREE
```

```

2925 ; HI BYTE PARITY CHECKERS USING TEST LOCATIONS 0,2000,4000.
2926 ; *****
2927 012000 000004 TST21: SCOPE
2928 012002 012737 012434 003056 MOV #TST22,NXTST ; STARTING ADDRESS OF NEXT TEST
2929
2930 012010 005037 003046 CLR TMPO ; IF TMPO=0, USE EVEN PATTERN (377) FOR LO BYTE
2931 ; IF TMPO=2, USE ODD PATTERN (376) FOR LO BYTE
2932 ; FOR BOTH CASES, VARY HI-BYTE FROM 0 TO 377
2933
2934 012014 005001 CLR R1 ; INITIALIZE WCS LOCATION, CORRESPONDING
2935 ; TO WHICH THE PARITY CHECKER WILL BE
2936 ; CHECKED. NOTE, THERE ARE THREE SETS OF
2937 ; PARITY CHECKERS:
2938 ; FOR SEGMENT 0 (LOCATIONS 0,1,2,...)
2939 ; FOR SEGMENT 1 (LOCATIONS 2000,2001,...)
2940 ; FOR SEGMENT 2 (LOCATIONS 4000,4001,...)
2941
2942 012016 012702 000002 MOV #PAR0,R2
2943 012022 010237 003050 MOV R2,TMP1
2944 012026 013704 003046 20$: MOV TMPO,R4
2945 012032 016400 006252 MOV PARPAT(R4),R0 ; INITIALIZE THE PATTERN
2946 ; LO BYTE =377 HI BYTE=0 (TO 377)
2947
2948 012036 012737 012052 001110 MOV #15,$LPERR ; LOOP ON ERROR POINT
2949 012044 012737 177400 003054 MOV #-400,COUNT0 ; INITIALIZE COUNT FOR NUMBER OF PATTERNS
2950
2951 012052 112737 177770 003064 1$: MOVB #-10,ERCNT0 ; DO NOT TYPE MORE THAN 10 ERRORS
2952
2953 ; WRITE WRONG PARITY INTO WCS LOCATION AND CHECK IF IT WAS
2954 ; STORED CORRECTLY. (P0, P1, P2)
2955
2956 012060 012777 000060 170746 2$: MOV #PARDIS+WWP,@WCSST
2957 012066 010177 170744 MOV R1,@WCSAR ; ADDRESS THE WCS
2958 012072 010004 MOV R0,R4
2959 012074 032701 002000 BIT #BIT10,R1 ; TESTING PAIRTY CHECKER FOR LOC 1?
2960 012100 001402 BEQ 25$ ; NO
2961 012102 004737 037262 JSR PC,XFRCHK ; REMAP THE PATTERN
2962 012106 010477 170726 25$: MOV R4,@WCSDR ; WRITE THE PATTERN
2963 012112 017705 170716 MOV @WCSST,R5 ; READ STATUS
2964 012116 013703 003050 MOV TMP1,R3 ; ISOLATE PARITY BIT STORED
2965 012122 005103 COM R3
2966 012124 040305 BIC R3,R5
2967 012126 004737 036666 JSR PC,GENWPAR ; GENERATE PARITY BIT (ODD) FOR
2968 ; THE PATTERN IN R4
2969 012132 005702 TST R2 ; AND SAVE IT IN R2
2970 012134 001402 BEQ 29$
2971 012136 013702 003050 MOV TMP1,R2
2972 012142 020502 29$: CMP R5,R2 ; WAS PARITY BIT STORED CORRECTLY?
2973 012144 001415 BEQ 21$ ; YES
2974
2975 012146 105737 003064 TSTB ERCNT0 ; DON'T REPORT MORE THAN 5 ERRORS
2976 012152 001464 BEQ 31$ ; ABORT REST OF SUB-TEST
2977 012154 105237 003064 INCB ERCNT0
2978 012160 010137 001162 MOV R1,$REG0 ; GET WCS LOCATION
2979 012164 010437 001164 MOV R4,$REG1 ; SAVE PATTERN THAT WAS WRITTEN
2980 012170 017737 170640 001166 MOV @WCSST,$REG2 ; SAVE STATUS

```



```

2981 012176 104021          ERROR 21          ; INCORRECT PARITY BIT WAS STORED
2982                          ; FOR THE PATTERN WRITTEN IN WCS
2983                          ; LOCATION. MESSAGE GIVES *
2984                          ; LOCATION AND PATTERN.
2985                          ; WRONG PARITY (WWP) I'          'N.
2986
2987                          ; CHECK THE PARITY CHECKER BY READING WCS LOCATION WI
2988                          ; BAD PARITY. PARITY ERROR TRAP SHOULD OCCUR, IF . PARITY
2989                          ; CHECKER DETECTED THE BAD PARITY.
2990
2991 012200 012737 012334 000114 21$: MOV #32$, @#114 ; PARITY TRAP VECTOR
2992 012206 042777 000020 170620 BIC #PARDIS, @WCSST ; ALLOW PARITY TRAP TO TAKE PLACE
2993 012214 010177 170616 MOV R1, @WCSAR ; READ BAD PARITY FROM THIS LOC
2994 012220 017705 170614 MOV @WCSDR, R5
2995 012224 000240 NOP ; TRAP SHOULD OCCUR
2996
2997                          ; REPORT ERROR IF BAD PARITY WAS NOT
2998                          ; DETECTED
2999 012226 012737 003344 000114 MOV #BADPAR, @#114 ; UNEXPECTED PARITY ERROR TRAP
3000 012234 105737 003064 TSTB ERCNTO ; REPORT ONLY MAXM NUMBER OF ERRORS
3001 012240 001431 BEQ 31$ ; ABORT REST OF SUB-TEST
3002 012242 105237 003064 INCB ERCNTO
3003 012246 010137 001162 MOV R1, $REG0 ; SAVE WCS LOCATION
3004 012252 010437 001164 MOV R4, $REG1 ; SAVE PATTERN
3005 012256 017737 170552 001166 MOV @WCSST, $REG2 ; SAVE STATUS
3006 012264 104025 ERROR 25 ; PARITY ERROR TRAP DID NOT OCCUR
3007                          ; ON READING BAD PARITY FROM
3008                          ; WCS LOC. FAULT LIKELY IN THE
3009                          ; PARITY CHECKER CHIP FOR THE HI
3010                          ; BYTE. ERROR MESSAGE GIVES THE
3011                          ; SEGMENT (LOCATION) TO WHICH
3012                          ; THIS CHIP BELONGS. THERE ARE THREE
3013                          ; SEGMENTS OF THE WCS MICRO-WORD
3014                          ; WPO, WP1, WP2 (PARO, PAR1, PAR2)
3015
3016                          ; IF THE BAD PARITY WAS NOT DETECTED (BY THE PARITY CHECKER FOR THE
3017                          ; HI BYTE), CHECK IF THE DATA WAS READ CORRECTLY.
3018
3019 012266 020504 CMP R5, R4 ; WAS THE CORRECT DATA READ?
3020 012270 001422 BEQ 23$ ; YES?
3021 012272 105737 003064 TSTB ERCNTO
3022 012276 001412 BEQ 31$ ; ABORT REST OF SUB-TEST
3023 012300 105237 003064 INCB ERCNTO
3024 012304 010137 001162 MOV R1, $REG0 ; SAVE WCS LOCATION
3025 012310 010437 001164 MOV R4, $REG1 ; SAVE EXPECTED DATA
3026 012314 010537 001166 MOV R5, $REG2 ; SAVE DATA READ
3027 012320 104012 ERROR 12 ; DATA READ FROM WCS LOCATION WAS
3028                          ; INCORRECT. ALSO, PARITY ERROR TRAP
3029                          ; DID NOT OCCUR ON READING
3030                          ; BAD PARITY FROM THAT LOC
3031
3032 012322 000405          22$: BR 23$
3033                          ; IF THERE WERE MORE THAN 10 ERRORS AND THE SUBTEST IS TO BE ABORTED
3034                          ; ENTER HERE.
3035 012324 012737 177777 003054 31$: MOV #-1, COUNTO ; INDICATE THAT THE REST OF SUBTEST
3036 012332 000401 BR 23$ ; IS TO BE ABORTED, PROCEED TO

```

```

3037                                     ; CLEAN-UP
3038
3039                                     ; IF THE BAD PARITY WAS DETECTED (AS EXPECTED) AND A TRAP OCCURRED
3040                                     ; ENTER HERE.  REWRITE GOOD PARITY INTO THE LOCATION
3041
3042 012334 022626          32$:  CMP      (SP)+, (SP)+      ; POP THE STACK
3043 012336 012737 003344 000114 23$:  MOV      #BADPAR, @#114  ; UNEXPECTED PARITY TRAP SERVICE
3044 012344 012777 000020 170462 30$:  MOV      #PARDIS, @WCSST ; REWRITE CORRECT PARITY, WHERE
3045 012352 010177 170460          MOV      R1, @WCSAR      ; BAD PARITY WAS WRITTEN
3046 012356 010477 170456          MOV      R4, @WCSDR
3047
3048                                     ; SETUP TO CHECK PARITY-CHECKER USING NEXT PATTERN (IN HI BYTE)
3049 012362 062700 000400          35$:  ADD      #400, R0        ; CREATE NEXT PATTERN, INCREMENT HI-BYTE
3050 012366 005237 003054          INC      COUNT0        ; IF NOT DONE, GO CHECK NEXT PATTERN
3051 012372 001232          BNE      25
3052
3053 012374 012705 000002          27$:  MOV      #2, R5
3054 012400 074537 003046          XOR      R5, TMP0      ; DONE CHECKING OF HI-BYTE PARITY
3055                                     ; CHECKER USING ODD PATTERN IN
3056                                     ; LO-BYTE?
3057 012404 001402          BEQ      28$
3058 012406 000137 012026          JMP      20$          ; IF NOT, CHECK IT USING (376) FOR LO BYTES
3059
3060                                     ; AT THIS POINT, THE HI-BYTE PARITY CHECKER FOR A PARTICULAR
3061                                     ; SEGMENT (AS INDICATED IN R1) HAS BEEN CHECKED.  SET UP
3062                                     ; TO CHECK THE NEXT SEGMENT OR EXIT IF ALL 3 ARE
3063                                     ; CHECKED.  THERE ARE THREE PARITY CHECKERS CORRESPONDING TO
3064                                     ; THE THREE SEGMENTS OF THE 48-BIT MICRO-WORD.
3065                                     ; SEGMENT 0 CORRESPONDS TO PAR0 AND WPO
3066                                     ; SEGMENT 1 CORRESPONDS TO PAR1 AND WP1
3067                                     ; SEGMENT 2 CORRESPONDS TO PAR2 AND WP2
3068
3069 012412 006337 003050          28$:  ASL      TMP1
3070 012416 062701 002000          ADD      #2000, R1     ; CREATE WCS LOCATION FOR NEXT SEGMENT
3071 012422 020127 006000          CMP      R1, #6000    ; DONE CHECKING 3 SEGMENTS?
3072 012426 001402          BEQ      TST22        ; YES.  EXIT
3073 012430 000137 012026          JMP      20$
3074
3075
3076
3077

```

```

*****
*TEST 22      CHECK THE THREE LO-BYTE PARITY CHECKERS
; THIS TEST CHECKS THE THREE LO BYTE PARITY CHECKERS, THAT THEY
; CAN DETECT BAD PARITY WHEN A WCS LOCATION IS READ.
; THE TEST CHECKS THE THREE LO-BYTE PARITY CHECKERS CORRESPONDING
; TO THE THREE SEGMENTS OF A 48-BIT MICRO-WORD.
; SEGMENT 0 CORRESPONDS TO 'PO' AND 'WPO' THE 16-BIT FIELD
; CORRESPONDING TO WCS LOCATIONS 0, 1, 2, 3, . . . 1777(OCTAL).
; SEGMENT 1 CORRESPONDS TO 'P1' AND 'WP1', THE 16 BIT FIELD
; CORRESPONDING TO WCS LOCATIONS 2000, 2001, 2002, . . . 3777(OCTAL).
; SEGMENT 2 CORRESPONDS TO 'P2' AND 'WP2', THE 16 BIT FIELD
; CORRESPONDING TO WCS LOCATIONS 4000, 4001, . . . 5777(OCTAL).
; LOCATIONS 0, 2000, 4000 ARE USED AS TEST LOCATIONS WHERE PATTERNS ARE STORED
; AND READ FROM, AND THE PARITY CHECKED.  BAD PARITY IS GENERATED AND
; STORED USING 'WRITE WRONG PARITY'.  WHEN THE BAD PARITY IS

```

```

3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092

```



```

3093 ; READ THE PARITY CHECKER SHOULD DETECT IT AND CAUSE A TRAP
3094 ; A COUNT PATTERN FROM 0 TO 377 IS USED TO TEST THE
3095 ; 8-BIT PARITY CHECKER (LO BYTE). MEANWHILE, THE HI-BYTE PATTERN.
3096 ; IS KEPT CONSTANT AT EVEN (377) AND ODD (376). IF THE
3097 ; BAD PARITY IS NOT DETECTED FOR A PARTICULAR PATTERN AN ERROR
3098 ; MESSAGE IS REPORTED.
3099 ; THE ENTIRE TEST IS REPEATED FOR EACH OF THE THREE
3100 ; LO BYTE PARITY CHECKERS USING TEST LOCATIONS 0, 2000, 4000.
3101 ; *****
3102 012434 000004 TST22: SCOPE
3103 012436 012737 013056 003056 MOV #TST23,NXTST ; STARTING ADDRESS OF NEXT TEST
3104
3105 012444 005037 003046 CLR TMPO ; IF TMPO=0, USE EVEN PATTERN (377) FOR HI BYTE
3106 ; IF TMPO=2, USE ODD PATTERN (376) FOR HI BYTE
3107 ; FOR BOTH CASES, VARY LO-BYTE FROM 0 TO 377
3108
3109 012450 005001 CLR R1 ; INITIALIZE WCS LOCATION, CORRESPONDING
3110 ; TO WHICH THE PARITY CHECKS WILL BE
3111 ; CHECKED. NOTE, THERE ARE THREE SETS OF
3112 ; PARITY CHECKERS:
3113 ; FOR SEGMENT 0 (LOCATIONS 0, 1, 2, ...)
3114 ; FOR SEGMENT 1 (LOCATIONS 2000, 2001, ...)
3115 ; FOR SEGMENT 2 (LOCATIONS 4000, 4001, ...)
3116
3117 012452 012702 000002 MOV #PAR0,R2
3118 012456 010237 003050 MOV R2,TMP1
3119 012462 013704 003046 20%: MOV TMPO,R4
3120 012466 016400 006252 MOV PARPAT(R4),R0 ; INITIALIZE THE PATTERN
3121 012472 000300 SWAB R0 ; HI BYTE =377 LO BYTE =0 (TO 377)
3122
3123 012474 012737 012502 001110 MOV #15,$LPERR ; LOOP ON ERROR POINT
3124
3125 012502 112737 177770 003064 1%: MOVB #-10,ERCNT0 ; DO NOT TYPE MORE THAN 10 ERRORS
3126
3127 ; WRITE WRONG PARITY INTO WCS LOCATION AND CHECK IF IT WAS
3128 ; STORED CORRECTLY. (P0, P1, P2)
3129
3130 012510 012777 000060 170316 2%: MOV #PARDIS+WWP,@WCSST
3131 012516 010177 170314 MOV R1,@WCSAR ; ADDRESS THE WCS
3132 012522 010004 MOV R0,R4
3133 012524 032701 002000 BIT #BIT10,R1 ; TESTING PARITY CHIP FOR LOC 1?
3134 012530 001402 BEQ 25% ; NO
3135 012532 004737 037262 JSR PC,XFRCHK ; REMAP THE PATTERN
3136 012536 010477 170276 25%: MOV R4,@WCSDR ; WRITE THE PATTERN
3137 012542 017705 170266 MOV @WCSST,R5 ; READ STATUS
3138 012546 013703 003050 MOV TMP1,R3 ; ISOLATE PARITY BIT STORED
3139 012552 005103 COM R3
3140 012554 040305 BIC R3,R5
3141 012556 004737 036666 JSR PC,GENWPAR ; GENERATE PARITY BIT (EVEN) FOR
3142 ; THE PATTERN IN R4
3143 ; AND SAVE IT IN R2
3143 012562 005702 TST R2
3144 012564 001402 BEQ 29%
3145 012566 013702 003050 MOV TMP1,R2
3146 012572 020502 29%: CMP R5,R2 ; WAS PARITY BIT STORED CORRECTLY?
3147 012574 001415 BEQ 21% ; YES
3148

```





```
3205 ;BAD PARITY FROM THAT LOC
3206
3207 012746 000404 205: BR 23$
3208 ;IF THERE WERE MORE THAN 10 ERRORS AND THE SUBTEST IS TO BE
3209 ;ABORTED ENTER HERE
3210 012750 112700 177777 315: MOVB #-1,R0 ;INDICATE THAT THE REST OF SUBTEST
3211 012754 000401 BR 23$ ;IS TO BE ABORTED & PROCEED TO CLEANUP
3212
3213 ;IF THE BAD PARITY WAS DETECTED (AS EXPECTED) AND A TRAP OCCURRED
3214 ;ENTER HERE. REWRITE GOOD PARITY INTO THE LOCATIN
3215
3216 012756 022626 325: CMP (SP)+,(SP)+ ;POP THE STACK
3217 012760 012737 003344 000114 235: MOV #BADPAR,@#114 ;UNEXPECTED PARITY TRAP SERVICE
3218 012766 012777 000020 170040 305: MOV #PARDIS,@WCSST ;REWRITE CORRECT PAITY, WHERE
3219 012774 010177 170036 MOV R1,@WCSAR ;BAD PARITY WAS WRITTEN
3220 013000 010477 170034 MOV R4,@WCSDR
3221
3222 ;SETUP TO CHECK PARITY-CHECKER USING NEXT PATTERN (IN LO BYTE)
3223 013004 105200 35: INCB R0 ;IF NOT DONE, GO CHECK NEXT PATTERN
3224 013006 001240 BNE 25$ ;DONE ALL PATTERNS IN LO BYTE?
3225
3226 013010 000400 BR 27$
3227 013012 012705 000002 275: MOV #2,R5
3228 013016 074537 003046 XOR R5,TMPO ;DONE CHECKING OF HI-BYTE PARITY
3229 ;CHECKER USING ODD PATTERN IN
3230 ;LO-BYTE?
3231 013022 001402 BEQ 28$
3232 013024 000137 012462 JMP 20$ ;IF NOT, CHECK IT USING (376) FOR LO BYTES
3233
3234 ;AT THIS POINT, THE LO-BYTE PARITY CHECKER FOR A PARTICULAR
3235 ;SEGMENT (AS INDICATED IN R1) HAS BEEN CHECKED. SET UP
3236 ;TO CHECK THE NEXT SEGMENT OR EXIT IF ALL 3 ARE
3237 ;CHECKED. THERE ARE THREE PARITY CHECKERS CORRESPONDING TO
3238 ;THE THREE SEGMENTS OF THE 48-BIT MICRO-WORD.
3239 ;SEGMENT 0 CORRESPONDS TO PAR0 AND WPO
3240 ;SEGMENT 1 CORRESPONDS TO PAR1 AND WP1
3241 ;SEGMENT 2 CORRESPONDS TO PAR2 AND WP2
3242
3243 013030 006337 003050 285: ASL TMP1 ;CREATE THE NEXT 'PARITY BIT STORED
3244 013034 013702 003050 MOV TMP1,R2
3245 013040 062701 002000 ADD #2000,R1 ;CREATE WCS LOCATION FOR NEXT SEGMENT
3246 013044 020127 006000 CMP R1,#6000 ;DONE CHECKING 3 SEGMENTS?
3247 013050 001402 BEQ TST23 ;;YES, EXIT
3248 013052 000137 012462 JMP 20$ ;IF NOT GO BACK
3249
3250
3251 ;*****
3252 ;*TEST 23 CHECK ADDRESSING FOR DATA ARRAY <11:0>, ADDRESS<9:0>
3253 ;THIS TEST CHECKS OUT THE ADDRESSING SCHEME FOR THE
3254 ;DATA ARRAY <11:00>. THIS GROUP OF 12 CHIPS IS
3255 ;ADDRESSED BY THE SAME SET OF ADDRESS SELECTORS. THIS TEST
3256 ;CHECKS THE COMMON ADDRESS LINES <11:0>. THE TEST
3257 ;CONTAINS A FAULT ANALYZER WHICH ATTEMPTS TO FIGURE OUT
3258 ;THE ADDRESS LINES WHICH ARE FAULTY. THERE ARE 10 ADDRESS
3259 ;LINES (00-09).
3260 ;AN INCREMENTING COUNT PATTERN IS WRITTEN INTO ALL THE
```

```

3261 ;LOCATIONS (1024) OF SEGMENT 0 (WPO) OF THE WCS
3262 ;(ADDRESSES 0,1,2,3,...1777. (OCTAL). THIS IS EQUIVALENT TO WRITING
3263 ;THE ADDRESS AS DATA FOR THE VERTICAL ARRAY. THEN A
3264 ;READ IS DONE.
3265 ;LOGICAL 'AND' AND 'OR' FUNCTION ARE PERFORMED ON ALL
3266 ;THE PATTERNS READ.
3267 ;A SEPERATE LOGICAL 'AND' AND 'OR' IS PERFORMED FOR
3268 ;ONLY THOSE PATTERNS THAT FAIL.
3269 ;THE ABOVE INFORMATION IS USED BY THE FAULT ANALYZER
3270 ;ROUTINE. MORE INFORMATION CAN BE FOUND AT
3271 ;THE BEGINNING OF THE ROUTINE.
3272 ;IT SHOULD BE REALIZED THAT THE FAULT ANALYZER
3273 ;WORKS ON A "SINGLE SOLID FAULT" ASSUMPTION AND
3274 ;HENCE RESULTS MAY NOT BE PREDICTABLE IN CASE OF MULTIPLE,
3275 ;SOFT ERRORS, OR DATA ERRORS.
3276 ;*****
3277 013056 000004 TST23: SCOPE
3278 013060 012737 013464 003056 MOV #TST24,NXTST ;STARTING ADDRESS OF NEXT TEST
3279
3280 ;WRITE ALL THE WORDS IN SEGMENT 0 OF WCS (CORRESPONDING
3281 ;TO WPO AND PARO. ADDRESS (IN SEGMENT 0) IS WRITTEN AS DATA.
3282
3283
3284 013066 004737 036624 JSR PC,CLRWCSS
3285 013072 005077 167736 CLR @WCSST
3286 013076 005000 CLR R0 ;INITIALIZE WCS LOCATION
3287 013100 005001 CLR R1 ;INITIALIZE WCS DATA
3288 013102 010077 167730 15: MOV R0,@WCSAR ;WRITE DATA INTO THIS WCS
3289 013106 010177 167726 MOV R1,@WCSDR ;LOCATION
3290 013112 005200 INC R0 ;NEXT WCS LOCATION
3291 013114 005201 INC R1 ;NEXT WCS DATA
3292 013116 020127 002000 CMP R1,#1024. ;DONE WRITING ENTIRE 1024 WORD
3293 013122 001367 BNE 15 ;SEGMENT?
3294
3295 ;READ BACK ALL THE WORDS (1024) OF SEGMENT 0 OF
3296 ;WCS. CHECK IF THEY WERE READ CORRECTLY.
3297 013124 012777 000020 167702 MOV #PARDIS,@WCSST
3298 013132 005000 CLR R0 ;INITIALIZE WCS LOCATION
3299 013134 005001 CLR R1 ;INITIALIZE WCS DATA
3300 013136 005004 CLR R4 ;R4 WILL CONTAIN A RUNNING
3301 ;'OR' OF ALL THE WORDS READ
3302 013140 012705 177777 MOV #-1,R5 ;R5 WILL CONTAIN A RUNNING
3303 ;'AND' OF ALL THE WORDS READ
3304 013144 005037 003046 CLR TMP0 ;'TMP0' WILL CONTAIN 'OR' OF THE
3305 ;DATA PATTERNS THAT FAILED
3306 013150 010537 003050 MOV R5,TMP1 ;'TMP1' WILL CONTAIN 'AND' OF
3307 ;THE DATA PATTERNS THAT FAILED
3308 013154 005003 CLR R3 ;KEEP COUNT OF NUMBER OF DATA ERRORS
3309 013156 112737 177773 003065 MOVB #-5,ERCNT1
3310 013164 112737 177773 003064 MOVB #-5,ERCNT0 ;DON'T REPORT MORE THAN 5 ERRORS
3311 013172 012737 013206 001110 MOV #25,$LPERR ;LOOP ON ERROR
3312 013200 012737 013344 000114 MOV #85,@#114 ;SET PARITY TRAP VECTOR
3313 013206 010077 167624 25: MOV R0,@WCSAR ;READ FROM THIS WCS LOCATION
3314 013212 017702 167622 MOV @WCSDR,R2 ;AND SAVE IN R2
3315 013216 050204 BIS R2,R4 ;KEEP RUNNING 'OR' OF DATA
3316 013220 005102 COM R2 ;PATTERNS READ, IN R4

```



```

3317 013222 005105      COM      R5
3318 013224 050205      BIS      R2,R5      ;KEEP RUNNING 'AND' OF DATA
3319 013226 005105      COM      R5      ;PATTERNS READ, IN R5
3320 013230 005102      COM      R2
3321
3322 013232 020102      CMP      R1,R2      ;DATA PATTERN READ OK?
3323 013234 001006      BNE      45      ;NO, DATA MISCOMPARE OCCURED
3324
3325 013236 005200      35:     INC      R0      ;SET UP TO READ NEXT ADDRESS
3326 013240 005201      INC      R1      ;NEXT DATA PATTERN
3327 013242 020127 002000  CMP      R1,#1024.  ;DONE READING ALL?
3328 013246 001357      BNE      25      ;NO
3329
3330 013250 000466      BR       65      ;YES, GO ANALYZE THE FAULT
3331                          ;IF ANY ERROR OCCURED
3332                          ;DATA MISCOMPARE OCCURED ON READING THE DATA
3333                          ;FROM THE WCS LOCATION.
3334                          ;R0 CONTAINS THE WCS LOCATION READ
3335                          ;R1 CONTAINS EXPECTED DATA
3336                          ;R2 CONTAINS DATA THAT WAS READ
3337
3338 013252 105237 003064      45:     INCB     ERCNTO      ;DON'T REPORT MORE THAN 5 ERRORS
3339 013256 001407      BEQ      55      ;SKIP, IF 5 REPORTED
3340
3341 013260 010037 001162      MOV      R0,$REG0      ;SAVE WCS LOCATION GIVING DATA ERROR
3342 013264 010137 001164      MOV      R1,$REG1      ;SAVE EXPECTED DATA
3343 013270 010237 001166      MOV      R2,$REG2      ;SAVE DATA RECEIVED
3344 013274 104012      ERROR    12      ;DATA MISCOMPARE OCCURRED ON READING
3345                          ;WCS LOCATION.
3346                          ;POSSIBLE ADDRESS ERROR FOR
3347                          ;DATA ARRAY <11: 0> THIS TEST
3348                          ;CONTAINS A FAULT ANALYZER
3349                          ;WHICH WILL ATTEMPT TO
3350                          ;ANALYZE THE ABOVE ERROR.
3351
3352 013276 010146      55:     MOV      R1,-(SP)
3353 013300 042716 176000      BIC      #176000,(SF) ;MAKE SURE THE DATA MISCOMPARE
3354 013304 010246      MOV      R2,-(SP)      ;WAS IN BITS <0-11>. IF OUTSIDE,
3355 013306 042716 176000      BIC      #176000,(SF) ;(BITS <15-12>), DON'T DO
3356 013312 022626      CMP      (SP)+,(SP)+  ;FAULT ANALYSIS (AND,OR)
3357 013314 001750      BEQ      35      ;SKIP FAULT ANALYSIS
3358 013316 005203      INC      R3      ;KEEP COUNT OF ERRORS IN DATA BITS <0-11>
3359 013320 050237 003046      BIS      R2,TMP0      ;'OR' THE FAILING DATA PATTERN INTO
3360                          ;'TMP0'
3361 013324 005102      COM      R2      ;'AND' THE FAILING PATTERN WITH
3362 013326 005137 003050      COM      TMP1      ;'TMP1'
3363 013332 050237 003050      BIS      R2,TMP1
3364 013336 005137 003050      COM      TMP1
3365
3366 013342 000735      BR       35      ;RETURN TO DO FURTHER CHECKING
3367
3368                          ;IF A PARITY ERROR OCCURED ON READING A WCS LOCATION,
3369                          ;ENTER HERE.
3370 013344 012737 003344 000114 55:     MOV      #BADPAR,@#114 ;NORMAL PARITY ERROR HANDLED
3371
3372 013352 032777 100000 167454  BIT      #PARERR,@WCSST ;WAS IT A WCS PARITY ERROR?

```

```

3373 013360 001420      BEQ      10$
3374 013362 022626      CMP      (SP)+,(SP)+ ;RESTORE THE STACK POINTER
3375 013364 017737 167444 001166      MOV      @WCSST,$REG2
3376 013372 012777 000020 167434      MOV      #PARDIS,@WCSST ;CLEAR OUT THE PARITY ERROR
3377 013400 105237 003065      INCB    ERCNT1 ;DON'T REPORT MORE THAN 5 ERRORS
3378 013404 001405      BEQ      9$
3379 013406 010037 001162      MOV      R0,$REG0 ;SAVE ADDRESS GIVING PARITY ERROR
3380 013412 010137 001164      MOV      R1,$REG1 ;SAVE EXPECTED DATA
3381 013416 104026      ERROR   26 ;PARITY ERROR OCCURED ON READING
3382 ;A WCS LOCATION. NOTE THAT
3383 ;PARITY ERROR CAN BE FROM ANY
3384 ;ONE OF THE THREE 16-BIT WORDS
3385 ;OF THE DATA D<47:0>
3386
3387 013420 000706      9$: BR      3$ ;DO THE REST OF CHECKING
3388
3389 013422 000137 003344      10$: JMP     BADPAR ;IF THE PARITY ERROR WAS NOT FROM
3390 ;WCS,GO TO THE COMMON PARITY
3391 ;ERROR HANDLER
3392
3393 ;GO TO THE FAULT ANALYZER AND PRINT OUT WHICH ADDRESS
3394 ;LINES ARE FAULTY.
3395
3396 013426 012737 003344 000114 6$: MOV      #BADPAR,@#114 ;NORMAL PARITY ERROR HANDLER
3397 013434 005703      TST      R3 ;ANY ERROR (D<11-0>) OCCURRED?
3398 013436 001412      BEQ      TST24 ;EXIT
3399 013440 013700 003046      MOV      TMP0,R0 ;GET 'OR'
3400 013444 013701 003050      MOV      TMP1,R1 ;GET 'AND'
3401 013450 012746 042725      MOV      #MSG22,-(SP) ;MESSAGE POINTER
3402 013454 012746 176000      MOV      #176000,-(SP) ;BIT MASK
3403 013460 004737 037350      JSR      PC,ANLYSO ;GO TO THE FAULT ANALYZER
3404

```

```

3405 ;*****
3406 ;*TEST 24 CHECK ADDRESSING FOR DATA ARRAY (23:12),ADDRESS(7:0)
3407 ;THIS TEST CHECKS OUT THE ADDRESSING SCHEME FOR THE DATA
3408 ;ARRAY <23:12>. THIS GROUP OF 12 CHIPS IS ADDRESSED BY
3409 ;THE SAME SET OF ADDRESS SELECTORS. (THERE ARE 10 ADDRESS
3410 ;LINES <00-09>). THIS TEST CHECKS THE COMMON ADDRESS LINES
3411 ;<00-07>. THE FAULT ANALYZER ATTEMPTS TO FIGURE OUT WHICH ONE
3412 ;OF THESE ADDRESS LINES (02-09) WAS FAULTY.
3413 ;AN INCREMENTING COUNT PATTERN IS WRITTEN INTO (FIRST 256)
3414 ;LOCATIONS OF THE SEGMENT 1 OF WCS (WP1) (ADDRESSES
3415 ;2000,2001,...). THIS IS EQUIVALENT TO WRITING
3416 ;THE ADDRESS AS DATA INTO THE FIRST 256 LOCATIONS OF
3417 ;THE VERTICAL ARRAY (D<31:16>). THEN A READ IS DONE.
3418
3419 ;LOGICAL 'AND' AND 'OR' FUNCTIONS ARE PERFORMED ON ALL
3420 ;THE PATTERNS READ.
3421 ;A SEPERATE LOGICAL 'AND' AND 'OR' IS PERFORMED FOR
3422 ;ONLY THOSE PATTERNS THAT FAIL.
3423 ;THE ABOVE INFORMATION IS USED BY THE FAULT ANALYZER
3424 ;ROUTINE (MORE INFO ON IT CAN BE FOUND AT THE
3425 ;BEGINNING OF THE ROUTINE).
3426 ;*****
3427 013464 000004      TST24: SCOPE
3428 013466 012737 014074 003056      MOV      #TST25,NXTST ;STARTING ADDRESS OF NEXT TEST

```





```

3485
3486 013662 105237 003064      45:  INCB  ERCNTO  ; DONT REPORT MORE THAN 5 ERRORS
3487 013666 001407
3488
3489 013670 010037 001162      MOV   R0, $REG0  ; SAVE WCS LOCATION GIVING DATA ERROR
3490 013674 010137 001164      MOV   R1, $REG1  ; SAVE EXPECTED DATA
3491 013700 010237 001166      MOV   R2, $REG2  ; SAVE DATA RECEIVED
3492 013704 104012      ERROR  12        ; DATA MISCOMPARE OCCURRED ON READING
3493                                     ; WCS LOCATION. THIS TEST CONTAINS
3494                                     ; A FAULT ANALYZER WHICH
3495                                     ; ATTEMPTS TO ANALYZE THE
3496                                     ; FAULT.
3497                                     ; POSSIBLE ADDRESS ERROR FOR DATA
3498                                     ; ARRAY <23:12>.
3499
3500 013706 010146      55:  MOV   R1, -(SP)  ; MAKE SURE THE DATA MISCOMPARE IS
3501 013710 042716 177400      BIC   #177400, (SP) ; IN BITS D<23:16> OF THE
3502 013714 010246      MOV   R2, -(SP)  ; WCS DATA ARRAY. IF OUTSIDE
3503 013716 042716 177400      BIC   #177400, (SP) ; DON'T DO FAULT
3504 013722 022626      CMP   (SP)+, (SP)+ ; ANALYSIS
3505 013724 001750      BEQ   35         ; SKIP FAULT ANALYSIS
3506 013726 005203      INC   R3         ; KEEP COUNT OF ERRORS IN
3507                                     ; DATA BITS D<23:16>
3508 013730 050237 003046      BIS   R2, TMPO   ; 'OR' THE FAILING DATA PATTERN
3509 013734 005102      COM   R2         ; INTO TMPO
3510 013736 005137 003050      COM   TMP1       ; 'AND' THE FAILING PATTERN WITH
3511 013742 050237 003050      BIS   R2, TMP1   ; 'TMP1'
3512 013746 005137 003050      COM   TMP1
3513 013752 000735      BR    35         ; RETURN TO DO FURTHER CHECKING.
3514
3515                                     ; IF A PARITY ERROR OCCURED ON READING A WCS LOCATION,
3516                                     ; ENTER HERE.
3517
3518 013754 012737 003344 000114 85:  MOV   #BADPAR, @#114 ; NORMAL BAD PARITY HANDLER
3519 013762 032777 100000 167044      BIT   #PARERR, @WCSST ; WAS IT A WCS PARITY ERROR?
3520 013770 001420      BEQ   105        ;
3521 013772 022626      CMP   (SP)+, (SP)+ ; RESTORE THE STACK POINTER
3522 013774 017737 167034 001166      MOV   @WCSST, $REG2
3523 014002 012777 000020 167024      MOV   #PARDIS, @WCSST ; CLEAR OUT THE PARITY ERROR
3524 014010 105237 003065      INCB  ERCNT1     ; DON'T REPORT MORE THAN 5 ERRORS
3525 014014 001405      BEQ   95         ;
3526 014016 010037 001162      MOV   R0, $REG0  ; SAVE ADDRESS GIVING PARITY ERROR
3527 014022 010137 001164      MOV   R1, $REG1  ; SAVE EXPECTED DATA
3528 014026 104026      ERROR  26        ; PARITY ERROR OCCURED ON READING
3529                                     ; A WCS LOCATION. NOTE THAT
3530                                     ; PARITY ERROR CAN BE FROM ANY
3531                                     ; ONE OF THE THREE 16-BIT WORDS
3532                                     ; OF THE DATA D<47:0>
3533
3534 014030 000706      95:  BR    35         ; DO THE REST OF CHECKING
3535
3536 014032 000137 003344      105: JMP   BADPAR     ; IF PARITY ERROR IS NOT FROM THE
3537                                     ; WCS GO TO THE COMMON PARITY
3538                                     ; ERROR HANDLER.
3539
3540                                     ; GO TO THE FAULT ANALYZER AND PRINT OUT WHICH ADDRESS LINES

```



```

3541 ;ARE FAULTY
3542
3543 014036 012737 003344 000114 65: MOV #BADPAR, @#114 ;PARITY ERROR HANDLER
3544 014044 005703 TST R3 ;ANY ERROR (D<23-12>) OCURRED?
3545 014046 001412 BEQ TST25 ;;EXIT
3546 014050 013700 003046 MOV TMP0, R0 ;GET 'OR'
3547 014054 013701 003050 MOV TMP1, R1 ;GET 'AND'
3548 014060 012746 042774 MOV #MSG25, -(SP) ;MESSAGE POINTER
3549 014064 012746 176000 MOV #176000, -(SP) ;BIT MASK
3550 014070 004737 037350 JSR PC, ANLYSO ;GO TO THE FAULT ANALYZER

```

3551  
3552  
3553

```

;*****
;*TEST 25 CHECK ADDRESSING FOR DATA ARRAY (23:12), ADDRESS (9:8)
;THIS TEST CHECKS OUT THE ADDRESSING SCHEME FOR DATA ARRAY <23:12>.
;THIS GROUP OF 12 CHIPS IS ADDRESSED BY THE SAME SET OF
;ADDRESS SELECTORS. THERE ARE 10 ADDRESS LINES <00:09>. THIS
;TEST CHECKS THE COMMON ADDRESS LINES <8:9> AND REPORTS ERRORS.
;THE FAULT ANALYZER ATTEMPTS TO FIGURE OUT WHICH ONE OF THE
;ADDRESS LINES WAS FAULTY.

```

3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568

```

;STARTING AT LOCATION 2077 (OCTAL, TREATING THE WCS AS A
;16 X 3K DATA ARRAY) AN INCREMENTING COUNT PATTERN IS
;WRITTEN IN 16 CONSECUTIVE LOCATIONS OF THE VERTICAL ARRAY D<31:16>.
;THIS IS EQUIVALENT TO WRITING ADDRESS AS DATA INTO A
;16 WORD DATA ARRAY CORRESPONDING TO A<6:9> AND D<19:16>.
;THEN A READ IS DONE AND DATA COMPARED.

```

3569  
3570  
3571  
3572  
3573  
3574  
3575  
3576  
3577  
3578

```

;LOGICAL 'AND' AND 'OR' FUNCTIONS ARE PERFORMED ON
;ALL THE PATTERNS READ.
;A SEPERATE LOGICAL 'AND' AND 'OR' IS PERFORMED FROM
;ONLY THOSE PATTERNS THAT FAIL.
;THE ABOVE INFORMATION IS USED BY THE FAULT ANALYZER
;ROUTINE (MORE INFO ON IT CAN BE FOUND AT THE BEGINING
;OF THE ROUTINE 'ANLYZO'). THE FAULT ANALYZER ALSO TYPES OUT
;TWO OCTAL NUMBERS (AND & OR). THESE TWO CAN BE
;INTERPRETED AS EXPLAINED AT THE BEGINING OF THE ANALYZER.

```

3579

```

;*****
TST25: SCOPE

```

```

3580 014074 000004
3581 014076 012737 014520 003056
3582 014104 004737 036624
3583 014110 005077 166720
3584 014114 012700 002077
3585
3586 014120 005001
3587 014122 010077 166710
3588 014126 010177 166706
3589 014132 062700 000100
3590 014136 005201
3591 014140 020127 000020
3592 014144 001366

```

```

MOV #TST26, NXTST ;STARTING ADDRESS OF NEXT TEST
JSR PC, CLRWCS ;CLEAR OUT WCS
CLR @WCSST
MOV #2077, R0 ;INITIALIZE WCS LOCATION (253)
;77, 01 (CONCANTANATE)
CLR R1 ;INITIALIZE WCS DATA
MOV R0, @WCSAR ;WRITE DATA INTO THIS WCS
MOV R1, @WCSOR ;LOCATION
ADD #100, R0 ;NEXT WCS LOCATION
INC R1 ;NEXT DATA PATTERN
CMP R1, #16. ;DONE?
BNE 15

```

3593  
3594  
3595  
3596

```

;READ BACK ALL 16 WORDS THAT WEPE WRITTEN AND CHECK IF
;THEY ARE READ CORRECTLY.

```

```

3597 014146 005077 166662 CLR @WCSST
3598 014152 012700 002077 MOV #2077,R0 ; INITIALIZE WCS LOCATION
3599 014156 005001 CLR R1 ; INITIALIZE WCS DATA
3600 014160 005004 CLR R4 ; R4 WILL CONTAIN A RUNNING 'OR'
3601 ; OF THE WORDS READ
3602 014162 012705 177777 MOV #-1,R5 ; R5 WILL CONTAIN A RUNNING 'AND'
3603 ; OF THE WORDS READ
3604 014166 005037 003046 CLR TMPO ; 'TMPO' WILL CONTAIN 'OR' OF THE
3605 ; DATA PATTERNS THAT FAILED
3606 014172 010537 003050 MOV R5,TMP1 ; 'TMP1' WILL CONTAIN 'AND' OF THE
3607 ; DATA PATTERNS THAT FAILED
3608 014176 112737 177773 003064 MOVB #-5,ERCNT0 ; DONT REPORT MORE THAN 5 ERRORS
3609 014204 112737 177773 003065 MOVB #-5,ERCNT1
3610 014212 012737 014226 001110 MOV #25,$LPERR ; LOOP ON ERROR
3611 014220 012737 014366 000114 MOV #85,@#114 ; SET PARITY TRAP VECTOR
3612 014226 010077 166604 25: MOV R0,@WCSAR ; READ FROM THIS WCS LOCATION AND
3613 014232 017702 166602 MOV @WCSADR,R2 ; SAVE IT IN R2
3614 014236 050204 BIS R2,R4 ; KEEPING RUNNING 'OR' OF DATA
3615 014240 005102 COM R2 ; PATTERNS READ, IN R4
3616 014242 005105 COM R5
3617 014244 050205 BIS R2,R5 ; KEEPING RUNNING 'AND' OF DATA
3618 014246 005105 COM R5 ; PATTERNS READ, IN R5
3619 014250 005102 COM R2
3620 014252 020102 CMP R1,R2 ; DATA PATTERN READ OK?
3621 014254 001007 BNE 45 ; NO
3622 014256 062700 000100 35: ADD #100,R0 ; SET UP TO READ NEXT ADDRESS
3623 014262 005201 INC R1 ; NEXT DATA PATTERN
3624 014264 020127 000020 CMP R1,#16. ; DONE?
3625 014270 001356 BNE 25 ; NO
3626 014272 000465 BR 65
3627
3628 ; DATA MISCOMPARE OCCURRED ON READING THE DATA FROM THE
3629 ; WCS LOCATION.
3630 ; R0 CONTAINS WCS LOCATION READ.
3631 ; R1 CONTAINS EXPECTED DATA
3632 ; R2 CONTAINS DATA THAT WAS READ
3633
3634 014274 105237 003064 45: INCB ERCNT0 ; DONT REPORT MORE THAN 5 ERRORS
3635 014300 001407 BEQ 55
3636 014302 010037 001162 MOV R0,$REG0 ; SAVE WCS LOCATION GIVING DATA ERROR
3637 014306 010137 001164 MOV R1,$REG1 ; SAVE EXPECTED DATA
3638 014312 010237 001166 MOV R2,$REG2 ; SAVE DATA RECEIVED
3639 014316 104012 ERROR 12 ; DATA MISCOMPARE OCCURRED ON READING
3640 ; WCS LOCATION. (POSSIBLE FAULT
3641 ; IN ADDRESS LINES FOR DATA ARRAY
3642 ; <23:12>). THIS TEST CONTAINS A
3643 ; FAULT ANALYZER WHICH ATTEMPTS TO
3644 ; ANALYZE THE FAULT.
3645 014320 010146 55: MOV R1,-(SP) ; MAKE SURE THE DATA MISCOMPARE
3646 014322 042716 177760 BIC #177760,(SP) ; IS IN BITS D<19:16> OF THE
3647 014326 010246 MOV R2,-(SP) ; WCS DATA ARRAY. IF OUTSIDE
3648 014330 042716 177760 BIC #177760,(SP) ; DONT DO FAULT ANALYSIS
3649 014334 022626 CMP (SP)+,(SP)+
3650 014336 001747 BEQ 35 ; SKIP FAULT ANALYSIS
3651 014340 005203 INC R3 ; KEEP COUNT OF ERRORS IN DATA
3652 ; BITS <19:16>

```



```

3653 014342 050237 003046      BIS      R2,TMPO      ;'OR' THE FAILING DATA PATTERN
3654                                ; INTO 'TMPO'
3655 014346 005102      COM      R2
3656 014350 005137 003050      COM      TMP1
3657 014354 050237 003050      BIS      R2,TMP1     ;'AND' THE FAILING PATTERN WITH
3658 014360 005137 003050      COM      TMP1     ;'TMP1'
3659 014364 000734      BR       3$         ;RETURN TO DO FURTHER CHECKING
3660
3661                                ; IF A PARITY ERROR OCCURRED ON READING A WCS LOCATION
3662                                ; ENTER HERE.
3663 014366 012737 003344 000114 8$:  MOV      #BADPAR,@#114 ;NORMAL BAD PARITY HANDLER
3664
3665 014374 032777 100000 166432      BIT      #PARERR,@WCSST ;WAS IT A WCS PARITY ERROR?
3666 014402 001417      BEQ      10$        ;NO
3667 014404 022626      CMP      (SP)+,(SP)+ ;RESTORE THE STACK POINTER
3668 014406 017737 166422 001166      MOV      @WCSST,$REG2
3669 014414 012777 000020 166412      MOV      #PARDIS,@WCSST ;CLEAR OUT THE PARITY ERROR
3670 014422 005077 166406      CLR      @WCSST
3671 014426 010037 001162      MOV      R0,$REG0   ;SAVE ADDRESS GIVING PARITY ERROR
3672 014432 010137 001164      MOV      R1,$REG1   ;SAVE EXPECTED DATA
3673 014436 104026      ERROR   26         ;PARITY ERROR OCCURRED ON READING A
3674                                ; WCS LOCATION. THE PARITY ERROR CAN
3675                                ; BE FROM ANY ONE OF THE THREE 16-BIT
3676                                ; WORDS OF THE DATA D<4>
3677 014440 000706      9$:  BR       3$         ;DO THE REST OF CHECKING
3678
3679 014442 000137 003344      10$:  JMP      BADPAR     ;IF PARITY ERROR IS NOT FROM THE
3680                                ; WCS GO TO THE COMMON
3681                                ; PARITY ERROR HANDLER
3682
3683                                ; GO TO THE FAULT ANALYZER AND PRINT OUT WHICH ADDRESS
3684                                ; LINES ARE FAULTY
3685
3686 014446 012737 003344 000114 6$:  MOV      #BADPAR,@#114 ;PARITY ERROR HANDLER
3687 014454 005703      TST      R3         ;ANY ERRORS IN D<19:16>?
3688 014456 001420      BEQ      TST26      ;EXIT
3689 014460 013700 003046      MOV      TMP0,R0    ;GET 'OR'
3690 014464 013701 003050      MOV      TMP1,R1    ;GET 'AND'
3691 014470 000300      SWAB    R0         ;POSITION THE 'OR AND 'AND'
3692 014472 000301      SWAB    R1         ;SO AS TO COVER THE ADDRESS
3693 014474 006200      ASR     R0         ;LINES BEING TESTED.
3694 014476 006200      ASR     R0         ;MAP BITS <3:0> TO <9:6>
3695 014500 006201      ASR     R1
3696 014502 006201      ASR     R1
3697 014504 012746 042774      MOV      #MSG25,-(SP) ;MESSAGE POINTER
3698 014510 012746 176077      MOV      #176077,-(SP) ;BIT MASK
3699 014514 004737 037350      JSR     PC,ANLYSO   ;GO TO THE FAULT ANALYZER
3700
3701                                ; *****
3702                                ; *TEST 26 CHECK ADDRESSING FOR DATA ARRAY (35:24), ADDRESS (7:0)
3703                                ; THIS TEST CHECKS OUT THE ADDRESSING SCHEME FOR THE DATA ARRAY (35:24)
3704                                ; THIS GROUP OF 12 CHIPS IS ADDRESSED BY THE SAME SET OF ADDRESS
3705                                ; SELECTORS. THERE ARE 10 ADDRESS LINES (9:0). THIS TEST CHECKS
3706                                ; COMMON ADDRESS LINES (7:0) AND REPORTS ERRORS. THE FAULT
3707                                ; ANALYZER ATTEMPTS TO FIGURE OUT WHICH ONE OF THE ADDRESS LINES
3708                                ; WAS FAULTY.

```

```

3709
3710 ; STARTING AT LOCATION 2000 (OCTAL, TREATING THE WCS AS A 16 X 3K
3711 ; DATA ARRAY) AN INCREMENTING COUNT PATTERN IS WRITTEN IN
3712 ; 256 CONSECUTIVE LOCATIONS OF THE VERTICAL ARRAY D<31:24>.
3713 ; THIS IS EQUIVALENT TO WRITING ADDRESS AS DATA INTO A 256
3714 ; WORD DATA ARRAY CORRESPONDING TO A<7:0> AND D<31:24>.
3715 ; THEN A READ IS DONE AND DATA COMPARED.
3716
3717 ; LOGICAL 'AND' AND 'OR' FUNCTIONS ARE PERFORMED ON ALL THE
3718 ; PATTERNS READ.
3719 ; A SEPERATE LOGICAL 'AND' AND 'OR' IS PERFORMED FOR ONLY THOSE PATTERNS
3720 ; THAT FAILED.
3721 ; THE ABOVE INFORMATION IS USED BY THE FAULT ANALYZER ROUTINE
3722 ; (MORE INFO ON IT CAN BE FOUND AT THE BEGINING OF THE
3723 ; ROUTINE 'ANLYZO'). THE FAULT ANALYZER ALSO TYPES OUT
3724 ; TWO OCTAL NUMBERS (AND & OR). THESE TWO CAN BE INTERPRETED AS
3725 ; EXPLAINED AT THE BEGINING OF THE ANALYZER.
3726 ; *****
3727 014520 000004 TST26: SCOPE
3728 014522 012737 015146 003056 MOV #TST27,NXTST ; STARTING ADDRESS OF NEXT TEST
3729 014530 004737 036624 JSR PC,CLR WCS ; CLEAR OUT WCS
3730
3731 014534 005077 166274 CLR @WCSST
3732 014540 012700 002000 MOV #2000,R0 ; INITIALIZE WCS LOCATION
3733 014544 005001 CLR R1 ; INITIALIZE WCS DATA
3734 014546 012702 000400 MOV #256,R2 ; COUNT
3735 014552 010077 166260 15: MOV R0,@WCSAR ; WRITE DATA INTO THIS WCS
3736 014556 010177 166256 MOV R1,@WCSDR ; LOCATION
3737 014562 005200 INC R0 ; NEXT WCS ADDRESS
3738 014564 062701 000400 ADD #400,R1 ; NEXT DATA PATTERN
3739 014570 077210 SOB R2,15 ; DONE?
3740
3741 ; READ BACK ALL (256) WORDS THAT WERE WRITTEN AND CHECK
3742 ; IF THEY ARE READ CORRECTLY.
3743
3744 014572 012777 000020 166234 MOV #PARDIS,@WCSST
3745 014600 012700 002000 MOV #2000,R0 ; INITIALIZE WCS LOCATION
3746 014604 005001 CLR R1 ; INITIALIZE WCS DATA
3747 014606 012737 000400 003052 MOV #256,COUNT ; INITIALIZE COUNT
3748 014614 005004 CLR R4 ; R4 WILL CONTAIN A RUNNING 'OR' OF
3749 ; THE WORDS READ
3750 014616 012705 177777 MOV #-1,R5 ; R5 WILL CONTAIN A RUNNING 'AND'
3751 ; OF THE WORDS READ
3752 014622 005037 003046 CLR TMPO ; 'TMPO' WILL CONTAIN 'OR' OF THE
3753 ; DATA PATTERNS THAT FAILED
3754 014626 010537 003050 MOV R5,TMP1 ; 'TMP1' WILL CONTAIN 'AND' OF THE
3755 ; DATA PATTERNS THAT FAILED
3756 014632 112737 177773 003064 MOVB #-5,ERCNT0
3757 014640 112737 177773 003065 MOVB #-5,ERCNT1 ; DONT REPORT MORE THAN 5 ERRORS
3758 014646 012737 014662 001110 MOV #25,$LPERR ; LOOP ON ERROR
3759 014654 012737 015022 000114 MOV #8,$@#114 ; SET PARITY TRAP VECTOR
3760 014662 010077 166150 25: MOV R0,@WCSAR ; READ FROM THIS LOCATION AND
3761 014666 017702 166146 MOV @WCSDR,R2 ; SAVE IT IN R2
3762 014672 050204 BIS R2,R4 ; KEEP RUNNING 'OR' OF DATA
3763 014674 005102 COM R2 ; PATTERNS READ. IN R4
3764 014676 005105 COM R5

```



```

3765 014700 050205      BIS      R2,R5
3766 014702 005105      COM      R5          ;KEEP RUNNING 'AND' OF DATA
3767 014704 005102      COM      R2          ;PATTERNS READ, IN R5
3768 014706 020102      CMP      R1,R2      ;DATA PATTERN READ OK?
3769 014710 001007      BNE      4$          ;NO
3770 014712 005200      3$:      INC      R0          ;SET UP TO READ NEXT ADDRESS
3771 014714 062701 000400  ADD      #400,R1     ;NEXT DATA PATTERN
3772 014720 005337 003052  DEC      COUNT      ;DONE?
3773 014724 001356      BNE      2$
3774 014726 000466      BR       6$
3775
3776                      ;DATA MISCOMPARE OCCURRED ON READING THE DATA FROM THE
3777                      ;WCS LOCATION.
3778                      ;R0 CONTAINS WCS LOCATION READ
3779                      ;R1 CONTAINS EXPECTED DATA
3780                      ;R2 CONTAINS DATA THAT WAS READ
3781
3782 014730 105237 003064  4$:      INCB   ERCNT0      ;DONT REPORT MORE THAN 5 ERRORS
3783 014734 001407      BEQ      5$
3784 014736 010037 001162  MOV      R0,$REG0    ;SAVE WCS LOCATION GIVING DATA
3785                      ;ERROR
3786 014742 010137 001164  MOV      R1,$REG1    ;SAVE EXPECTED DATA
3787 014746 010237 001166  MOV      R2,$REG2    ;SAVE DATA RECEIVED
3788 014752 104012      ERROR   12          ;DATA MISCOMPARE OCCURRED ON READING
3789                      ;WCS LOCATION. POSSIBLE FAULT IN
3790                      ;ADDRESS LINES FOR DATA ARRAY
3791                      ;<35:24>. THIS TEST CONTAINS A
3792                      ;FAULT ANALYZER WHICH ATTEMPTS TO
3793                      ;ANALYZE THE FAULT
3794 014754 010146      5$:      MOV      R1,-(SP)    ;MAKE SURE THE DATA MISCOMPARE IS
3795 014756 042716 000377  BIC      #377,(SP)   ;IN BITS D<31:24> OF THE WCS
3796 014762 010246      MOV      R2,-(SP)    ;DATA ARRAY. IF OUTSIDE DONT DO
3797 014764 042716 000377  BIC      #377,(SP)   ;FAULT ANALYSIS.
3798 014770 022626      CMP      (SP)+,(SP)+
3799 014772 001747      BEQ      3$          ;SKIP FAULT ANALYSIS
3800 014774 005203      INC      R3          ;KEEP COUNT OF ERRORS IN DATA
3801                      ;ARRAY BITS <31:24>
3802 014776 050237 003046  BIS      R2,TMPO     ;'OR' THE FAILING DATA PATTERN IN
3803                      ;'TMPO'
3804 015002 005102      COM      R2
3805 015004 005137 003050  COM      TMP1        ;'AND' THE FAILING DATA PATTERN
3806 015010 050237 003050  BIS      R2,TMP1     ;WITH 'TMP1'
3807 015014 005137 003050  COM      TMP1
3808 015020 000734      BR       3$          ;RETURN TO DO FURTHER CHECKING
3809
3810                      ;IF A PARITY ERROR OCCURRED ON READING A WCS LOCATION.
3811                      ;ENTER HERE
3812
3813 015022 012737 003344 000114  8$:      MOV      #BADPAR,@#114 ;PARITY ERROR HANDLER
3814 015030 032777 100000 165776  BIT      #PARERR,@WCSST ;WAS IT A WCS PARITY ERROR?
3815 015036 001420      BEQ      10$         ;NO
3816 015040 022626      CMP      (SP)+,(SP)+ ;RESTORE THE STACK POINTER
3817 015042 017737 165766 001166  MOV      @WCSST,$REG2 ;SAVE STATUS REGISTER
3818 015050 012777 000020 165756  MOV      #PARDIS,@WCSST ;CLEAR PARITY ERROR
3819 015056 105237 003065  INCB   ERCNT1      ;DONT REPORT MORE THAN 5 ERRORS
3820 015062 001405      BEQ      9$

```

```

3821 015064 010037 001162      MOV      R0,$REG0      ;SAVE ADDRESS GIVING PARITY ERROR
3822 015070 010137 001164      MOV      R1,$REG1      ;SAVE EXPECTED DATA
3823 015074 104026              ERROR    26            ;PARITY ERROR OCCURRED ON READING
3824                                ;A WCS LOCATION NOTE THE PARITY
3825                                ;ERROR CAN BE FROM ANY ONE OF THE
3826                                ;THREE 16-BIT WORDS OF THE DATA
3827                                ;D<47:0>
3828 015076 000705              95:     BR      35      ;DO THE REST OF CHECKING
3829
3830 015100 000137 003344      105:    JMP      BADPAR      ;IF PARITY ERROR IS NOT FROM THE
3831                                ;WCS GO TO THE COMMON PARITY
3832                                ;ERROR HANDLER
3833
3834                                ;GO TO THE FAULT ANALYZER AND PRINT OUT WHICH ADDRESS LINES
3835                                ;ARE FAULTY.
3836
3837 015104 012737 003344 000114 65:  MOV      #BADPAR, @#114 ;PARITY ERROR HANDLER
3838 015112 005703              TST      R3            ;ANY ERROR IN D<31:24>?
3839 015114 001414              BEQ      TST27         ;EXIT
3840 015116 013700 003046      MOV      TMP0,R0       ;GET 'OR'
3841 015122 013701 003050      MOV      TMP1,R1       ;GET 'AND'
3842 015126 000300              SWAB     R0            ;MAP BITS <15:8> TO <7:0>
3843 015130 000301              SWAB     R1            ;POSITION THE 'AND' AND 'OR' SO
3844                                ;AS TO COVER THE ADDRESS
3845 015132 012746 043026      MOV      #MSG26,-(SP)  ;MESSAGE POINTER
3846 015136 012746 177400      MOV      #177400,-(SP);BIT MASK
3847 015142 004737 037350      JSR      PC,ANLYSO    ;GO TO THE FAULT ANALYZER
3848

```

```

;*****
;*TEST 27 CHECK ADDRESSING FOR DATA ARRAY (35:24), ADDRESS (9:8)
;THIS TEST CHECKS OUT THE ADDRESSING SCHEME FOR THE DATA
;ARRAY <35:24>. THIS GROUP OF 12 CHIPS IS ADDRESSED BY
;THE SAME SET OF ADDRESS SELECTORS. THERE ARE 10 ADDRESS LINES
;<9:0>. THIS TEST CHECKS THE COMMON ADDRESS LINES <9:8> AND REPORTS
;ERRORS. THE FAULT ANALYZER ATTEMPTS TO FIGURE OUT WHICH
;ONE OF THE ADDRESS LINES WAS FAULTY.

;STARTING AT LOCATION 375 (OCTAL, TREATING THE WCS AS A
;16 X 3K DATA ARRAY) AN INCREMENTING COUNT PATTERN IS WRITTEN
;IN 16 LOCATIONS OF THE VERTICAL ARRAY D<27:24>.
;THIS IS EQUIVALENT TO WRITING ADDRESS AS DATA INTO A
;16 WORD DATA ARRAY CORRESPONDING TO A<11:8> AND
;D<27:24>. THEN A READ IS DONE AND DATA COMPARED.

;LOGICAL 'AND' AND 'OR' FUNCTIONS ARE PERFORMED ON ALL
;THE PATTERNS READ.
;A SEPERATE LOGICAL 'AND' AND 'OR' FUNCTIONS ARE PERFORMED
;FOR ONLY THOSE PATTERNS THAT FAIL.
;THE ABOVE INFORMATION IS USED BY THE FAULT ANALYZER ROUTINE
;(MORE INFO ON IT CAN BE FOUND AT THE BEGINING OF
;THE ROUTINE 'ANLYZO'). THE FAULT ANALYZER ALSO TYPES OUT
;TWO OCTAL NUMBERS (AND & OR). THESE TWO CAN BE
;INTERPRETED AS EXPLAINED AT THE BEGINING OF THE ANALYZER.

```

```

3874
3875 015146 000004      TST27:  SCOPE
3876 015150 012737 015576 003056      MOV      #TST30,NXTST ;STARTING ADDRESS OF NEXT TEST

```



```

3877
3878 015156 004737 036624      JSR    PC,CLR WCS      ;CLEAR OUT WCS
3879 015162 005077 165646      CLR    @WCSST
3880 015166 012700 002000      MOV    #2000,R0       ;INITIALIZE WCS LOCATION (253)
3881                                     ;77, 01 (CONCATENATE)
3882 015172 005001      CLR    R1             ;INITIALIZE WCS DATA
3883 015174 012702 000020      MOV    #16,R2
3884 015200 010077 165632      MOV    R0,@WCSAR     ;WRITE DATA INTO THIS
3885 015204 010177 165630      MOV    R1,@WCSDR     ;WCS LOCATION
3886 015210 062700 000100      ADD    #100,R0       ;NEXT WCS LOCATION
3887 015214 062701 000400      ADD    #400,R1       ;NEXT DATA PATTERN
3888 015220 077211      SOB    R2,1$         ;DONE?
3889
3890                                     ;READ BACK ALL 16 WORDS THAT WERE WRITTEN AND CHECK IF
3891                                     ;THEY ARE READ CORRECTLY.
3892
3893 015222 012777 000020 165604  MOV    #PARDIS,@WCSST
3894 015230 012700 002000      MOV    #2000,R0       ;INITIALIZE WCS LOCATION
3895 015234 005001      CLR    R1             ;INITIALIZE WCS DATA
3896 015236 005004      CLR    R4             ;R4 WILL CONTAIN A RUNNING 'OR'
3897                                     ;OF THE WORDS READ
3898 015240 012705 177777      MOV    #-1,R5        ;R5 WILL CONTAIN A RUNNING 'AND'
3899                                     ;OF THE WORDS READ
3900 015244 005037 003046      CLR    TMP0          ;'TMP0' WILL CONTAIN 'OR' OF THE
3901                                     ;DATA PATTERN THAT FAILED.
3902 015250 010537 003050      MOV    R5,TMP1       ;'TMP1' WILL CONTAIN 'AND' OF THE
3903                                     ;DATA PATTERNS THAT FAILED
3904 015254 112737 177773 003064  MOVB   #-5,ERCNT0
3905 015262 112737 177773 003065  MOVB   #-5,ERCNT1    ;DONT REPORT MORE THAN 5 ERRORS
3906 015270 012737 000020 003052  MOV    #16,COUNT     ;INITIALIZE COUNT
3907 015276 012737 015312 001110  MOV    #25,$LPERR    ;LOOP ON ERROR
3908 015304 012737 015454 000114  MOV    #8,$@#114     ;SET PARITY TRAP VECTOR
3909 015312 010077 165520      MOV    R0,@WCSAR     ;READ FROM THIS LOCATION AND
3910 015316 017702 165516      MOV    @WCSDR,R2     ;SAVE THIS IN R2
3911 015322 050204      BIS    R2,R4         ;KEEP RUNNING 'OR' OF DATA
3912 015324 005102      COM    R2            ;PATTERNS READ, IN R4
3913 015326 005105      COM    R5
3914 015330 050205      BIS    R2,R5        ;KEEP RUNNING 'AND' OF PATTERNS
3915 015332 005105      COM    R5            ;READ IN R5
3916 015334 005102      COM    R2
3917 015336 020102      CMP    R1,R2        ;DATA PATTERNS READ OK?
3918 015340 001010      BNE    4$           ;NO
3919 015342 062700 000100      ADD    #100,P0       ;SET UP TO READ NEXT ADDRESS
3920 015346 062701 000400      ADD    #400,R1       ;NEXT DATA PATTERN
3921 015352 005337 003052      DEC    COUNT         ;DONE?
3922 015356 001355      BNE    2$           ;YES
3923 015360 000463      BR     6$
3924
3925                                     ;DATA MISCOMPARE OCCURRED ON READING THE DATA FROM THE
3926                                     ;WCS LOCATION.
3927                                     ;R0 CONTAINS WCS LOCATION READ
3928                                     ;R1 CONTAINS EXPECTED DATA
3929                                     ;R2 CONTAINS DATA THAT WAS READ
3930
3931 015362 105237 003064      INCB   ERCNT0        ;DONT REPORT MORE THAN 5 ERRORS
3932 015366 001407      BEQ    5$

```

```

3933 015370 010037 001162      MOV      R0,$REG0      ;SAVE WCS LOCATION GIVING DATA ERROR
3934 015374 010137 001164      MOV      R1,$REG1      ;SAVE EXPECTED DATA
3935 015400 010237 001166      MOV      R2,$REG2      ;SAVE DATA RECEIVED
3936 015404 104012      ERROR    12            ;DATA MISCOMPARE OCCURRED ON READING
3937                                ;WCS LOCATION. (POSSIBLE FAULT
3938                                ;IN ADDRESS LINES FOR DATA
3939                                ;ARRAY (35:24)). THIS TEST
3940                                ;CONTAINS A FAULT ANALYZER WHICH
3941                                ;ATTEMPTS TO ANALYZE THE FAULT.
3942 015406 010146      55:      MOV      R1,-(SP)      ;MAKE SURE THE DATA MISCOMPARE IS
3943 015410 042716 170377      BIC      #170377,(SP)  ;IN BITS D<27:24> OF THE
3944 015414 010246      MOV      R2,-(SP)      ;WCS DATA ARRAY
3945 015416 042716 170377      BIC      #170377,(SP)
3946 015422 022626      CMP      (SP)+,(SP)+
3947 015424 001746      BEQ      35            ;SKIP FAULT ANALYSIS
3948 015426 005203      INC      R3            ;KEEP COUNT OF ERROR IN DATA
3949                                ;BITS D<27:24>
3950 015430 050237 003046      BIS      R2,TMPO      ;'OR' THE FAILING DATA PATTERNS
3951                                ;INTO 'TMPO'
3952 015434 005102      COM      R2
3953 015436 005137 003050      COM      TMP1
3954 015442 050237 003050      BIS      R2,TMP1      ;'AND' THE FAILING PATTERN WITH
3955 015446 005137 003050      COM      TMP1          ;'TMP1'
3956 015452 000733      BR       35            ;RETURN TO DO FURTHER CHECKING
3957
3958                                ;IF A PARITY ERROR OCCURRED ON READING A WCS LOCATION,
3959                                ;ENTER HERE.
3960
3961 015454 012737 003344 000114 85:      MOV      #BADPAR,@#114 ;PARITY ERROR HANDLER
3962 015462 052777 100000 165344      BIS      #PARERR,@WCSST ;WCS PARITY ERROR?
3963 015470 001415      BEQ      105          ;NO
3964 015472 022626      CMP      (SP)+,(SP)+  ;RESTORE THE STACK POINTER
3965 015474 017737 165334 001166      MOV      @WCSST,$REG2  ;SAVE STATUS REGISTER
3966 015502 012777 000020 165324      MOV      #PARDIS,@WCSST ;CLEAR OUT THE PARITY ERROR
3967 015510 010037 001162      MOV      R0,$REG0      ;SAVE ADDRESS GIVING PARITY ERROR
3968 015514 010137 001164      MOV      R1,$REG1      ;SAVE EXPECTED DATA
3969 015520 104026      ERROR    26            ;PARITY ERROR OCCURRED ON READING
3970                                ;A WCS LOCATION. THE PARITY
3971                                ;ERROR CAN BE FROM ANY ONE OF
3972                                ;THE THREE 16-BIT WORDS OF THE
3973                                ;DATA ARRAY D<47:0>
3974 015522 000707      95:      BR       35            ;DO REST OF THE CHECKING
3975
3976 015524 000137 003344      105:     JMP      BADPAR        ;IF PARITY ERROR IS NOT FROM
3977                                ;THE WCS GO TO THE COMMON
3978                                ;PARITY ERROR HANDLER
3979
3980                                ;GO TO THE FAULT ANALYZER AND PRINT OUT WHICH ADDRESS
3981                                ;LINES ARE FAULTY.
3982
3983 015530 012777 003344 162356 65:      MOV      #BADPAR,@114 ;PARITY ERROR HANDLER
3984 015536 005703      TST      R3            ;ANY ERROR IN D<27:24>?
3985 015540 001416      BEQ      TST30         ;EXIT
3986 015542 013700 003046      MOV      TMPO,R0       ;GET 'OR'
3987 015546 013701 003050      MOV      TMP1,R1       ;GET 'AND'
3988 015552 006200      ASR      R0            ;POSITION 'AND', 'OR' SO AS TO

```



3989 015554 006200  
3990 015556 006201  
3991 015560 006201  
3992 015562 012746 043026  
3993 015566 012746 176077  
3994 015572 004737 037350

ASR R0 ;COVER THE ADDRESS LINES BEING CHECKED  
ASR R1 ;MAP BITS <11:8> TO <9:5>  
ASR R1  
MOV #MSG26,-(SP) ;MESSAGE POINTER  
MOV #176077,-(SP) ;BIT MASK?  
JSR PC,ANLYSO ;GO TO THE FAULT ANALYZER

;;\*\*\*\*\*  
;\*TEST 30 CHECK ADDRESSING FOR DATA ARRAY (47:36), ADDRESS (9:0)  
;THIS TEST CHECKS OUT THE ADDRESSING SCHEME FOR DATA ARRAY <47:36>.  
;THIS GROUP OF 12 CHIPS IS ADDRESSED BY THE SAME SET OF ADDRESS  
;SELECTORS. THERE ARE 10 ADDRESS LINES <9:0>. THIS TEST CHECKS  
;ALL THE COMMON ADDRESS LINES AND REPORTS ERRORS. THE FAULT  
;ANALYZER ATTEMPTS TO FIGURE OUT WHICH ONE OF THE ADDRESS  
;LINES WAS FAULTY.  
  
;STARTING AT LOCATION 4000 (TREATING THE WCS AS A 16 X 3K DATA  
;ARRAY) AN INCREMENTING COUNT PATTERN IS WRITTEN IN 1024  
;CONSECUTIVE LOCATIONS OF THE VERTICAL ARRAY D<45:36>. THIS  
;IS EQUIVALENT TO WRITING ADDRESS AS DATA INTO A 1024  
;WORD DATA ARRAY CORRESPONDING TO A<9:0> AND D<45:36>.  
;THEN A READ IS DONE AND DATA COMPARED.  
  
;LOGICAL 'AND' AND 'OR' FUNCTIONS ARE PERFORMED ON ALL THE PATTERN  
;READ.  
;A SEPERATE LOGICAL 'AND' AND 'OR' IS PERFORMED FOR ONLY  
;THOSE PATTERNS THAT FAIL.  
;THE ABOVE INFORMATION IS USED BY THE FAULT ANALYZER ROUTINE  
;(MORE INFO ON IT CAN BE FOUND AT THE BEGINING OF  
;THE ROUTINE 'ANLYZO') THE FAULT ANALYZER ALSO TYPES  
;OUT TWO OCTAL NUMBERS (AND & OR). THESE TWO CAN  
;BE INTERPRETED AS EXPLAINED AT THE BEGINING OF THE ANALYZER.

4022 015576 000004  
4023 015600 012737 016240 003056  
4024 015606 004737 036624  
4025 015612 005003  
4026 015614 005077 165214  
4027 015620 012700 004000  
4028 015624 005001  
4029 015626 012702 002000  
4030 015632 010077 165200  
4031 015636 010177 165176  
4032 015642 005200  
4033 015644 062701 000020  
4034 015650 077210

;;\*\*\*\*\*  
TST30: SCOPE  
MOV #TST31,NXTST ;STARTING ADDRESS OF NEXT TEST  
JSR PC,CLR WCS ;CLEAR OUT WCS  
CLR R3 ;CLEAR ERROR INDICATOR  
CLR @WCSST  
MOV #4000,R0 ;INITIALIZE WCS LOCATION  
CLR R1 ;INITIALIZE WCS DATA  
MOV #1024,R2 ;INITIALIZE COUNT  
15: MOV R0,@WCSAR ;WRITE DATA INTO THIS  
MOV R1,@WCSOR ;WCS LOCATION  
INC R0 ;NEXT WCS ADDRESS  
ADD #20,R1 ;NEXT DATA PATTERN  
SOB R2,15 ;DONE?

;READ BACK ALL (1024) WORDS THAT WERE WRITTEN AND CHECK  
;IF THEY ARE READ CORRECTLY.

4039 015652 012777 000020 165154  
4040 015660 012700 004000  
4041 015664 005001  
4042 015666 012737 002000 003052  
4043 015674 005004  
4044

MOV #PARDIS,@WCSST  
MOV #4000,R0 ;INITIALIZE WCS LOCATION  
CLR R1 ;INITIALIZE WCS DATA  
MOV #1024,COUNT ;INITIALIZE COUNT  
CLR R4 ;R4 WILL CONTAIN A RUNNING 'OR'  
;OF THE WORDS READ

```

4045 015676 012705 177777      MOV      #-1,R5      ;R5 WILL CONTAIN A RUNNING 'AND'
4046                                ;OF THE WORDS READ
4047 015702 005037 003046      CLR      TMP0        ;'TMP0' WILL CONTAIN 'OR' OF THE
4048                                ;DATA PATTERNS THAT FAILED
4049 015706 010537 003050      MOV      R5,TMP1     ;'TMP1' WILL CONTAIN 'AND' OF THE
4050                                ;DATA PATTERNS THAT FAILED
4051 015712 112737 177773 003064  MOVB     #-5,ERCNT0  ;DONT REPORT MORE THAN 5 ERRORS
4052 015720 112737 177773 003065  MOVB     #-5,ERCNT1  ;
4053 015726 012737 015742 001110  MOV      #25,$LPERR  ;LOOP ON ERROR
4054 015734 012737 016102 000114  MOV      #85,@#114   ;SET PARITY TRAP VECTOR
4055 015742 010077 165070      MOV      R0,@WCSAR   ;HEAD FROM THIS LOCATION AND
4056 015746 017702 165066      MOV      @WCSDR,R2   ;SAVE IT IN R2
4057 015752 050204      BIS      R2,R4       ;KEEP RUNNING 'OR' OF THE DATA
4058                                ;PATTERNS READ IN R4
4059 015754 005102      COM      R2
4060 015756 005105      COM      R5          ;KEEP RUNNING 'AND' OF PATTERNS
4061 015760 050205      BIS      R2,R5      ;READ, IN R5
4062 015762 005105      COM      R5
4063 015764 005102      COM      R2
4064 015766 020102      CMP      R1,R2      ;DATA PATTERN READ OK?
4065 015770 001007      BNE      45         ;NO
4066 015772 005200      35: INC      R0       ;SET UP TO READ NEXT ADDRESS
4067 015774 062701 000020      ADD      #20,R1     ;NEXT DATA PATTERN
4068 016000 005337 003052      DEC      C0JNT      ;DONE?
4069 016004 001356      BNE      25
4070 016006 000466      BR       65
4071
4072                                ;DATA MISCOMPARE OCCURRED ON READING THE DATA FROM THE
4073                                ;WCS LOCATION.
4074                                ;R0 CONTAINS WCS LOCATION READ
4075                                ;R1 CONTAINS EXPECTED DATA
4076                                ;R2 CONTAINS DATA THAT WAS READ
4077
4078 016010 105237 003064      45: INCB     ERCNT0   ;DONT REPORT MORE THAN 5 ERRORS
4079 016014 001407      BEQ      55
4080 016016 010037 001162      MOV      R0,$REG0    ;SAVE WCS LOCATION GIVING
4081                                ;DATA ERROR
4082 016022 010137 001164      MOV      R1,$REG1    ;SAVE EXPECTED DATA
4083 016026 010237 001166      MOV      R2,$REG2    ;SAVE DATA RECEIVED
4084 016032 104012      ERROR    12         ;DATA MISCOMPARE OCCURRED ON READING
4085                                ;WCS LOCATION. POSSIBLE FAULT IN
4086                                ;ADDRESS LINES FOR DATA ARRAY <47:36>
4087                                ;THIS TEST CONTAINS A FAULT ANALYZER
4088                                ;WHICH ATTEMPTS TO ANALYZE THE FAULT.
4089 016034 010146      55: MOV      R1,-(SP)   ;MAKE SURE THE DATA MISCOMPARE IS
4090 016036 042716 140013      BIC      #140013,(SP) ;IN BITS D<45:36> OF THE WCS
4091 016042 010246      MOV      R2,-(SP)   ;DATA ARRAY
4092 016044 042716 140013      BIC      #140013,(SP)
4093 016050 022626      CMP      (SP)+,(SP)+
4094 016052 001747      BEQ      35         ;SKIP FAULT ANALYSIS
4095 016054 005203      INC      R3         ;KEEP COUNT OF ERRORS IN DATA
4096                                ;ARRAY BITS D<45:36>
4097 016056 050237 003046      BIS      R2,TMP0    ;'OR' THE FAILING DATA PATTERN IN
4098                                ;'TMP0'
4099 016062 005102      COM      R2
4100 016064 005137 003050      COM      TMP1       ;'AND' THE FAILING DATA PATTERN
```



```
4101 016070 050237 003050 BIS R2,TMP1 ;WITH 'TEMP1'
4102 016074 005137 003050 COM TMP1
4103 016100 000734 BR 3% ;RETURN TO DO FURTHER CHECKING
4104
4105 ;IF A PARITY ERROR OCCURRED ON READING A WCS LOCATION,
4106 ;ENTER HERE.
4107
4108 016102 012737 003344 000114 8%: MOV #BADPAR,@#114 ;PARITY ERROR HANDLER
4109 016110 032777 100000 164716 BIT #PARERR,@WCSST ;WCS PARITY ERROR?
4110 016116 001420 BEQ 10% ;NO
4111 016120 022626 CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
4112 016122 017737 164706 001166 MOV @WCSST,$REG2 ;SAVE STATUS REGISTER
4113 016130 012777 000020 164676 MOV #PARDIS,@WCSST ;CLEAR PARITY ERROR
4114 016136 105237 003065 INCB ERCNT1 ;DONT REPORT MORE THAN 5 ERRORS
4115 016142 001405 BEQ 9%
4116 016144 010037 001162 MOV R0,$REG0 ;SAVE ADDRESS GIVING PARITY ERROR
4117 016150 010137 001164 MOV R1,$REG1 ;SAVE EXPECTED DATA
4118 016154 104026 ERROR 26 ;PARITY ERROR OCCURRED ON READING A
4119 ;WCS LOCATION. THE PARITY ERROR
4120 ;CAN BE FROM ANY ONE OF THE THREE
4121 ;16 BIT WORDS OF THE DATA D<47:0>
4122 016156 000705 9%: BR 3% ;DO THE REST OF CHECKING
4123
4124 016160 000137 003344 10%: JMP BADPAR ;IF PARITY ERROR IS NOT FROM
4125 ;THE WCS GO TO THE COMMON
4126 ;PARITY ERROR HANDLER
4127 ;GO TO THE FAULT ANALYZER AND PRINT OUT WHICH ADDRESS
4128 ;LINES ARE FAULTY.
4129
4130 016164 012737 003344 000114 6%: MOV #BADPAR,@#114 ;PARITY ERROR HANDLER
4131 016172 005077 164636 CLR @WCSST
4132 016176 005703 TST R3 ;ANY ERROR IN D<45:36>?
4133 016200 001417 BEQ TST31 ;;EXIT
4134 016202 013700 003046 MOV TMP0,R0 ;GET 'OR'
4135 016206 013701 003050 MOV TMP1,R1 ;GET 'AND'
4136 016212 072027 177774 ASH #-4,R0 ;POSITION 'AND', 'OR' SO AS TO
4137 016216 072127 177774 ASH #-4,R1 ;MAP BITS <13:4> TO <9:0>
4138 016222 006201 ASR R1
4139 016224 012746 043060 MOV #MSG27,-(SP) ;MESSAGE POINTER
4140 016230 042746 176000 BIC #176000,-(SP) ;BIT MASK
4141 016234 004737 037350 JSR PC,ANLYSO ;GO TO THE FAULT ANALYZER
4142
4143
4144 ;*****
4145 ;*TEST 31 CHECK ADDRESS REGISTER - ADREG(11:0)
4146 ;THIS TESTS CHECKS THAT THE ADDRESS REGISTER (ADREG<11:0>)
4147 ;BY RIPPLING A COUNT PATTERN FROM 0 TO 5777.
4148 ;*****
4149 016240 000004 TST31: SCOPE
4150 016242 012737 016360 003056 MOV #TST32,NXTST ;STARTING ADDRESS OF NEXT TEST
4151 016250 013704 003036 MOV WCSAR,R4
4152 016254 013705 003040 MOV WCSR,R5
4153 016260 012777 000020 164546 MOV #PARDIS,@WCSST
4154 016266 012703 177766 MOV #-10.,R3 ;DONT REPORT MORE THAN 10 ERRORS
4155 016272 005001 CLR R1
4156 016274 010114 1%: MOV R1,@R4 ;ADDRESS THE WCS
```

```
4157 016276 010115      MOV    R1,@R5      ;WRITE DATA INTO WCS
4158 016300 005201      INC    R1          ;NEXT LOC
4159 016302 020127 006000  CMP    R1,#3072.   ;DONE?
4160 016306 001372      BNE    1$
4161
4162 016310 005001      CLR    R1
4163 016312 010114      2$:   MOV    R1,@R4      ;ADDRESS THE WCS
4164 016314 011502      MOV    @R5,R2     ;READ DATA
4165 016316 020102      CMP    R1,R2     ;DATA CORRECT?
4166 016320 001411      BEQ    3$        ;OK
4167 016322 005203      INC    R3        ;NO MORE THAN 10 ERRORS
4168 016324 100013      BPL    4$
4169 016326 010137 001162  MOV    R1,$REGO   ;GET WCS LOC
4170 016332 010137 001164  MOV    R1,$REG1   ;GET EXPCTD DATA
4171 016336 010237 001166  MOV    R2,$REG2   ;GET DATA RECVD
4172 016342 104012      ERROR  12        ;DATA ERROR ON READING FROM WCS
4173
4174 016344 005201      3$:   INC    R1        ;NEXT LOC.
4175 016346 020127 006000  CMP    R1,#3072.   ;DONE?
4176 016352 001357      BNE    2$
4177
4178 016354 005077 164454  4$:   CLR    @WCSST
4179
4180      ;*****
4181      ;*TEST 32      TEST DATA ARRAY-MARCHING 1'S AND 0'S
4182      ;THIS IS A MEMORY-TYPE TEST OF THE WCS DATA ARRAY (16 X 3K); AND
4183      ;IS KNOWN AS THE "MARCHING ONES AND ZEROES" TEST.  THE MEMORY
4184      ;IS FIRST WRITTEN TO THE ALL ZEROES STATE.  THEN SEQUENTIALLY,
4185      ;STARTING AT THE FIRST ADDRESS, THE ZERO IS READ AND A
4186      ;ONE (177777) IS WRITTEN.  THIS SEQUENCE IS CONTINUED TO THE
4187      ;LAST LOCATION.  AT THIS STAGE, THE DATA ARRAY IS FULL OF 1'S.
4188      ;THEN STARTING AT THE HIGHEST LOCATION, A ONE IS READ AND
4189      ;A ZERO IS WRITTEN.  THE ADDRESS IS DECREMENTED AND THE
4190      ;SEQUENCE IS REPEATED UNTILL THE LOWEST LOCATION IS REACHED.
4191      ;THIS TEST WILL DETECT FAULTS IN THE INDIVIDUAL MEMORY CHIP
4192      ;ADDRESSING LINES AS WELL AS DATA PROBLEMS.  GENERALLY, ADDRESSING
4193      ;FAULTS WILL SHOW UP IN REPEATED ERRORS WITH A DEFINITE FAILURE
4194      ;PATTERN.
4195      ;*****
4196 016360 000004      TST32: SCOPE
4197 016362 012737 016576 003056  MOV    #TST33,NXTST ;STARTING ADDRESS OF NEXT TEST
4198 016370 012737 177766 003052  MOV    #-10.,COUNT ;DONT REPORT MORE THAN
4199 016376 012737 177766 003054  MOV    #-10.,COUNT ;10 ERRORS
4200 016404 012777 000020 164422  MOV    #PARDIS,@WCSST ;DISABLE PARITY ERROR TRAPS
4201 016412 013704 003036  MOV    WCSAR,R4    ;WCS ADDRESS REGISTER
4202 016416 013705 003040  MOV    WCSDR,R5    ;WCS DATA REGISTER
4203 016422 012703 177777  MOV    #-1,R3     ;INITIALIZE DATA PATTERN TO BE WRITTEN
4204
4205      ;CLEAR THE ENTIRE WCS DATA ARRAY
4206
4207 016426 012701 006000  MOV    #3072.,R1
4208 016432 005014      CLR    @R4        ;ADDRESS THE WCS
4209 016434 005015      1$:   CLR    @R5        ;CLEAR THE LOCATION
4210 016436 005214      INC    @R4
4211 016440 077103      SOB    R1,1$     ;DONE?
4212
```



```
4213 ;WRITE AND READ "MARCHING ONES."  
4214 ;STARTING AT THE LOWEST ADDRESS, EACH LOCATION IS READ AND CHECKED  
4215 ;IF IT WAS 0, THEN IT IS WRITTEN WITH ALL 1'S. THE TEST IS REPEATED  
4216 ;INCREMENTALLY FOR ALL ADDRESSES.  
4217  
4218 016442 012700 006000 MOV #3072.,R0 ;INITIALIZE COUNT  
4219 016446 005001 CLR R1  
4220 016450 012737 016456 001110 MOV #2$,SLPERR ;LOOP ON ERROR  
4221 016456 010114 2$: MOV R1,@R4 ;ADDRESS THE WCS LOCATION  
4222 016460 011502 MOV @R5,R2 ;READ THE LOCATION  
4223 016462 001412 BEQ 3$ ;OK, IF IT WAS 0  
4224 016464 005237 003052 INC COUNT ;NO MORE THAN 10 ERRORS  
4225 016470 100007 BPL 3$  
4226 016472 010137 001162 MOV R1,$REG0 ;SAVE ADDRESS THAT WAS READ  
4227 016476 005037 001164 CLR $REG1 ;SAVE EXPECTED DATA PATTERN (0)  
4228 016502 010237 001166 MOV R2,$REG2 ;SAVE DATA PATTERN THAT WAS READ  
4229 016506 104012 ERROR 12 ;DATA ERROR! LOCATION(GIVEN IN  
4230 ;ERROR MESSAGE) WAS READ AND IT  
4231 ;WAS NOT 0. THE ENTIRE DATA  
4232 ;ARRAY WAS CLEARED PREVIOUSLY.  
4233 016510 010315 3$: MOV R3,@R5 ;WRITE A PATTERN OF ALL 1'S IN THIS  
4234 ;LOCATION  
4235 016512 005201 INC R1 ;POINT TO THE NEXT WCS LOC.  
4236 016514 077020 SOB R0,2$ ;DONE?  
4237  
4238 ;AT THIS POINT, THE ENTIRE WCS DATA ARRAY (16X3K) SHOULD CONTAIN  
4239 ;ALL 1'S. THE FOLLOWING IS A "MARCHING ZERO" TEST. STARTING  
4240 ;AT THE HIGHEST ADDRESS, EACH LOCATION IS READ AND CHECKED  
4241 ;IF IT WAS ALL 1'S, THEN IT IS WRITTEN TO BE 0. THE SAME  
4242 ;TEST IS REPEATED (DECREMENTALLY) FOR ALL LOCATIONS.  
4243  
4244 016516 012701 005777 MOV #3071.,R1 ;INITIALIZE WCS ADDRESS (LAST)  
4245 016522 012700 006000 MOV #3072.,R0 ;INITIALIZE COUNT  
4246 016526 012737 016534 001110 MOV #4$,SLPERR ;LOOP ON ERROR  
4247 016534 010114 4$: MOV R1,@R4 ;ADDRESS THE WCS LOC.  
4248 016536 011502 MOV @R5,R2 ;READ THAT LOC.  
4249 016540 020203 CMP R2,R3 ;DATA CORRECT? (177777)  
4250 016542 001412 BEQ 5$ ;YES  
4251 016544 005237 003054 INC COUNT0  
4252 016550 100012 BPL TST33 ;EXIT  
4253 016552 010137 001162 MOV R1,$REG0 ;SAVE WCS ADDRESS  
4254 016556 010337 001164 MOV R3,$REG1 ;SAVE EXPECTED DATA (177777)  
4255 016562 010237 001166 MOV R2,$REG2 ;SAVE DATA RECIEVED  
4256 016566 104012 ERROR 12 ;DATA ERROR! LOCATION (GIVEN IN  
4257 ;ERROR MESSAGE) WAS READ AND  
4258 ;IT DID NOT CONTAIN 177777 AS  
4259 ;EXPECTED.  
4260 016570 005015 5$: CLR @R5 ;CLEAR THAT WCS LOCATION  
4261 016572 005301 DEC R1 ;POINT TO NEXT ADDRESS  
4262 016574 077021 SOB R0,4$ ;DONE?  
4263  
4264 ;*****  
4265 ;*TEST 33 TEST DATA ARRAY MARCHING 0'S AND 1'S  
4266 ;THIS IS A MEMORY-TYPE TEST OF THE WCS DATA ARRAY (16 X 3K);  
4267 ;AND IS KNOWN AS THE "MARCHING ONES AND ZEROES" TEST. THE  
4268 ;MEMORY IS FIRST WRITTEN WITH ALL 1'S. THEN SEQUENTIALLY, STARTING
```



```

4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279 016576 000004
4280 016600 012737 017014 003056
4281 016606 012777 000020 164220
4282 016614 012737 177766 003052
4283 016622 012737 177766 003054
4284 016630 013704 003036
4285 016634 013705 003040
4286 016640 012703 177777
4287
4288
4289
4290 016644 012701 006000
4291 016650 005014
4292 016652 010315
4293 016654 005214
4294 016656 077103
4295
4296
4297
4298
4299
4300
4301 016660 012700 006000
4302 016664 005001
4303 016666 012737 016674 001110
4304 016674 010114
4305 016676 011502
4306 016700 020203
4307 016702 001412
4308 016704 005237 003052
4309 016710 100007
4310 016712 010137 001162
4311 016716 010337 001164
4312 016722 010237 001166
4313 016726 104012
4314
4315
4316
4317
4318 016730 005015
4319 016732 005201
4320 016734 077021
4321
4322
4323
4324

```

```

; AT THE FIRST ADDRESS, THE PATTERN (177777) IS READ AND A ZERO IS
; WRITTEN. THIS SEQUENCE IS CONTINUED TO THE LAST LOCATION. AT
; THIS STAGE, THE DATA ARRAY IS CLEARED. THEN STARTING
; AT THE HIGHEST LOCATION, A ZERO IS READ AND ALL 1'S (177777)
; IS WRITTEN. THE ADDRESS IS DECREMENTED AND THE SEQUENCE
; IS REPEATED UNTILL THE LOWEST LOCATION IS REACHED.
; THIS TEST WILL DETECT FAULTS IN INDIVIDUAL CHIP ADDRESSING
; LINES AND DATA PROBLEMS. GENERALLY, ADDRESSING FAULTS WILL SHOW
; UP IN REPEATED ERRORS WITH A DEFINITE FAILURE PATTERN.
; *****
TST33: SCOPE
MOV #TST34,NXTST ; STARTING ADDRESS OF NEXT TEST
MOV #PARDIS,@WCSST ; DISABLE PARITY ERROR TRAPS
MOV #-10.,COUNT ; DONT REPORT MORE THAN
MOV #-10.,COUNT0 ; 10 ERRORS
MOV WCSAR,R4 ; WCS ADDRESS REGISTERS
MOV WCSDR,R5 ; WCS DATA REGISTER
MOV #-1,R3 ; INITIALIZE DATA PATTERN

; FILL THE ENTIRE WCS DATA ARRAY WITH 1'S.
MOV #3072.,R1
CLR @R4 ; ADDRESS THE WCS
MOV R3,@R5 ; WRITE A PATTERN (177777)
INC @R4 ; NEXT WCS ADDRESS
SOB R1,1$ ; DONE?

; WRITE AND READ "MARCHING 0'S".
; STARTING AT THE LOWEST ADDRESS, EACH LOCATION IS READ AND
; CHECKED IF IT WAS ALL 1'S, THEN IT IS WRITTEN WITH A 0.
; THE TEST IS REPEATED INCREMENTALLY FOR ALL ADDRESSES.
MOV #3072.,R0 ; INITIALIZE COUNT
CLR R1 ; INITIALIZE WCS LOCATION
MOV #2$,SLPERR ; LOOP ON ERROR
MOV R1,@R4 ; ADDRESS THE WCS LOCATION
MOV @R5,R2 ; READ THE LOCATION
CMP R2,R3 ; DATA CORRECT (177777)?
BEQ 3$ ; YES
INC COUNT
BPL 3$
MOV R1,$REG0 ; SAVE ADDRESS THAT WAS READ
MOV R3,$REG1 ; SAVE EXPECTED DATA PATTERN (177777)
MOV R2,$REG2 ; SAVE DATA PATTERN THAT WAS READ
ERROR 12 ; DATA ERROR! LOCATION (GIVEN IN
; ERROR MESSAGE) WAS READ AND THE
; DATA WAS NOT CORRECT. THE ENTIRE
; ARRAY WAS WRITTEN WITH A BACKGROUND
; PATTERN OF 1'S PREVIOUSLY
CLR @R5 ; WRITE A PATTERN (0) IN THIS LOCATION
INC R1 ; POINT TO THE NEXT WCS LOC.
SOB R0,2$ ; DONE?

; AT THIS POINT, THE ENTIRE WCS DATA AFRAY (16X3K) SHOULD BE
; CLEARED. THE FOLLOWING IS A "MARCHING ONE" TEST. STARTING
; AT THE HIGHEST ADDRESS, EACH LOCATION IS READ AND CHECKED

```



```
4325 ; IF IT WAS ALL 0'S, THEN IT IS WRITTEN TO BE ALL 1'S.  
4326 ; (177777). THE SAME TEST IS REPEATED (DECREMENTALLY FOR ALL  
4327 ; LOCATIONS.  
4328  
4329 016736 012701 005777 MOV #3071.,R1 ; INITIALIZE WCS ADDRESS (LAST)  
4330 016742 012700 006000 MOV #3072.,R0 ; INITIALIZE COUNT  
4331 016746 012737 016754 001110 MOV #45,$LPERR ; LOOP ON ERROR  
4332 016754 010114 45: MOV R1,@R4 ; ADDRESS THE WCS LOC.  
4333 016756 011502 MOV @R5,R2 ; READ THAT LOC.  
4334 016760 001412 BEQ 55 ; DATA CORRECT (0)?  
4335 016762 005237 003054 INC COUNT0  
4336 016766 100012 BPL TST34 ; EXIT  
4337 016770 010137 001162 MOV R1,$REG0 ; SAVE WCS ADDRESS  
4338 016774 005037 001164 CLR $REG1 ; SAVE EXPECTED DATA (0)  
4339 017000 010237 001166 MOV R2,$REG2 ; SAVE DATA RECEIVED  
4340 017004 104012 ERROR 12 ; DATA ERROR! LOCATION (GIVEN IN  
4341 ; ERROR MESSAGE) WAS READ AND  
4342 ; IT DID NOT CONTAIN 0 AS  
4343 ; EXPECTED.  
4344 017006 010315 55: MOV R3,@R5 ; WRITE ALL 1'S (177777) IN THE LOC. ?  
4345 017010 005301 DEC R1 ; POINT TO NEXT ADDRESS  
4346 017012 077020 SOB R0,45 ; DONE?  
4347  
4348 ; *****  
4349 ; *TEST 34 TEST PARITY BIT STORAGE CHIPS-MARCHING 1'S AND 0'S  
4350 ; THIS IS A TEST OF THE WCS PARITY BIT STORAGE CHIP (1 X 1K); AND  
4351 ; IS KNOWN AS THE "MARCHING ONES AND ZEROES" TEST. THE MEMORY  
4352 ; IS FIRST WRITTEN TO THE ALL ZEROES STATE. THEN SEQUENTIALLY,  
4353 ; STARTING AT THE FIRST ADDRESS, THE ZERO IS READ AND A  
4354 ; ONE IS WRITTEN. THIS SEQUENCE IS CONTINUED TO THE  
4355 ; LAST LOCATION. AT THIS STAGE, THE DATA ARRAY IS FULL OF 1'S.  
4356 ; THEN STARTING AT THE HIGHEST LOCATION, A ONE IS READ AND  
4357 ; A ZERO IS WRITTEN. THE ADDRESS IS DECREMENTED AND THE  
4358 ; SEQUENCE IS REPEATED UNTILL THE LOWEST LOCATION IS REACHED.  
4359 ; THIS TEST WILL DETECT FAULTS IN THE INDIVIDUAL MEMORY CHIP  
4360 ; ADDRESSING LINES AS WELL AS DATA PROBLEMS. GENERALLY, ADDRESSING  
4361 ; FAULTS WILL SHOW UP IN REPEATED ERRORS WITH A DEFINITE FAILURE  
4362 ; PATTERN.  
4363 ; THE TEST IS REPEATED FOR THE THREE PARITY BIT STORAGE CHIPS:  
4364 ; CHIP 0, SEGMEN 0, PAR 0, ADDRESSES 0,1,2,.....  
4365 ; CHIP 1, SEGMEN 1, PAR 1, ADDRESSES 2000,2001,.....  
4366 ; CHIP 2, SEGMEN 2, PAR 2, ADDRESSES 4000,4001,.....  
4367 ; *****  
4368 017014 000004 TST34: SCOPE  
4369 017016 012737 017372 003056 MOV #TST35,NXTST ; STARTING ADDRESS OF NEXT TEST  
4370 017024 012737 177766 003052 MOV #-10.,COUNT ; DONT REPORT MORE THAN  
4371 017032 012737 177766 003054 MOV #-10.,COUNT0 ; 10 ERRORS  
4372 017040 012777 000020 163766 MOV #PARDIS,@WCSST ; DISABLE PARITY ERROR TRAPS  
4373 017046 012703 077777 MOV #77777,R3 ; INITIALIZE DATA PATTERN TO BE WRITTEN  
4374 017052 005037 003046 CLR TMP0 ; SEGMENT NUMBER  
4375 017056 012700 000002 MOV #PAR0,R0 ; PAR BIT MASK  
4376 017062 005037 003050 CLR TMP1 ; INIT. WCS ADR.  
4377  
4378 ; CLEAR THE PAR CHIP  
4379  
4380 017066 013701 003050 65: MOV TMP1,R1
```

```

4381 017072 012704 002000      MOV      #1024.,R4
4382 017076 010177 163734      MOV      R1,@WCSAR      ;ADDRESS THE WCS
4383 017102 005077 163732      CLR      @WCSDR        ;CLEAR THE LOCATION
4384 017106 005201              INC      R1
4385 017110 077404              SOB      R4,15          ;DONE?
4386
4387                          ;WRITE AND READ "MARCHING ONES."
4388                          ;STARTING AT THE LOWEST ADDRESS, EACH LOCATION IS READ AND CHECKED
4389                          ;IF IT WAS 0, THEN IT IS WRITTEN WITH ALL 1'S. THE TEST IS REPEATED
4390                          ;INCREMENTALLY FOR ALL ADDRESSES.
4391
4392 017112 012704 002000      MOV      #1024.,R4      ;INITIALIZE COUNT
4393 017116 013701 003050      MOV      TMP1,R1
4394 017122 012737 017130 001110  MOV      #25,$LPERR    ;LOOP ON ERROR
4395 017130 010177 163702      MOV      R1,@WCSAR    ;ADDRESS THE WCS LOCATION
4396 017134 017702 163700      MOV      @WCSDR,R2
4397 017140 017705 163670      MOV      @WCSST,R5
4398 017144 030005              BIT      R0,R5          ;PAR BIT CLEAR?
4399 017146 001410              BEQ      3$            ;YES
4400 017150 005237 003052      INC      COUNT        ;NO MORE THAN 10 ERRORS
4401 017154 100005              BPL      3$
4402 017156 010137 001162      MOV      R1,$REG0     ;SAVE ADDRESS THAT WAS READ
4403 017162 010537 001164      MOV      R5,$REG1
4404 017166 000451              BR       7$            ;REPORT ERROR
4405 017170 010377 163644      MOV      R3,@WCSDR    ;WRITE A PATTERN OF ALL 1'S IN THIS
4406                          ;LOCATION
4407 017174 005201              INC      R1            ;POINT TO THE NEXT WCS LOC.
4408 017176 077424              SOB      R4,25        ;DONE?
4409
4410                          ;AT THIS POINT, THE PARITY CHIP SHOULD CONTAIN
4411                          ;ALL 1'S. THE FOLLOWING IS A "MARCHING ZERO" TEST. STARTING
4412                          ;AT THE HIGHEST ADDRESS, EACH LOCATION IS READ AND CHECKED
4413                          ;IF IT WAS ALL 1'S, THEN IT IS WRITTEN TO BE 0. THE SAME
4414                          ;TEST IS REPEATED (DECREMENTALLY) FOR ALL LOCATIONS.
4415
4416 017200 005301              DEC      R1
4417 017202 012704 002000      MOV      #1024.,R4    ;INITIALIZE COUNT
4418 017206 012737 017214 001110  MOV      #45,$LPERR   ;LOOP ON ERROR
4419 017214 010177 163616      MOV      R1,@WCSAR    ;ADDRESS THE WCS LOC.
4420 017220 017702 163614      MOV      @WCSDR,R2    ;READ THAT LOC.
4421 017224 017705 163604      MOV      @WCSST,R5
4422 017230 030005              BIT      R0,R5        ;PAR BIT SET?
4423 017232 001010              BNE      5$            ;YES
4424 017234 005237 003054      INC      COUNT0
4425 017240 100054              BPL      TST35        ;EXIT
4426 017242 010137 001162      MOV      R1,$REG0     ;SAVE WCS ADDRESS
4427 017246 010537 001164      MOV      R5,$REG1
4428 017252 000433              BR       8$            ;REPORT ERROR
4429 017254 005077 163560      CLR      @WCSDR        ;CLEAR THAT WCS LOCATION
4430 017260 005301              DEC      R1            ;POINT TO NEXT ADDRESS
4431 017262 077424              SOB      R4,45        ;DONE?
4432
4433 017264 062737 002000 003050  ADD      #2000,TMP1
4434 017272 006300              ASL      R0            ;PAR BIT MASK FOR NEXT PAR CHIP
4435 017274 005237 003046      INC      TMP0
4436 017300 023727 003046 000003  CMP      TMP0,#3      ;DONE?

```



```
4437 017306 001267      BNE      65
4438 017310 000430      BR       TST35      ;;EXIT
4439
4440
4441 017312 013702 003046 75:      MOV      TMPO,R2
4442 017316 006302          ASL      R2          ;FORM OFFSET
4443 017320 006302          ASL      R2
4444 017322 000162 017326 95:      JMP      95(R2)     ;GO TO THE ERROR CALL
4445 017326 104071          ERROR    71        ;CORRECT PARITY BIT WAS NOT READ BACK
4446                                     ;FROM PAR CHIP 0 (PAR0). PAR0 (BIT
4447                                     ;1 IN STATUS REG) SHOULD BE A 0.
4448 017330 000717          BR       35
4449 017332 104072          ERROR    72        ;CORRECT PARITY BIT WAS NOT READ BACK
4450                                     ;FROM PAR CHIP 1 (PAR1). PAR1 (BIT
4451                                     ;2 IN STATUS REG) SHOULD BE A 0.
4452 017334 000715          BR       35
4453 017336 104073          ERROR    73        ;CORRECT PARITY BIT WAS NOT READ BACK
4454                                     ;FROM PAR CHIP 2 (PAR2). PAR2 (BIT
4455                                     ;3 IN STATUS REG) SHOULD BE A 0.
4456 017340 000713          BR       35
4457
4458 017342 013702 003046 85:      MOV      TMPO,R2
4459 017346 006302          ASL      R2
4460 017350 006302          ASL      R2
4461 017352 000162 017356 105:     JMP      105(R2)    ;CORRECT PARITY BIT WAS NOT READ BACK
4462 017356 104071          ERROR    71        ;FROM PAR CHIP 0 (PAR0). PAR0 (BIT
4463                                     ;1 IN STATUS REG) SHOULD BE A 1.
4464
4465 017360 000735          BR       55
4466 017362 104072          ERROR    72        ;CORRECT PARITY BIT WAS NOT READ BACK
4467                                     ;FROM PAR CHIP 1 (PAR1). PAR1 (BIT
4468                                     ;2 IN STATUS REG) SHOULD BE A 1.
4469 017364 000733          BR       55
4470 017366 104073          ERROR    73        ;CORRECT PARITY BIT WAS NOT READ BACK
4471                                     ;FROM PAR CHIP 2 (PAR2). PAR2 (BIT
4472                                     ;3 IN STATUS REG) SHOULD BE A 1.
4473 017370 000731          BR       55
4474 ;;*****
4475 ;*TEST 35      TEST PARITY BIT STORAGE CHIPS, MARCHING 0'S AND 1'S
4476 ;THIS IS A TEST OF THE WCS PARITY BIT STORAGE CHIP (1 X 1K); AND
4477 ;IS KNOWN AS THE "MARCHING ONES AND ZEROES" TEST. THE MEMORY
4478 ;IS FIRST WRITTEN TO THE ALL ONES STATE. THEN SEQUENTIALLY,
4479 ;STARTING AT THE FIRST ADDRESS, THE ONE IS READ AND A
4480 ;ZERO IS WRITTEN. THIS SEQUENCE IS CONTINUED TO THE
4481 ;LAST LOCATION. AT THIS STAGE, THE DATA ARRAY IS FULL OF 0'S.
4482 ;THEN STARTING AT THE HIGHEST LOCATION, A ZERO IS READ AND
4483 ;A ONE IS WRITTEN. THE ADDRESS IS DECREMENTED AND THE
4484 ;SEQUENCE IS REPEATED UNTILL THE LOWEST LOCATION IS REACHED.
4485 ;THIS TEST WILL DETECT FAULTS IN THE INDIVIDUAL MEMORY CHIP
4486 ;ADDRESSING LINES AS WELL AS DATA PROBLEMS. GENERALLY, ADDRESSING
4487 ;FAULTS WILL SHOW UP IN REPEATED ERRORS WITH A DEFINITE FAILURE
4488 ;PATTERN.
4489 ;THE TEST IS REPEATED FOR THE THREE PARITY BIT STORAGE CHIPS:
4490 ;CHIP 0, SFGMENT 0, PAR 0, ADDRESSES 0,1,2,.....
4491 ;CHIP 1, SEGMENT 1, PAR 1, ADDRESSES 2000,2001,.....
4492 ;CHIP 2, SEGMENT 2, PAR 2, ADDRESSES 4000,4001,.....
```

```

4493 ; ;*****
4494 017372 000004 TST35: SCOPE
4495 017374 012737 017750 003056 MOV #TST36,NXTST ;STARTING ADDRESS OF NEXT TEST
4496 017402 012777 000020 163424 MOV #PARDIS,@WCSST ;DISABLE PARITY ERROR TRAPS
4497 017410 012737 177766 003052 MOV #-10.,COUNT ;DONT REPORT MORE THAN
4498 017416 012737 177766 003054 MOV #-10.,COUNTO ;10 ERRORS
4499 017424 012703 077777 MOV #77777,R3 ;INITIALIZE DATA PATTERN
4500 017430 005037 003046 CLR TMP0 ;SEGMENT NUMBER
4501 017434 012700 000002 MOV #PARO,RO ;PAR BIT MASK
4502 017440 005037 003050 CLR TMP1
4503
4504 ;FILL THE PARITY CHIP WITH 1'S
4505
4506 017444 013701 003050 65: MOV TMP1,R1
4507 017450 012704 002000 MOV #1024.,R4
4508 017454 010177 163356 15: MOV R1,@WCSAR
4509 017460 010377 163354 MOV R3,@WCSDR ;WRITE A PATTERN (177777)
4510 017464 005201 INC R1 ;NEXT WCS ADDRESS
4511 017466 077406 SOB R4,15 ;DONE?
4512
4513 ;WRITE AND READ "MARCHING 0'S".
4514 ;STARTING AT THE LOWEST ADDRESS, EACH LOCATION IS READ AND
4515 ;CHECKED IF IT WAS ALL 1'S, THEN IT IS WRITTEN WITH A 0.
4516 ;THE TEST IS REPEATED INCREMENTALLY FOR ALL ADDRESSES.
4517
4518 017470 012704 002000 MOV #1024.,R4 ;INITAILIZE COUNT
4519 017474 013701 003050 MOV TMP1,R1
4520 017500 012737 017506 001110 MOV #25,$LPERR ;LOOP ON ERROR
4521 017506 010177 163324 25: MOV R1,@WCSAR ;ADDRESS THE WCS LOCATION
4522 017512 017702 163322 MOV @WCSDR,R2
4523 017516 017705 163312 MOV @WCSST,R5 ;READ PARITY BIT
4524 017522 030005 BIT RO,R5 ;PAR BIT SET?
4525 017524 001010 BNE 35 ;YES
4526 017526 005237 003052 INC COUNT
4527 017532 100005 BPL 35
4528 017534 010137 001162 MOV R1,$REGO ;SAVE ADDRESS THAT WAS READ
4529 017540 010537 001164 MOV R5,$REG1
4530 017544 000451 BR 75 ;REPORT ERROR
4531 017546 005077 163266 35: CLR @WCSDR ;WRITE A PATTERN (0) IN THIS LOCATION
4532 017552 005201 INC R1 ;POINT TO THE NEXT WCS LOC.
4533 017554 077424 SOB R4,25 ;DONE?
4534
4535 ;AT THIS POINT, THE WCS PARITY CHIP (1X1K) SHOULD BE
4536 ;CLEARED. THE FOLLOWING IS A "MARCHING ONE" TEST. STARTING
4537 ;AT THE HIGHEST ADDRESS, EACH LOCATION IS READ AND CHECKED
4538 ;IF IT WAS ALL 0'S, THEN IT IS WRITTEN TO BE ALL 1'S.
4539 ;THE SAME TEST IS REPEATED (DECREMENTALLY FOR ALL
4540 ;LOCATIONS.
4541
4542 017556 005301 DEC R1
4543 017560 012704 002000 MOV #1024.,R4 ;INITIALIZE COUNT
4544 017564 012737 017572 001110 MOV #45,$LPERR ;LOOP ON ERROR
4545 017572 010177 163240 45: MOV R1,@WCSAR ;ADDRESS THE WCS LOC.
4546 017576 017702 163236 MOV @WCSDR,R2 ;READ THAT LOC.
4547 017602 017705 163226 MOV @WCSST,R5 ;READ PAR BIT
4548 017606 030005 BIT RO,R5 ;PAR BIT CLEAR?

```



```

4549 017610 001410      BEQ      55      ; YES
4550 017612 005237 003054  INC      COUNT0
4551 017616 100054      BPL      TST36      ;; EXIT
4552 017620 010137 001162  MOV     R1, $REG0    ; SAVE WCS ADDRESS
4553 017624 010537 001164  MOV     R5, $REG1
4554 017630 000433      BR       85      ; REPORT ERROR
4555 017632 010377 163202 55:     MOV     R3, @WCSDR
4556 017636 005301      DEC     R1      ; POINT TO NEXT ADDRESS
4557 017640 077424      SOB     R4, 45    ; DONE?
4558 017642 062737 002000 003050  ADD     #2000, TMP1
4559 017650 006300      ASL     R0      ; PAR BIT MASK FOR NEXT CHIP
4560 017652 005237 003046      INC     TMP0
4561 017656 023727 003046 000003  CMP     TMP0, #3    ; DONE?
4562 017664 001267      BNE     65
4563
4564 017666 000430      BR       TST36      ;; EXIT
4565
4566 017670 013702 003046 75:     MOV     TMP0, R2
4567 017674 006302      ASL     R2      ; FORM OFFSET
4568 017676 006302      ASL     R2
4569 017700 000162 017704 95:     JMP     95(R2) ; GO TO THE ERROR CALL
4570 017704 104071      ERROR   71      ; CORRECT PARITY BIT WAS NOT READ BACK
4571                                     ; FROM PAR CHIP 0 (PAR0). PAR0 (BIT
4572                                     ; 1 IN STATUS REG) SHOULD BE A 1.
4573 017706 000717      BR       35
4574 017710 104072      ERROR   72      ; CORRECT PARITY BIT WAS NOT READ BACK
4575                                     ; FROM PAR CHIP 1 (PAR1). PAR1 (BIT
4576                                     ; 2 IN STATUS REG) SHOULD BE A 1.
4577 017712 000715      BR       35
4578 017714 104073      ERROR   73      ; CORRECT PARITY BIT WAS NOT READ BACK
4579                                     ; FROM PAR CHIP 2 (PAR2). PAR2 (BIT
4580                                     ; 3 IN STATUS REG) SHOULD BE A 1.
4581 017716 000713      BR       35
4582
4583 017720 013702 003046 85:     MOV     TMP0, R2
4584 017724 006302      ASL     R2
4585 017726 006302      ASL     R2
4586 017730 000162 017734 105:    JMP     105(R2)
4587 017734 104071      ERROR   71      ; CORRECT PARITY BIT WAS NOT READ BACK
4588                                     ; FROM PAR CHIP 0 (PAR0). PAR0 (BIT
4589                                     ; 1 IN STATUS REG) SHOULD BE A 0.
4590 017736 000735      BR       55
4591 017740 104072      ERROR   72      ; CORRECT PARITY BIT WAS NOT READ BACK
4592                                     ; FROM PAR CHIP 1 (PAR1). PAR1 (BIT
4593                                     ; 2 IN STATUS REG) SHOULD BE A 0.
4594 017742 000733      BR       55
4595 017744 104073      ERROR   73      ; CORRECT PARITY BIT WAS NOT READ BACK
4596                                     ; FROM PAR CHIP 2 (PAR2). PAR2 (BIT
4597                                     ; 3 IN STATUS REG) SHOULD BE A 0.
4598 017746 000731      BR       55
4599
4600 *****
4601 *TEST 36 CHECK WCS WATCH-DOG TIMER
4602 ; THIS TEST CHECKS THAT THE WATCH-DOG-TIMER ON THE WCS
4603 ; MODULE IS OPERATIONAL AND PROVIDES TIME-OUT CAPABILITY
4604 ; WITHIN A CERTAIN TIME-RANGE. THE ACCEPTED RANGE OF THE TIMER
; IS 60-160 US. AN ERROR IS REPORTED IF IT IS NOT WITHIN

```

```
4605 ; THIS RANGE. IF THE TIMER IS NON-OPERATIONAL THE RESULTS OF
4606 ; SUBSEQUENT TESTS WHICH USE THE TIMER MAY NOT BE CORRECT.
4607 ; *****
4608 017750 000004 TST36: SCOPE
4609 017752 012737 020052 003056 MOV #TST37,NXTST ; STARTING ADDRESS OF NEXT TEST
4610
4611 017760 012702 177160 MOV #-400,R2 ; COUNT
4612 017764 012737 020030 000114 MOV #35,@#114 ; TIMEOUT VECTOR FOR WCS WATCH DOG
4613 017772 012777 001000 163034 MOV #MAINT,@WCSST ; ENABLE TIMER
4614
4615 020000 005777 163034 TST @WCSDR ; KICK OFF THE THE WCS TIMER
4616 020004 005202 15: INC R2 ; KEEP TIME
4617 020006 001376 BNE 15 ; WAITED ENOUGH?
4618 ; YES, THE WCS WATCH DOG DID
4619 ; NOT TIME OUT-REPORT ERROR
4620 020010 042777 001000 163016 BIC #MAINT,@WCSST ; SHUT OFF TIMER
4621 020016 012737 003344 000114 MOV #BADPAR,@#114
4622 020024 104064 ERROR 64 ; WCS-WATCH-DOG TIMER DID NOT
4623 ; TIME OUT AS REQUIRED. THE
4624 ; PROGRAM WAITED FOR 2-56 SECTENDS
4625 ; BEFORE GIVING THIS ERROR,
4626 ; NORMALLY THE WCS-WATCH-DOG
4627 ; SHOULD TIME OUT BETWEEN
4628 ; 70-150 M. S.
4629 020026 000411 BR TST37 ; EXIT
4630 ; ENTER HERE WHEN WCS WATCH DOG TIMEP-OUT
4631
4632
4633 020030 022626 35: CMP (SP)+,(SP)+ ; RESTORE STACK
4634 020032 012777 000020 162774 MOV #PARDIS,@WCSST ; CLEAR ERROR BIT 15
4635 020040 005077 162770 CLR @WCSST
4636 020044 012737 003344 000114 MOV #BADPAR,@#114
4637
4638
4639 ; *****
4640 ; *TEST 37 CHECK THE WCS INITIALIZATION BY BM MICRO-CODE
4641 ; *****
4642 020052 000004 TST37: SCOPE
4643 020054 012737 020250 003056 MOV #TST40,NXTST ; STARTING ADDRESS OF NEXT TEST
4644 020062 005002 CLR R2 ; ERROP COUNT
4645
4646 ; WRITE WRONG PARITY IN THE ENTIRE WCS APPRAY
4647
4648 020064 012777 000060 162742 MOV #WWP+PARDIS,@WCSST ; DISABLE PARITY TRAPS. WWP
4649 020072 005001 CLR R1
4650 020074 010177 162736 15: MOV R1,@WCSAR ; WRITE WRONG PARITY INTO
4651 020100 005077 162734 CLR @WCSDR ; THIS WCS LOC
4652 020104 005201 INC R1 ; NEXT LOC
4653 020106 020127 006000 CMP R1,#6000 ; DONE?
4654 020112 001370 BNE 15
4655 020114 042777 000040 162712 BIC #WWP,@WCSST ; CLEAR WWP
4656
4657 ; INITIALIZE THE WCS DATA ARRAY BY SETTING BITS 15 AND 10 IN
4658 ; THE INIT REGISTER IN THE BASE MACHINE (BM). INITIALIZATION CLEANS
4659 ; UP BAD PARITY, IF PRESENT, IN THE WCS. NOTE, SIMILIAR INIT-
4660 ; -IALIZATION IS DONE BY BASE MACHINE ON POWER-UP IF WCS IS PRESENT.
```



```

4661
4662 020122 012700 102000      MOV      #BIT15+BIT10,R0 ;SET THESE BITS IN "INIT" REGISTER
4663 020126 076600              MED
4664 020130 000352              352          ;MED CODE FOR "INIT"
4665
4666                          ;READ THE WCS DATA ARRAY. THE INITIALIZATION MICRO-CODE SHOULD HAVE
4667                          ;CLEANED UP THE WCS ARRAY, HENCE WCS PARITY ERROR SHOULD NOT OCCUR.
4668
4669 020132 012737 020170 000114  MOV      #4$,@#114      ;PARITY TRAP VECTOR
4670 020140 005077 162670              CLR      @WCSST        ;CLEAR UP PARDIS
4671 020144 005000              CLR      R0
4672
4673 020146 010077 162664          2$:      MOV      R0,@WCSAR      ;READ THE WCS
4674 020152 017701 162662          MOV      @WCSDR,R1      ;ARRAY
4675 020156 005200          3$:      INC      R0              ;NEXT LOC
4676 020160 020027 006000          CMP      R0,#6000       ;DONE?
4677 020164 001370          BNE      2$
4678 020166 000425          BR       6$
4679
4680                          ;ENTER HERE IF A PARITY ERROR OCCURS ON READING WCS
4681
4682 020170 005777 162640          4$:      TST      @WCSST        ;PARITY ERROR FROM WCS?
4683 020174 100020          BPL      5$              ;NO
4684 020176 012777 000020 162630  MOV      #PARDIS,@WCSST ;CLEAR ERROR
4685 020204 005077 162624          CLR      @WCSST        ;CLEAR PARDIS
4686 020210 022626          CMP      (SP)+,(SP)+    ;POP THE STACK
4687 020212 017737 162616 001162  MOV      @WCSST,$REG0   ;SAVE STATUS
4688 020220 020227 000012          CMP      R2,#10        ;DONT REPORT MORE THAN 10 ERRORS
4689 020224 101006          BHI      6$
4690
4691 020226 010037 001164          MOV      R0,$REG1      ;SAVE WCS ADDRESS WHICH GAVE PARITY ERROR
4692 020232 104067          ERROR  67              ;PARITY ERROR TRAP OCCURED ON
4693                          ;READING WCS LOC (GIVEN IN ERROR MESSAGE)
4694                          ;THIS INDICATES THAT THE BASE MACHINE
4695                          ;INITIALIZATION MICROCODE DID NOT CLEAN
4696                          ;UP THE BAD PARITY IN THE WCS. (BAD PARITY
4697                          ;WAS WRITTEN, PREVIOUSLY, IN THE WCS).
4698
4699 020234 000750          BR       3$
4700
4701                          ;ENTER HERE IF A PARITY ERROR OCCURED AND IT WAS NOT FROM THE WCS.
4702
4703 020236 000137 003344          5$:      JMP      BADPAR        ;GO TO COMMON PARITY ERROR HANDLER
4704
4705 020242 012737 003344 000114  6$:      MOV      #BADPAR,@#114 ;REESTABLISH PARITY TRAP VECTOR

```

.SBTTL ---TEST OF WCS XFC OPCODE DISPATCH---

4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751  
4752  
4753  
4754  
4755  
4756  
4757  
4758  
4759  
4760  
4761

```

;*****
;*TEST 40      CHECK "OTHER USER" DISPATCH TO 6001, XFC=076000-076077
;THIS TEST CHECKS THE "OTHER USER" DISPATCH OF THE XFC
;INSTRUCTION TO 6001.  THE XFC OPCODE GROUP CHECKED
;IN THIS TEST IS 076000-076077.  WHEN AN XFC
;OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE
;SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY
;POINT IN THE WCS IS 6001 (MICRO-ADDRESS IN
;PAGE6).
;THE TEST SEQUENCE IS AS FOLLOWING:

;MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH
;AT 6001.  THE LOADED MICRO CODE FUNCTIONS IS TO READ
;CSP MD AND WRITE IT IN R3 (BOTH A&B SIDE).

;AN XFC OP-CODE IS EXECUTED.
;THE CONTROL SHOULD TRANSFER TO 6001 IN WCS AND
;MICRO-CODE GETS EXECUTED.  RETURN IS MADE BACK TO
;BASE MACHINE.

;UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF
;THE CORRECT XFC-OPCODE WAS SAVED IN R3
;(FROM CSP MD ) WHILE RUNNING FROM WCS.

;IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTED

;THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC
;OP-CODES IN THE GROUP.
;THIS TEST USES THE WCS WATCH-DOG-TIMER.  IF, AS A RESULT
;OF SOME FAULTY CONDITION, THE CONTROL IS NOT
;TRANSFERRED BACK TO THE BASE MACHINE THE WATCH-DOG
;TIMES OUT AND A TRAP TO 114 OCCURS.

;FAILURE OF THIS TEST COULD INDICATE A FAULT IN:
;   BASE MACHINE XFC DECODE AND DISPATCH.
;   TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT
;   CONTROL-PATH USED BY THE MICRO-CODE EXECUTING FROM
;   THE WCS. (NUA<9: 0>, ARRAY ADDRESS MUX-APORT,
;   BUS-U MUX-BPORT, BUS-U LINES.

```

```

;*****
TST40:  SCOPE
        MOV      #TST41,NXTST  ;STARTING ADDRESS OF NEXT TEST
        MOV      #100,R0      ;LOAD COUNT
        MOV      #76000,R1    ;INITIALIZE XFC OPCODE
        JSR      R5,LDWCS     ;LOAD WCS STARTING AT THIS
        6151                ;WCS ADDRESS (IN CONTROL SPACE)
        MXFC1              ;FROM THIS MEMORY LOCATION
        OTHDIS             ;ENTRY POINT INTO WCS

```

```

020250 000004
020252 012737 020420 003056
020260 012700 000100
020264 012701 076000
020270 004537 037472
020274 006151
020276 061760
020300 006001

```



```

4762 020302 010137 020326          15:  MOV      R1,25          ;SET UP XFC CODE FOR EXECUTION
4763 020306 012737 020370 000114    MOV      #45,@#114      ;SET UP SERVICE VECTOR FOR WCS
4764                                     ;WATCH DOG TIMEOUT.
4765 020314 012777 001000 162512    MOV      #MAINT,@WCSST  ;SET MAINTENANCE BIT
4766 020322 005777 162512          TST      @WCSDR         ;KICK OFF THE WATCH-DOG TIMER
4767
4768 020326 000000          25:  .WORD    0            ;THIS WORD WILL CONTAIN THE XFC OPCODE
4769                                     ;TO BE EXECUTED. AT THIS POINT CONTROL
4770                                     ;IS TRANSFERRED TO THE WCS.
4771
4772                                     ;RETURN FROM WCS OCCURS HERE
4773 020330 042777 001000 162476    BIC      #MAINT,@WCSST  ;SHUT OFF THE TIMER
4774 020336 012737 003344 000114    MOV      #BADPAR,@#114
4775 020344 020301          CMP      R3,R1          ;WAS CORRECT OPCODE SAVED IN R4,
4776                                     ;WHILE IN WCS?
4777 020346 001405          BEQ      35            ;YES
4778 020350 010137 001162          MOV      R1,$REGO      ;EXPCED XFC CODE
4779 020354 010337 001164          MOV      R3,$REG1      ;CODE RECVD
4780 020360 104030          ERROR   30            ;CORRECT XFC OPCODE WAS NOT SAVED IN R3
4781                                     ;BY THE MICRO-CODE WHILE INSIDE WCS.
4782                                     ;NOTE, THAT XFC OPCODE SHOULD HAVE
4783                                     ;TRANSFERRED CONTROL TO THE WCS.
4784
4785 020362 005201          35:  INC      R1            ;NEXT XFC OPCODE
4786 020364 077032          SOB     RO,15         ;DONE?
4787 020366 000414          BR      TST41         ;;EXIT
4788
4789                                     ;ENTER HERE IF THE WATCH-DOG TIMER ON THE WCS TIMED OUT BEFORE
4790                                     ;AN ORDERLY EXIT AND RETURN OF CONTROL COULD BE DONE (FROM THE
4791                                     ;WCS) TO THE BASE MACHINE.
4792
4793 020370 022626          45:  CMP      (SP)+,(SP)+  ;POP THE STACK
4794
4795 020372 012737 003344 000114    MOV      #BADPAR,@#114
4796 020400 012777 000020 162426    MOV      #PARDIS,@WCSST
4797 020406 005077 162422          CLR      @WCSST
4798 020412 010137 001162          MOV      R1,$REGO      ;XFC OP-CODE
4799 020416 104031          ERROR   31            ;WCS WATCH DOG TIMER TIMED OUT
4800                                     ;UPON TRANSFERRING CONTROL FROM THE
4801                                     ;BASE MACHINE TO THE WCS MICO-CODE.
4802                                     ;NORMALLY, AFTER UCODE EXECUTION
4803                                     ;CONTROL SHOULD HAVE RETURNED TO
4804                                     ;BASE MACHINE.
4805
4806
4807
4808
4809
4810
4811                                     ;*****
4812                                     ;*TEST 41 CHECK "OTHER USER" DISPATCH TO 6011, XFC=076100-076177
4813                                     ;THIS TEST CHECKS THE "OTHER USER" DISPATCH OF THE XFC
4814                                     ;INSTRUCTION TO 6011. THE XFC OPCODE GROUP CHECKED
4815                                     ;IN THIS TEST IS 076100-076177. WHEN AN XFC
4816                                     ;OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE
4817                                     ;SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY

```

```

4818 ;POINT IN THE WCS IN 6011 (MICRO-ADDRESS IN
4819 ;PAGE 6).
4820 ;THE TEST SEQUENCE IS AS FOLLOWING:
4821
4822 ;MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH
4823 ;AT 6011. THE LOADED MICRO CODES FUNCTIONS IS TO READ
4824 ;CSP MD AND WRITE IT IN R3(BOTH A&B SIDE).
4825
4826 ;AN XFC OP-CODE IS EXECUTED.
4827 ;THE CONTROL SHOULD TRANSFER TO 6011 IN WCS AND
4828 ;MICRO-CODE GETS EXECUTED. RETURN IS MADE BACK TO
4829 ;BASE MACHINE.
4830
4831 ;UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF
4832 ;THE CORRECT XFC-OPCODE WAS SAVED IN R3
4833 ;(FROM CSP MD WHILE RUNNING FROM WCS.
4834
4835 ; IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTD.
4836
4837 ;THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC
4838 ;OP-CODES IN THE GROUP.
4839 ;THIS TEST USES THE WCS WATCH-DOG-TIMES. IF, AS A RESULT
4840 ;OF SOME FAULTY CONDITION, THE CONTROL IS NOT
4841 ;TRANSFERRED BACK TO THE BASE MACHINE THE WATCH-DOG
4842 ;TIMES OUT AND A TRAP TO 114 OCCURS.
4843
4844 ;FAILURE OF THIS TEST COULD INDICATE A FAULT IN:
4845 ;     BASE MACHINE XFC DECODE AND DISPATCH
4846 ;     TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT.
4847 ;     CONTROL-PATH USED BY THE MIRO-CODE EXECUTING FROM
4848 ;     THE WCS. (NUA<9: 0>),ARRAY ADDRESS MUX-APORT,
4849 ;     BUS-U MUS-BPORT, BUS-U LINES.
4850 ;*****
4851 020420 000004          TST41: SCOPE
4852 020422 012737 020570 003056  MOV     #TST42,NXTST ;STARTING ADDRESS OF NEXT TEST
4853 020430 012700 000100          MOV     #100,R0      ;LOAD COUNT
4854 020434 012701 076100          MOV     #76100,R1   ;INITIALIZE XFC OPCODE
4855 020440 004537 037472          JSR     R5,LDWCS    ;LOAD WCS WITH MICRO-CODE
4856 020444 006351          6351      ;MAIN-CONTROL-BLOCK LOADED HERE
4857 020446 062012          MXFC11 ;FROM THIS MEMORY LOCATION
4858 020450 006011          OTH11   ;ENTRY POINT UNITS WCS
4859
4860 020452 010137 020476          15:    MOV     R1,25       ;SET UP XFC CODE FOR EXECUTION
4861 020456 012737 020540 000114  MOV     #45,@#114  ;SET UP SERVICE VECTOR FOR WCS
4862                                     ;WATCH DOG TIMEOUT.
4863 020464 012777 001000 162342  MOV     #MAINT,@WCSST ;SET MAINTENANCE BIT
4864 020472 005777 162342          TST     @WCSDR     ;KICK OFF THE WATCH-DOG TIMER
4865
4866 020476 000000          25:    .WORD 0        ;THIS WORD WILL CONTAIN THE XFC OPCODE
4867                                     ;TO BE EXECUTED. AT THIS POINT CONTROL
4868                                     ;IS TRANSFERRED TO THE WCS.
4869
4870                                     ;RETURN FROM WCS OCCURS HERE
4871 020500 042777 001000 162326  BIC     #MAINT,@WCSST ;SHUT OFF THE TIMER
4872 020506 012737 003344 000114  MOV     #BADPAR,@#114
4873 020514 020301          CMP     R3,R1     ;WAS CORRECT OPCODE SAVED IN R4.

```



```
4874                                     ;WHILE IN WCS?  
4875 020516 001405                       BEQ     3$      ;YES  
4876 020520 010137 001162                MOV     R1,$REG0 ;EXPCTD XFC CODE  
4877 020524 010337 001164                MOV     R3,$REG1 ;CODE RECVD  
4878 020530 104030                       ERROR    30      ;CORRECT XFC OPCODE WAS NOT SAVED IN R3  
4879                                     ;BY THE MICRO-CODE WHILE INSIDE WCS.  
4880                                     ;NOTE, THAT XFC OPCODE SHOULD HAVE  
4881                                     ;TRANSFERRED CONTROL TO THE WCS.  
4882  
4883 020532 005201                       3$:     INC     R1      ;NEXT XFC OPCODE  
4884 020534 077032                       SOB     R0,1$     ;DONE?  
4885 020536 000414                       BR      TST42     ;EXIT  
4886  
4887                                     ;ENTER HERE IF THE WATCH-DOG TIMER ON THE WCS TIMED OUT BEFORE  
4888                                     ;AN ORDERLY EXIT AND RETURN OF CONTROL COULD BE DONE (FROM THE  
4889                                     ;WCS) TO THE BASE MACHINE.  
4890  
4891 020540 022626                       4$:     CMP     (SP)+,(SP)+ ;POP THE STACK  
4892  
4893 020542 012737 003344 000114          MOV     #BADPAR,@#114  
4894 020550 012777 000020 162256          MOV     #FARDIS,@WCSST  
4895 020556 005077 162252  
4896 020562 010137 001162                MOV     R1,$REG0 ;XFC OPCODE  
4897 020566 104031                       ERROR    31      ;WCS WATCH DOG TIMER TIMED OUT  
4898                                     ;UPON TRANSFERRING CONTROL FROM THE  
4899                                     ;BASE MACHINE TO THE WCS MICO-CODE.  
4900                                     ;NORMALLY, AFTER UCODE EXECUTION  
4901                                     ;CONTROL SHOULD HAVE RETURNED TO  
4902                                     ;BASE MACHINE.  
4903  
4904  
4905  
4906  
4907  
4908  
4909  
4910                                     ;*****  
4911 *TEST 42 CHECK "OTHER USER" DISPATCH TO 6012, XFC=076200-076277  
4912                                     ;THIS TEST CHECKS THE "OTHER USER" DISPATCH OF THE XFC  
4913                                     ;INSTRUCTION TO 6012. THE XFC OPCODE GROUP CHECKED  
4914                                     ;IN THIS TEST IS 076200-076277. WHEN AN XFC  
4915                                     ;OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE  
4916                                     ;SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY  
4917                                     ;POINT IN THE WCS IS 6012. (MICRO-ADDRESS IN  
4918                                     ;PAGE 6.  
4919                                     ;THE TEST SEQUENCE IS AS FOLLOWING:  
4920  
4921                                     ;MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH  
4922                                     ;AT 6012. THE LOADED MICRO CODES FUNCTION IS TO READ  
4923                                     ;CSP MD AND WRITE IT IN R3(BOTH A&B SIDE).  
4924  
4925                                     ;AN XFC OP-CODE IS EXECUTED.  
4926                                     ;THE CONTROL SHOULD TRANSFER TO 6012 IN WCS AND  
4927                                     ;MICRO-CODE GETS EXECUTED. RETURN IS MADE BACK TO  
4928                                     ;BASE MACHINE.  
4929                                     ;UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF
```

```

4930 ;THE CORRECT XFC-OPCODE WAS SAVED IN R3
4931 ;(FROM CSP MD ) WHILE RUNNING FROM WCS.
4932
4933 ; IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTED.
4934
4935 ;THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC
4936 ;OP-CODES IN THE GROUP.
4937 ;THIS TEST USES THE WCS WATCH-DOG-TIMER. IF, AS A RESULT
4938 ;OF SOME FAULTY CONDITION, THE CONTROL IS NOT
4939 ;TRANSFERRED BACK TO THE BASE MACHINE THE WATCH-DOG
4940 ;TIMES OUT AND A TRAP TO 114 OCCURS.
4941
4942
4943
4944
4945
4946
4947
4948
4949

```

```

; FAILURE OF THIS TEST COULD INDICATE A FAULT IN:
;     BASE MACHINE XFC DECODE AND DISPATCH
;     TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT.
;     CONTROL-PATH USED BY THE MICRO-CODE EXECUTING FROM
;     THE WCS. (NUA<9:0>, ARRAY ADDRESS MUS-APORT,
;     BUS-U MUX-BPORT, BUS-U LINES.

```

\*\*\*\*\*

```

4950 020570 000004          TST42: SCOPE
4951 020572 012737 020740 003056  MOV #TST43,NXTST ; STARTING ADDRESS OF NEXT TEST
4952 020600 012700 000100  MOV #100,R0 ; LOAD COUNT
4953 020604 012701 076200  MOV #76200,R1 ; INITIALIZE XFC OPCODE
4954 020610 004537 037472  JSR R5,LDWCS ; LOAD WCS WITH MICRO-CODE
4955 020614 006361 6361 ; MAIN-CONTROL-BLOCK LOADED HERE
4956 020616 062044 MXFC12 ; FROM THIS MEMORY LOCATION
4957 020620 006012 OTH12 ; ENTRY POINT INTO WCS
4958
4959 020622 010137 020646 15: MOV R1,25 ; SET UP XFC CODE FOR EXECUTION
4960 020626 012737 020710 000114  MOV #45,@#114 ; SET UP SERVICE VECTOR FOR WCS
4961 ; WATCH DOG TIMEOUT.
4962 020634 012777 001000 162172  MOV #MAINT,@WCSST ; SET MAINTENANCE BIT
4963 020642 005777 162172  TST @WCSDR ; KICK OFF THE WATCH-DOG TIMER
4964
4965 020646 000000 25: .WORD 0 ; THIS WORD WILL CONTAIN THE XFC OPCODE
4966 ; TO BE EXECUTED. AT THIS POINT CONTROL
4967 ; IS TRANSFERRED TO THE WCS.
4968
4969 ; RETURN FROM WCS OCCURS HERE
4970 020650 042777 001000 162156  BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
4971 020656 012737 003344 000114  MOV #BADPAR,@#114
4972 020664 020301  CMP R3,R1 ; WAS CORRECT OPCODE SAVED IN R4,
4973 ; WHILE IN WCS?
4974 020666 001405  BEQ 35 ; YES
4975 020670 010137 001162  MOV R1,$REG0 ; EXPECTD XFC CODE
4976 020674 010337 001164  MOV R3,$REG1 ; CODE RECVD
4977 020700 104030  ERROR 30 ; **CORRECT XFC OPCODE WAS NOT SAVED
4978 ; BY THE MICRO-CODE WHILE INSIDE WCS.
4979 ; NOTE, THAT XFC OPCODE SHOULD HAVE
4980 ; TRANSFERRED CONTROL TO THE WCS.
4981
4982 020702 005201 35: INC R1 ; NEXT XFC OPCODE
4983 020704 077032  SOB R0,15 ; DONE?
4984 020706 000414  BR TST43 ; EXIT
4985

```



4986 ;ENTER HERE IF THE WATCH-DOG TIMER ON THE WCS TIMED OUT BEFORE  
4987 ;AN ORDERLY EXIT AND RETURN OF CONTROL COULD BE DONE (FROM THE  
4988 ;WCS) TO THE BASE MACHINE.  
4989  
4990 020710 022626 45: CMP (SP)+,(SP)+ ;POP THE STACK  
4991  
4992 020712 012737 003344 000114 MOV #BADPAR,@#114  
4993 020720 012777 000020 162106 MOV #PARDIS,@WCSST  
4994 020726 005077 162102 CLR @WCSST  
4995 020732 010137 001162 MOV R1,\$REGO  
4996 020736 104031 ERROR 31 ;\*\*WCS WATCH DOG TIMER TIMED OUT

4997  
4998  
4999  
5000  
5001  
5002 ;\*\*\*\*\*  
5003 ;\*TEST 43 CHECK "OTHER USER" DISPATCH TO 6013, XFC=076300-076377  
5004 ;THIS TEST CHECKS THE "OTHER USER" DISPATCH OF THE XFC  
5005 ;INSTRUCTION TO 6013. THE XFC OPCODE GROUP CHECKED  
5006 ;IN THIS TEST IS 076300-76377. WHEN AN XFC  
5007 ;OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE  
5008 ;SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY  
5009 ;POINT IN THE WCS IS 6013 (MICRO-ADDRESS IN  
5010 ;PAGE 6.  
5011 ;THE TEST SEQUENCE IS AS FOLLOWING:  
5012  
5013 ;MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH  
5014 ;AT 6013. THE LOADED MICRO CODES FUNCTION IS TO READ  
5015 ;CSP MD AND WRITE IT IN R3(BOTH A&B SIDE).  
5016  
5017 ;AN XFC OP-CODE IS EXECUTED.  
5018 ;THE CONTROL SHOULD TRANSFER TO 6013 IN WCS AND  
5019 ;MICRO-CODE GETS EXECUTED. RETURN IS MADE BACK TO  
5020 ;BASE MACHINE.  
5021  
5022 ;UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF  
5023 ;THE CORRECT XFC-OPCODE WAS SAVED IN R3  
5024 ;(FROM CSP MD WHILE RUNNING FROM WCS.  
5025  
5026 ;IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTED.  
5027  
5028 ;THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC  
5029 ;OP-CODE IN THE GROUP.  
5030  
5031 ;THIS TEST USES THE WCS WATCH-DOG-TIMES. IF, AS A RESULT  
5032 ;OF SOME FAULTY CONDITION, THE CONTROL IS NOT  
5033 ;TRANSFERRED BACK TO THE BASE MACHINE THE WATCH-DOG  
5034 ;TIMES OUT AND A TRAP TO 114 OCCURS  
5035  
5036 ;FAILURE OF THIS TEST COULD INDICATE A FAULT IN:  
5037 ; BASE MACHINE XFC DECODE AND DISPATCH  
5038 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT  
5039 ; CONTROL-PATH USED BY THE MICRO-CODE EXECUTING FROM  
5040 ; THE WCS. (NUA<9:0>), ARRAY ADDRESS MUX-APORT,  
5041 ; BUS-U MUX-BPORT, BUS-U LINES.

```
5042 ;*****  
5043 020740 000004 TST43: SCOPE  
5044 020742 012737 021110 003056 MOV #TST44,NXTST ;STARTING ADDRESS OF NEXT TEST  
5045 020750 012700 000100 MOV #100,R0 ;LOAD COUNT  
5046 020754 012701 076300 MOV #76300,R1 ;INITIALIZE XFC OPCODE  
5047 020760 004537 037472 JSR R5,LDWCS ;LOAD WCS WITH MICRO-CODE  
5048 020764 006371 6371 ;MAIN CONTROL BLOCK LOADED HERE  
5049 020766 062076 MXFC13 ;FROM THIS MEMORY ADDRESS  
5050 020770 006013 OTH13 ;ENTRY POINT INTO WCS  
5051  
5052 020772 010137 021016 15: MOV R1,25 ;SET UP XFC CODE FOR EXECUTION  
5053 020776 012737 021060 000114 MOV #45,@#114 ;SET UP SERVICE VECTOR FOR WCS  
5054 ;WATCH DOG TIMEOUT.  
5055 021004 012777 001000 162022 MOV #MAINT,@WCSST ;SET MAINTENANCE BIT  
5056 021012 005777 162022 TST @WCSDR ;KICK OFF THE WATCH-DOG TIMER  
5057  
5058 021016 000000 25: .WORD 0 ;THIS WORD WILL CONTAIN THE XFC OPCODE  
5059 ;TO BE EXECUTED. AT THIS POINT CONTROL  
5060 ;IS TRANSFERRED TO THE WCS.  
5061  
5062 ;RETURN FROM WCS OCCURS HERE  
5063 021020 042777 001000 162006 BIC #MAINT,@WCSST ;SHUT OFF THE TIMER  
5064 021026 012737 003344 000114 MOV #BADPAR,@#114  
5065 021034 020301 CMP R3,R1 ;WAS CORRECT OPCODE SAVED IN R4,  
5066 ;WHILE IN WCS?  
5067 021036 001405 BEQ 35 ;YES  
5068 021040 010137 001162 MOV R1,$REGO ;EXPTD XFC CODE  
5069 021044 010337 001164 MOV R3,$REG1 ;CODE RECVD  
5070 021050 104030 ERROR 30 ;CORRECT XFC OPCODE WAS NOT SAVED IN R3  
5071 ;BY THE MICRO-CODE WHILE INSIDE WCS.  
5072 ;NOTE, THAT XFC OPCODE SHOULD HAVE  
5073 ;TRANSFERRED CONTROL TO THE WCS.  
5074  
5075 021052 005201 35: INC R1 ;NEXT XFC OPCODE  
5076 021054 077032 SOB R0,15 ;DONE?  
5077 021056 000414 BR TST44 ;EXIT  
5078  
5079 ;ENTER HERE IF THE WATCH-DOG TIMER ON THE WCS TIMED OUT BEFORE  
5080 ;AN ORDERLY EXIT AND RETURN OF CONTROL COULD BE DONE (FROM THE  
5081 ;WCS) TO THE BASE MACHINE.  
5082  
5083 021060 022626 45: CMP (SP)+,(SP)+ ;POP THE STACK  
5084  
5085 021062 012737 003344 000114 MOV #BADPAR,@#114  
5086 021070 012777 000020 161736 MOV #PARDIS,@WCSST  
5087 021076 005077 161732 CLR @WCSST  
5088 021102 010137 001162 MOV R1,$REGO  
5089 021106 104031 ERROR 31 ;**WCS WATCH DOG TIMER TIMED OUT  
5090  
5091  
5092  
5093  
5094  
5095 ;*****  
5096 ;*TEST 44 CHECK "OTHER USER" DISPATCH TO 6014, XFC=076400-076477  
5097 ;THIS TEST CHECKS THE "OTHER USER" DISPATCH OF THE XFC
```



5098  
5099  
5100  
5101  
5102  
5103  
5104  
5105  
5106  
5107  
5108  
5109  
5110  
5111  
5112  
5113  
5114  
5115  
5116  
5117  
5118  
5119  
5120  
5121  
5122  
5123  
5124  
5125  
5126  
5127  
5128  
5129  
5130  
5131  
5132  
5133  
5134  
5135  
5136  
5137  
5138  
5139  
5140  
5141  
5142  
5143  
5144  
5145  
5146  
5147  
5148  
5149  
5150  
5151  
5152  
5153

; INSTRUCTION TO 6014. THE XFC OPCODE GROUP CHECKED  
; IN THIS TEST IS 076400-076477. WHEN AN XFC  
; OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE  
; SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY  
; POINT IN THE WCS IS 6014 (MICRO-ADDRESS IN  
; PAGE 6.  
; THE TEST SEQUENCE IS AS FOLLOWING:  
  
; MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH  
; AT 6014. THE LOADED MICRO CODES FUNCTION IS TO READ  
; CSP MD AND WRITE IT IN R3(BOTH A&B SIDE).  
  
; AN XFC OP-CODE IS EXECUTED.  
; THE CONTROL SHOULD TRANSFER TO 6014 IN WCS AND  
; MICRO-CODE GETS EXECUTED. RETURN IS MADE BACK TO  
; BASE MACHINE.  
  
; UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF  
; THE CORRECT XFC-OPCODE WAS SAVED IN R3  
; (FROM CSP MD WHILE RUNNING FROM WCS.  
  
; IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTED.  
  
; THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC  
; OP-CODES IN THE GROUP.  
  
; THIS TEST USES THE WCS WATCH-DOG-TIMES. IF, AS A RESULT  
; OF SOME FAULTY CONDITION, THE CONTROL IS NOT  
; TRANSFERRED BACK TO THE BASE MACHINE THE WATCH-DOG  
; TIMES OUT AND A TRAP TO 114 OCCURS  
  
; FAILURE OF THIS TEST COULD INDICATE A FAULT IN:  
; BASE MACHINE XFC DECODE AND DISPATCH  
; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT  
; CONTROL-PATH USED BY THE MICRO-CODE EXECUTING FROM  
; THE WCS. (NUA<9:0>, ARRAY ADDRESS MUX-APORT,  
; BUS-U MUX-BPORT, BUS-U LINES.

\*\*\*\*\*

TST44: SCOPE  
MOV #TST45,NXTST ; STARTING ADDRESS OF NEXT TEST  
MOV #100,R0 ; LOAD COUNT  
MOV #76400,R1 ; INITIALIZE XFC OPCODE  
JSR R5,LDWCS ; LOAD WCS WITH MICRO-CODE  
6401 ; MAIN-CONTROL-BLOCK LOADED HERE  
MXFC14 ; FROM THIS MEMORY ADRES  
OTH14 ; ENTRY POINT INTO WCS  
  
15: MOV R1,25 ; SET UP XFC CODE FOR EXECUTION  
MOV #45,@#114 ; SET UP SERVICE VECTOR FOR WCS  
; WATCH DOG TIMEOUT.  
  
MOV #MAINT,@WCSST ; SET MAINTENANCE BIT  
TST @WCSDR ; KICK OFF THE WATCH-DOG TIMER  
  
25: .WORD 0 ; THIS WORD WILL CONTAIN THE XFC OPCODE  
; TO BE EXECUTED. AT THIS POINT CONTROL  
; IS TRANSFERRED TO THE WCS.

021110 000004  
021112 012737 021260 003056  
021120 012700 000100  
021124 012701 076400  
021130 004537 037472  
021134 006401  
021136 062130  
021140 006014  
  
021142 010137 021166 15:  
021146 012737 021230 000114  
  
021154 012777 001000 161652  
021162 005777 161652  
  
021166 000300 25:

```

5154
5155
5156 021170 042777 001000 161636      BIC      #MAINT,@WCSST      ;RETURN FROM WCS OCCURS HERE
5157 021176 012737 003344 000114      MOV      #BADPAR,@#114    ;SHUT OFF THE TIMER
5158 021204 020301                      CMP      R3,R1            ;WAS CORRECT OPCODE SAVED IN R4,
5159                                     ;WHILE IN WCS?
5160 021206 001405                      BEQ      3$              ;YES
5161 021210 010137 001162      MOV      R1,$REG0        ;EXPCTD XFC CODE
5162 021214 010337 001164      MOV      R3,$REG1        ;CODE RECVD
5163 021220 104030                      ERROR   30              ;CORRECT XFC OPCODE WAS NOT SAVED
5164                                     ;BY THE MICRO-CODE WHILE INSIDE WCS.
5165                                     ;NOTE, THAT XFC OPCODE SHOULD HAVE
5166                                     ;TRANSFERRED CONTROL TO THE WCS.
5167
5168 021222 005201                      3$:    INC      R1            ;NEXT XFC OPCODE
5169 021224 077032                      SOB      R0,1$          ;DONE?
5170 021226 000414                      BR       TST45          ;EXIT
5171
5172                                     ;ENTER HERE IF THE WATCH-DOG TIMER ON THE WCS TIMED OUT BEFORE
5173                                     ;AN ORDERLY EXIT AND RETURN OF CONTROL COULD BE DONE (FROM THE
5174                                     ;WCS) TO THE BASE MACHINE.
5175
5176 021230 022626                      4$:    CMP      (SP)+,(SP)+ ;POP THE STACK
5177
5178 021232 012737 003344 000114      MOV      #BADPAR,@#114
5179 021240 012777 000020 161566      MOV      #PARDIS,@WCSST
5180 021246 005077 161562      CLR      @WCSST
5181 021252 010137 001162      MOV      R1,$REG0        ;XFC OPCODE
5182 021256 104031                      ERROR   31              ;WCS WATCH DOG TIMER TIMED OUT
5183                                     ;UPON TRANSFERRING CONTROL FROM
5184                                     ;BASE MACHINE TO WCS MICROCODE.
5185                                     ;NORMALLY, AFTER UCODE EXECUTION CONTROL
5186                                     ;SHOULD BE RETURNED TO BASE MACHINE.
5187
5188
5189
5190
5191
5192
5193                                     ;*****
5194                                     ;*TEST 45      CHECK "OTHER USER" DISPATCH TO 6015, XFC=076500-076577
5195                                     ;THIS TEST CHECKS THE "OTHER USER" DISPATCH OF THE XFC
5196                                     ;INSTRUCTION TO 6015. THE XFC OPCODE GROUP CHECKED
5197                                     ;IN THIS TEST IS 076500-076577. WHEN AN XFC
5198                                     ;OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE
5199                                     ;SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY
5200                                     ;POINT IN THE WCS IS 6015 (MICRO-ADDRESS IN
5201                                     ;PAGE 6.
5202                                     ;THE TEST SEQUENCE IS AS FOLLOWING:
5203
5204                                     ;MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH
5205                                     ;AT 6015. THE LOADED MICRO CODES FUNCTION IS TO READ
5206                                     ;CSP MD AND WRITE IT IN R3(BOTH A&B SIDE).
5207
5208                                     ;AN XFC OP-CODE IS EXECUTED
5209                                     ;THE CONTROL SHOULD TRANSFER TO 6015 IN WCS AND

```



```

5210 ; MICRO-CODE GETS EXECUTED. RETURN IS MADE BACK TO
5211 ; BASE MACHINE.
5212
5213 ; UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF
5214 ; THE CORRECT XFC-OPCODE WAS SAVED IN R3
5215 ; (FROM CSP MD WHILE RUNNING FROM WCS.
5216
5217 ; IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTED.
5218
5219 ; THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC
5220 ; OP-CODES IN THE GROUP.
5221
5222 ; THIS TEST USES THE WCS WATCH-DOG-TIMES. IF, AS A RESULT
5223 ; OF SOME FAULTY CONDITION, THE CONTROL IS NOT
5224 ; TRANSFERRED BACK TO THE BASE MACHINE THE WATCH-DOG
5225 ; TIMES OUT AND A TRAP TO 114 OCCURS
5226
5227 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN:
5228 ; BASE MACHINE XFC DECODE AND DISPATCH
5229 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT
5230 ; CONTROL-PATH USED BY THE MICRO-CODE EXECUTING FROM
5231 ; THE WCS. (NUA<9:0>), ARRAY ADDRESS MUX-APORT,
5232 ; BUS-U MUX-BPORT, BUS-U LINES.

```

\*\*\*\*\*

```

5234 021260 000004          TST45: SCOPE
5235 021262 012737 021430 003056  MOV #TST46,NXTST ; STARTING ADDRESS OF NEXT TEST
5236 021270 012700 000100  MOV #100,R0 ; LOAD COUNT
5237 021274 012701 076500  MOV #76500,R1 ; INITIALIZE XFC OPCODE
5238 021300 004537 037472  JSR R5,LDWCS ; LOAD WCS WITH MICRO-CODE
5239 021304 006405          6405 ; MAIN-CONTROL-BLOCK LOADED HERE
5240 021306 062162          MXFC15 ; FROM THIS MEMORY ADRES
5241 021310 006015          OTH15 ; ENTRY POINT INTO WCS
5242
5243 021312 010137 021336 15: MOV R1,25 ; SET UP XFC CODE FOR EXECUTION
5244 021316 012737 021400 000114  MOV #45,@#114 ; SET UP SERVICE VECTOR FOR WCS
5245 ; WATCH DOG TIMEOUT.
5246 021324 012777 001000 161502  MOV #MAINT,@WCSST ; SET MAINTENANCE BIT
5247 021332 005777 161502  TST @WCSDR ; KICK OFF THE WATCH-DOG TIMER
5248
5249 021336 000000 25: .WORD 0 ; THIS WORD WILL CONTAIN THE XFC OPCODE
5250 ; TO BE EXECUTED. AT THIS POINT CONTROL
5251 ; IS TRANSFERRED TO THE WCS.
5252
5253 ; RETURN FROM WCS OCCURS HERE
5254 021340 042777 001000 161466  BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
5255 021346 012737 003344 000114  MOV #BADPAR,@#114
5256 021354 020301          CMP R3,R1 ; WAS CORRECT OPCODE SAVED IN R4.
5257 ; WHILE IN WCS?
5258 021356 001405          BEQ 35 ; YES
5259 021360 010137 001162  MOV R1,$REG0 ; EXPCD XFC CODE
5260 021364 010337 001164  MOV R3,$REG1 ; CODE RECVD
5261 021370 104030          ERROR 30 ; CORRECT XFC OPCODE WAS NOT SAVED
5262 ; BY THE MICRO-CODE WHILE INSIDE WCS.
5263 ; NOTE THAT XFC OPCODE SHOULD HAVE
5264 ; TRANSFERRED CONTROL TO THE WCS.
5265

```

5266 021372 005201  
5267 021374 077032  
5268 021376 000414  
5269  
5270  
5271  
5272  
5273  
5274 021400 022626  
5275  
5276 021402 012737 003344 000114  
5277 021410 012777 000020 161416  
5278 021416 005077 161412  
5279 021422 010137 001162  
5280 021426 104031

35: INC R1 ;NEXT XFC OPCODE  
SOB RO,15 ;DONE?  
BR TST46 ;;EXIT  
  
;ENTER HERE IF THE WATCH-DOG TIMER ON THE WCS TIMED OUT BEFORE  
;AN ORDERLY EXIT AND RETURN OF CONTROL COULD BE DONE (FROM THE  
;WCS) TO THE BASE MACHINE.  
  
45: CMP (SP)+,(SP)+ ;POP THE STACK  
  
MOV #BADPAR,@#114  
MOV #PARDIS,@WCSST  
CLR @WCSST  
MOV R1,%REGO ;XFC OPCODE  
ERROR 31 ;WCS WATCH DOG TIMER TIMED OUT  
;UPON TRANSFERRING CONTROL FROM  
;BASE MACHINE TO WCS MICROCODE.  
;NORMALLY, AFTER UCODE EXECUTION CONTROL  
;SHOULD BE RETURNED TO BASE MACHINE.

5281  
5282  
5283  
5284  
5285  
5286  
5287  
5288  
5289  
5290  
5291  
5292  
5293  
5294  
5295  
5296  
5297  
5298  
5299  
5300  
5301  
5302  
5303  
5304  
5305  
5306  
5307  
5308  
5309  
5310  
5311  
5312  
5313  
5314  
5315  
5316  
5317  
5318  
5319  
5320  
5321

\*\*\*\*\*  
\*TEST 46 CHECK "USER" DISPATCH TO 6002, XFC=076700-076777  
;THIS TEST CHECKS THE "USER" DISPATCH OF THE XFC  
;INSTRUCTION TO 6002. THE XFC OPCODE GROUP CHECKED  
;IN THIS TEST IS 076700-076777. WHEN AN XFC  
;OPCODE FROM THIS GROUP IS EXECUTED, THE BASE MACHINE  
;SHOULD TRANSFER CONTROL TO THE WCS; THE ENTRY  
;POINT IN THE WCS IS 6002 (MICRO-ADDRESS IN  
;PAGE 6.  
;THE TEST SEQUENCE IS AS FOLLOWING:  
  
;MICRODE IS LOADED INTO THE WCS FOR SERVICING THE DISPATCH  
;AT 6002. THE LOADED MICRO CODES FUNCTION IS TO READ  
;CSP MD AND WRITE IT IN R3(BOTH A&B SIDE).  
  
;AN XFC OP-CODE IS EXECUTED.  
;THE CONTROL SHOULD TRANSFER TO 6002 IN WCS AND  
;MICRO-CODE GETS EXECUTED. RETURN IS MADE BACK TO  
;BASE MACHINE.  
  
;UPON RETURN TO THE BASE MACHINE, IT IS CHECKED IF  
;THE CORRECT XFC-OPCODE WAS SAVED IN R3  
;(FROM CSP MD WHILE RUNNING FROM WCS.  
  
;IF CORRECT OP-CODE WAS NOT SAVED AN ERROR IS REPORTED  
  
;THE ABOVE SEQUENCE IS REPEATED FOR ALL THE XFC  
;OP-CODES IN THE GROUP.  
  
;THIS TEST USES THE WCS WATCH-DOG-TIMES. IF, AS A RESULT  
;OF SOME FAULTY CONDITION, THE CONTROL IS NOT





;BASE MACHINE TO WCS MICROCODE.  
;NORMALLY, AFTER UCODE EXECUTION CONTROL  
;SHOULD BE RETURNED TO BASE MACHINE.

5378  
5379  
5380  
5381  
5382  
5383  
5384  
5385  
5386  
5387  
5388  
5389  
5390  
5391  
5392  
5393  
5394  
5395  
5396  
5397  
5398  
5399  
5400  
5401  
5402  
5403  
5404  
5405  
5406  
5407  
5408  
5409  
5410  
5411  
5412  
5413  
5414  
5415  
5416  
5417  
5418  
5419  
5420  
5421  
5422  
5423  
5424  
5425  
5426  
5427  
5428  
5429  
5430  
5431  
5432  
5433

;;\*\*\*\*\*

\*TEST 47 CHECK "ODD PC" DISPATCH TO WCS AT 6004  
;THIS TEST CHECKS THE "ODD PC" DISPATCH (FROM BASE MACHINE)  
;TO WCS AT 6004 (MICRO-ADDRESS IN PAGE6, WCS). WHEN  
;A TRAP/INTERRUPT OCCURS AND THE SERVICE VECTOR FOR  
;THAT CONTAINS AN ODD PC, THE BASE MACHINE  
;TRANSFERS CONTROL TO THE WCS (PROVIDED THE WCS  
;IS ENABLED). THE ENTRY IN WCS IS MADE AT 6004  
;(PAGE 6).  
;THE SEQUENCE OF TESTING IS:  
  
;BPT SERVICE VECTOR (14) IS SET UP TO CONTAIN AN ODD PC.  
;THE WCS IS LOADED UP WITH MICRO-CODE TO SERVICE THE  
;DISPATCH AT 6004. THE FUNCTION OF THIS MICRO-CODE  
;IS TO INCR R3 BY 1, WHICH WAS CLEARED PREVIOUSLY  
  
;BPT INSTRUCTION IS EXECUTED, BPT TRAP SHOULD TAKE PLACE  
;AND CONTROL SHOULD TRANSFER TO WCS AT MICRO-ADDRESS 6004.  
  
;UPON RETURN, THE FLAG (R3) IS CHECKED.  
;IF THE FLAG IS NOT =1 AN ERROR IS REPORTED.  
  
;IF THE ODD-PC-DISPATCH TO WCS DOES NOT OCCUR,  
;AN ODD-ADDRESS TRAP WILL OCCUR.  
;THIS TEST USES THE WCS WATCH-DOG TIMER. IF AS A RESULT  
;SOME FAULT CONDITION, THE CONTROL IS NOT TRANSFERRED BACK  
;TO THE BASE MACHINE, THE WATCH DOG WILL TIME OUT AND A  
;TRAP TO 114 WILL OCCUR.

;FAILURE OF THIS TEST COULD INDICATE A FAULT IN:  
; BASE-MACHINE ODD-PC DISPATCH MICRO-CODE  
; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT.  
; CONTROL-PATH USED BY THE MICRO-CODE EXECUTING.  
; FROM THE WCS. (NUA<9:0>, ARRAY ADDRESS MUX-A PORT,  
; BUS-U MUX -B PORT, BUS-U LINES.

;;\*\*\*\*\*

TST47: SCOPE  
MOV #TST50,NXTST ;STARTING ADDRESS OF NEXT TEST  
CLR R3 ;CLEAR FLAG  
MOV #35,@#4 ;ODD ADDRESS TRAP VECTOR  
MOV #1001,@#14 ;SET UP ODD PC AT BPT VECTOR  
MOV #25,@#114 ;SET UP SERVICE VECTOR WCS WATCH-DOG TIMER  
JSR R5,LDWCS ;LOAD UP THE WCS  
6161 ;STARTING AT THIS WCS ADDRESS (IN CONTROL SPACE)  
MODDPC ;FROM THIS MEMORY ADDRESS  
ODDPC ;ENTRY POINT INTO WCS  
MOV #MAINT,@WCSST ;INITIATE AND

021566 000004  
021570 012737 021754 003056  
021576 005003  
021600 012737 021702 000004  
021606 012737 001001 000014  
021614 012737 021722 000114  
021622 004537 037472  
021626 006161  
021630 062214  
021632 006004  
021634 012777 001000 161172



```

5434 021642 005777 161172      TST      @WCSDR      ; KICK OFF THE TIMER
5435 021646 000003              BPT              ; EXECUTE BPT, THE BPT VECTOR CONTAINS
5436                               ; AN ODD PC. CONTROL SHOULD TRANSFER TO
5437                               ; WCS AT CS ADDRESS = 6004.
5438
5439                               ; RETURN HERE FROM WCS
5440 021650 042777 001000 161156  BIC      #MAINT,@WCSST ; SHUT OFF THE WCS TIMER.
5441 021656 012737 003344 000114  MOV      #BADPAR,@#114 ; RESET THE UNEXPECTED PARITY HANDLER
5442 021664 022703 000001              CMP      #1,R3      ; WAS THE FLAG SET WHILE IN WCS?
5443 021670 001426              BEQ      4$          ; OK
5444 021672 010337 001162              MOV      R3,$REGO   ;
5445 021676 104034              ERROR    34         ; FLAG (IN R3) WAS NOT =1 BY THE MICRO-CODE
5446                               ; IN WCS, AFTER CONTROL WAS TRANSFERRED
5447                               ; TO WCS (6004) USING ODD PC DISPATCH.
5448 021700 000422              BR       4$          ;
5449 021702 012737 003276 000004 3$:  MOV      #BADTMO,@#4
5450 021710 011637 001162              MOV      (SP),$REGO
5451 021714 022626              CMP      (SP)+,(SP)+ ; POP THE STACK
5452 021716 104062              ERROR    62         ; INSTEAD OF "ODD PC" DISPATCH
5453                               ; TO WCS, AN ODD ADDRESS
5454                               ; TRAP OCCURED. FAULT COULD BE
5455                               ; IN ODD-PC DISPATCH IN
5456                               ; BASE MACHINE
5457
5458 021720 000415              BR       TST50      ; EXIT
5459
5460                               ; ENTER HERE IF WCS WATCH DOG TIMER TIMED OUT FOLLOWING A TRANSFER
5461                               ; OF CONTROL TO WCS AT 6004 USING ODD PC DISPATCH.
5462
5463 021722 012737 003344 000114 2$:  MOV      #BADPAR,@#114
5464 021730 022626              CMP      (SP)+,(SP)+
5465 021732 012777 000020 161074  MOV      #PARDIS,@WCSST
5466 021740 005077 161070              CLR      @WCSST
5467 021744 104035              ERROR    35         ; WCS WATCH DOG TIMER TIMED OUT
5468                               ; UPON TRANSFERRING CONTROL FROM
5469                               ; BASE MACHINE TO WCS MICROCODE.
5470                               ; NORMALLY, AFTER UCODE EXECUTION CONTROL
5471                               ; SHOULD BE RETURNED TO BASE MACHINE.
5472 021746 012737 003276 000004 4$:  MOV      #BADTMO,@#4
5473
5474
5475
5476
5477
5478                               ; *****
5479                               ; *TEST 50 CHECK "MICRO-BREAK" DISPATCH TO WCS AT 6000
5480                               ; THIS TEST CHECKS THE "MICRO-BREAK" DISPATCH (FROM-BASE
5481                               ; MACHINE) TO WCS OF 6000 (MICRO-ADDRESS IN PAGE6, WCS).
5482                               ; WHEN A MICROBREAK OCCURS (ENABLE MICROBREAK IN FLAG/INT.
5483                               ; REGISTER, CLEAR MICROBREAK-TRAP IN WHAMI) AND WCS IS
5484                               ; PRESENT AND ENABLED, THE BASE MACHINE TRANSFER
5485                               ; CONTROL TO THE WCS. THE WCS IS ENTERED AT
5486                               ; MICRO-ADDRESS 6000.
5487                               ; THE SEQUENCE OF TESTING IS=
5488
5489                               ; THE MICRO-BREAK REGISTER IS LOADED WITH MICRO-ADDRESS

```

```

5490 ;"SWB01" (IN SWAB MICRO-FLOW), THE MICRO-BREAK
5491 ;TRAP IS DISABLED, THE MICRO-BREAK IS ENABLED). WCS
5492 ;IS LOADED WITH MICRO-CODE TO SERVICE THE DISPATCH
5493 ;AT 6000. THE FUNCTION OF THIS MICRO-CODE IS TO INCR
5494 ;FLAG R3 BY 1, WHICH WAS CLEARED PREVIOUSLY.
5495
5496 ;"SWAB" INSTRUCTION IS EXECUTED, MICRO-BREAK SHOULD TAKE
5497 ;PLACE AND THE BASE MACHINE SHOULD TRANSFER
5498 ;CONTROL TO WCS AT 6000. MICRO-CODE SHOULD
5499 ;GET EXECUTED TO INCR R3 BY 1. RETURN IS MADE
5500 ;TO THE BASE MACHINE.
5501
5502 ;UPON RETURN, THE FLAG (R3) IS CHECKED IF IT =1
5503 ;IF NOT AN ERROR IS REPORTED.
5504
5505 ;THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS A
5506 ;RESULT OF SOME FAULT CONDITION, THE CONTROL IS NOT
5507 ;TRANSFERRED BACK TO THE BASE MACHINE, THE WATCH DOG
5508 ;WILL TIME OUT AND A TRAP TO 114 WILL OCCUR.
5509
5510 ;FAILURE OF THIS TEST COULD UNDIKATE A FAULT IN =
5511 ;
5512 ;   BASE MACHINE MICRO-BREAK DISPATCH MICRO-CODE
5513 ;   TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT.
5514 ;   CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
5515 ;   THE WCS. (NUA<9=0>, ARRAY ADDRESS MUX-A PORT,
5516 ;   BUS-U MUX-B PORT, BUS U LINES).

```

```

5516 ;:*****
5517 021754 000004 TST50: SCOPE
5518 021756 012737 022136 003056 MOV #TST51,NXTST ;STARTING ADDRESS OF NEXT TEST
5519 021764 005003 CLR R3 ;CLEAR FLAG
5520 021766 012737 000071 177770 MOV #SWB01,@#MBKREG ;LOAD MICRO-BREAK REGISTER
5521 021774 076600 MED
5522 021776 000022 RDWHAMI ;READ WHAMI
5523 022000 042700 001000 BIC #BIT9,R0 ;DISABLE UBREAK TRAP
5524 022004 076600 MED
5525 022006 000222 WRWHAMI ;WRITE BACK INTO WHAMI
5526 022010 076600 MED
5527 022012 000144 RDFLAG ;READ FLAG/INT
5528 022014 052700 100000 BIS #BIT15,R0 ;ENABLE MICROBREAK IN FLAG/INT
5529 022020 076600 MED
5530 022022 000344 WRFLAG ;WRITE BACK INTO FLAG REGISTER
5531 022024 012737 022112 000114 MOV #29,@#114 ;SET UP SERVICE VECTOR FOR WCS WATCH
5532 ;DOG TIMER
5533 022032 004537 037472 JSR R5,LDWCS ;LOAD WITH MICRO-CODE WCS
5534 022036 006171 6171 ;MAIN CONTROL BLOCK LOADED HERE
5535 022040 062246 MUBRK ;FROM THIS MEMORY LOCATION
5536 022042 006000 UBRK ;ENTRY POINT INTO WCS
5537 022044 012777 001000 160762 MOV #MAINT,@WCSST ;SET MAINT. BIT AND
5538 022052 005777 160762 TST @WCSDR ;INITIATE THE WCS TIMER
5539 022056 000300 SWAB R0 ;EXECUTE INSTRUCTION LEADING TO
5540 ;MICRO-BREAK. SINCE THE WCS IS
5541 ;ENABLED, CONTROL SHOULD TRANSFER
5542 ;TO WCS (6000) AT THIS POINT.
5543
5544 ;RETURN HERE FROM WCS
5545 022060 042777 001000 160746 BIC #MAINT,@WCSST ;SHUT OFF WCS TIMER

```



5546 022066 012737 003344 000114  
5547 022074 022703 000001  
5548 022100 001416  
5549 022102 010337 001162  
5550 022106 104036  
5551  
5552  
5553  
5554 022110 000412  
5555  
5556  
5557  
5558  
5559 022112 012737 003344 000114 25:  
5560 022120 022626  
5561 022122 012777 000020 160704  
5562 022130 005077 160700  
5563 022134 104037  
5564  
5565  
5566  
5567  
5568  
5569  
5570  
5571  
5572  
5573  
5574  
5575  
5576  
5577  
5578  
5579  
5580  
5581  
5582  
5583  
5584  
5585  
5586  
5587  
5588  
5589  
5590  
5591  
5592  
5593  
5594  
5595  
5596  
5597  
5598  
5599  
5600  
5601

```
MOV #BADPAR, @#114 ;REINCR UNEXPECTED PARITY HANDLER
CMP #1, R3 ;WAS THE FLAG INCR WHILE IN WCS?
BEQ TST51 ;;EXIT
MOV R3, $REGO ;
ERROR 36 ;FLAG WAS NOT INCR BY THE
;MICRO-CODE IN WCS, AFTER CONTROL
;WAS TRANSFERRED TO WCS (6000),
;USING "MICRO-BREAK" DISPATCH
BR TST51 ;;EXIT

;ENTER HERE IF WCS WATCH DOG TIMER TIMED OUT FOLLOWING A TRANSFER
;OF CONTROL TO WCS AT 6000 USING MICRO-BREAK DISPATCH.

MOV #BADPAR, @#114
CMP (SP)+, (SP)+
MOV #PARDIS, @WCSST
CLR @WCSST
ERROR 37 ;WCS WATCH DOG TIMER TIMED OUT
;UPON TRANSFERRING CONTROL FROM
;BASE MACHINE TO WCS MICROCODE.
;NORMALLY, AFTER UCODE EXECUTION CONTROL
;SHOULD BE RETURNED TO BASE MACHINE.

;*****
;*TEST 51 CHECK "FLAG<7> WCS SERVICE" DISPATCH TO WCS AT 6005
;THIS TEST CHECKS THE "FLAG<7> WCS SERVICE" DISPATCH (FROM-BASE
;MACHINE) TO WCS OF 6005 (MICRO-ADDRESS IN PAGE 6, WCS).
;WHEN WCS IS ENABLED, AND FLAG<7> (BIT14) IN FLAG REGISTER IS SET,
;BM SERVICE CODE WILL ENTER WCS AT MICROADDRESS 6005, ASSUMING
;NO OTHER SERVICE CONDITIONS OF HIGHER PRIORITY ARE PENDING.
;THE SEQUENCE OF TESTING IS:

;THE WCS IS LOADED WITH MICRO-CODE TO SERVICE THE DISPATCH
;TO 6005. THE FUNCTION OF THIS MICRO-CODE IS TO INCR
;FLAG R3 BY 1, WHICH WAS CLEARED PREVIOUSLY.

;MED INSTRUCTION IS USED TO SET FLAG<7> (BIT14). THIS FORCES
;BM "SERVICE" TO BE SIGNALLED, AND THUS CONTROL PASSES TO
;WCS AT 6005. MICRO-CODE SHOULD GET EXECUTED TO INCR R3 BY 1.
;RETURN IS MADE TO THE BASE MACHINE AT "FETO1" WITHOUT CHECKING
;FOR "SERVICE" AGAIN. THIS ALLOWS NEXT MACRO-INSTRUCTION TO
;BE EXECUTED BEFORE "SERVICE" IS CHECKED AGAIN.

;UPON RETURN, THE FLAG (R3) IS CHECKED IF IT =2
;IF NOT AN ERROR IS REPORTED.

;THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS A
;RESULT OF SOME FAULT CONDITION, THE CONTROL IS NOT
;TRANSFERRED BACK TO THE BASE MACHINE, THE WATCH DOG
;WILL TIME OUT AND A TRAP TO 114 WILL OCCUR.

;FAILURE OF THIS TEST COULD UNDIKATE A FAULT IN =
```

```
5602 ; BASE MACHINE MICRO-BREAK DISPATCH MICRO-CODE
5603 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT CIRCUIT.
5604 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
5605 ; THE WCS. (NUA<9=0>, ARRAY ADDRESS MUX-A PORT,
5606 ; BUS-U MUX-B PORT, BUS U LINES).
5607 ; *****
5608 022136 000004 TST51: SCOPE
5609 022140 012737 022304 003056 MOV #TST52,NXTST ;STARTING ADDRESS OF NEXT TEST
5610 022146 005003 CLR R3 ;CLEAR FLAG
5611 022150 012737 022260 000114 MOV #25,@#114 ;SET UP SERVICE VECTOR FOR WCS WATCH DOG TIMER
5612 022156 004537 037472 JSR R5,LDWCS ;LOAD WITH MICRO-CODE WCS
5613 022162 006770 6770 ;MAIN CONTROL BLOCK LOADED HERE
5614 022164 062332 MF7SVC ;FROM THIS MEMORY LOCATION
5615 022166 006005 F7SVC ;ENTRY POINT INTO WCS
5616 ;
5617 022170 076600 000144 MED ,RDFLAG ;GET FLAGS IN R0
5618 022174 052700 040000 BIS #BIT14,R0 ;SET FLAG<7> AT BIT14
5619 022200 012777 001000 160626 MOV #MAINT,@WCSST ;SET MAINT. BIT AND
5620 022206 005777 160626 TST @WCSDR ;INITIATE THE WCS TIMER
5621 ;
5622 022212 076600 000344 MED ,WRFLAG ;SET WCS SERVICE FLAG<7>
5623 ;SINCE THE WCS IS ENABLED, CONTROL SHOULD
5624 ;TRANSFER TO WCS (6005) AT THIS POINT.
5625 ;WCS DOES R3 <- R3 PLUS 1 SINCE FLAG<7> SET, R3 = 1 AFTER
5626 ;
5627 022216 042700 040000 BIC #BIT14,R0 ;NOW CLR FLAG<7> AT BIT14
5628 ;WCS DOES R3 <- R3 PLUS 1 SINCE FLAG<7> SET, R3 = 2 AFTER
5629 ;
5630 022222 076600 000344 MED ,WRFLAG ;NOW CLR WCS SERVICE FLAG<7>
5631 ;NO MORE ENTRY TO WCS, FLAG<7> NOW CLEAR
5632 ;
5633 022226 042777 001000 160600 BIC #MAINT,@WCSST ;SHUT OFF WCS TIMER
5634 022234 012737 003344 000114 MOV #BADPAR,@#114 ;RESTORE UNEXPECTED PARITY HANDLER
5635 022242 022703 000002 CMP #2,R3 ;WAS THE FLAG INCR TWICE WHILE IN WCS?
5636 022246 001416 BEQ TST52 ;EXIT
5637 022250 010337 001162 MOV R3,$REGO ;
5638 022254 104074 ERROR 74 ;FLAG WAS NOT INCR BY THE
5639 ;MICRO-CODE IN WCS, AFTER CONTROL
5640 ;WAS TRANSFERRED TO WCS (6005),
5641 ;USING "FLAG<7> WCS SERVICE" DISPATCH
5642 022256 000412 BR TST52 ;EXIT
5643 ;
5644 ;ENTER HERE IF WCS WATCH DOG TIMER TIMED OUT FOLLOWING A TRANSFER
5645 ;OF CONTROL TO WCS AT 6005 USING FLAG<7> WCS SERVICE DISPATCH.
5646 ;
5647 022260 012737 003344 000114 25: MOV #BADPAR,@#114
5648 022266 022626 CMP (SP)+,(SP)+
5649 022270 012777 000020 160536 MOV #PARDIS,@WCSST
5650 022276 005077 160532 CLR @WCSST
5651 022302 104075 ERROR 75 ;WCS WATCH DOG TIMER TIMED OUT
5652 ;UPON TRANSFERRING CONTROL FROM
5653 ;BASE MACHINE TO WCS MICROCODE.
5654 ;NORMALLY, AFTER UCODE EXECUTION CONTROL
5655 ;SHOULD BE RETURNED TO BASE MACHINE.
5656
5657
```



5658  
5659  
5660  
5661  
5662  
5663  
5664  
5665  
5666  
5667  
5668  
5669  
5670  
5671  
5672  
5673  
5674  
5675  
5676  
5677  
5678  
5679  
5680  
5681  
5682  
5683  
5684  
5685  
5686  
5687  
5688  
5689  
5690  
5691  
5692  
5693  
5694  
5695  
5696  
5697  
5698  
5699  
5700

```

;*****
;*TEST 52 CHECK "RESERVED INSTRUCTION" DISPATCH TO 6003 IN WCS
;THIS TEST CHECKS "RESERVED INSTRUCTION" DISPATCH
;FROM THE BASE MACHINE TO THE WCS. THE ENTRY
;POINT IN WCS IS AT MICRO-ADDRESS 6003 (PAGE 6).
;IF THE WCS IS ENABLED AND A RESERVED
;INSTRUCTION IS EXECUTED, THE BASE MACHINE TRANSFERS
;CONTROL TO THE WCS AT MICRO-ADDRESS 6003.

;THE SEQUENCE OF TESTING IS AS FOLLOWING:
;THE WCS IS LOADED WITH MICRO-CODE TO SERVICE
;THE "RESERVED INSTRUCTION" DISPATCH.

;A RESERVED INSTRUCTION IS EXECUTED. SINCE THE
;WCS IS ENABLED THE BASE MACHINE SHOULD
;TRANSFER CONTROL TO THE WCS AT 6003

;THE PREVIOUSLY LOADED MICRO-CODE IN THE WCS
;SHOULD SAVE RESERVED INSTRUCTION OPCODE IN R3. AND
;RETURN CONTROL TO THE BASE MACHINE.

;BACK IN BASE MACHINE, IT IS CHECKED IF THE
;CORRECT OPCODE WAS SAVED IN R3. IF NOT, AN
;ERROR IS REPORTED.

;THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS A
;RESULT OF SOME FAULT CONDITION, THE CONTROL IS
;NOT TRANSFERRED BACK TO THE BASE MACHINE, THE
;WATCH-DOG WILL TIME OUT AND A TRAP TO 114
;WILL OCCUR

;FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE =
; RESERVED INSTRUCTION DECODE AND DISPATCH
; IN BASE MACHINE.
; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
; THE WCS. (NUA<9:0>, APRAY ADDRESS MUX-A-PORT,
; BUS-U-MUX-B-PORT, BUS-U-LINES.
; ENTRY POINT REGISTER IN WCS.

```

```

5701
5702 022304 000004
5703 022306 012737 022502 003056
5704
5705 022314 012704 177773
5706 022320 010402
5707 022322 012737 022470 000010
5708 022330 012700 000020
5709 022334 012701 000210
5710 022340 004537 037472
5711 022344 006201
5712 022346 062300
5713 022350 006003

```

```

;*****
TST52: SCOPE
MOV #TST53,NXTST ;STARTING ADDRESS OF NEXT TEST

MOV #-5,R4 ;DONT REPORT MORE THAN
MOV R4,R2 ;5 ERRORS EACH
MOV #65,0#10 ;ILLEGAL INST. TRAP
MOV #20,R0
MOV #210,R1
JSR R5,LDWCS
6201
MRESIN
URESIN

```

```

5714
5715 022352 010137 022376      15:  MOV      R1,25
5716 022356 012737 022436 000114  MOV      #45,@#114
5717 022364 012777 001000 160442  MOV      #MAINT,@WCSST
5718 022372 005777 160442      TST      @WCSDR
5719
5720 022376 000000      25:  .WORD    0
5721 022400 042777 001000 160426  BIC      #MAINT,@WCSST
5722 022406 020301      CMP      R3,R1
5723 022410 001407      BEQ      35
5724 022412 005204      INC      R4
5725 022414 100032      BPL      TST53          ;;EXIT
5726 022416 010137 001162  MOV      R1,$REG0
5727 022422 010337 001164  MOV      R3,$REG1
5728 022426 104044      ERROR   44
5729
5730 022430 005201      35:  INC      R1
5731 022432 077031      SOB     RO,15
5732 022434 000422      BR      TST53          ;;EXIT
5733
5734 022436 022626      45:  CMP      (SP)+,(SP)+
5735 022440 012737 003344 000114  MOV      #BADPAR,@#114
5736 022446 012777 000020 160360  MOV      #PARDIS,@WCSST
5737 022454 005077 160354      CLR     @WCSST
5738 022460 005202      INC     R2
5739 022462 100007      BPL     TST53          ;;EXIT
5740 022464 104045      ERROR   45          ;WCS WATCH DOG TIMED OUT
5741                                     ;UPON TRANSFERRING CONTROL FROM
5742                                     ;BASE MACHINE TO WCS MICROCODE.
5743                                     ;NORMALLY, AFTER UCODE EXECUTION CONTROL
5744                                     ;SHOULD BE RETURNED TO BASE MACHINE.
5745
5746 022466 000760      BP      35
5747
5748 022470 022626      65:  CMP      (SP)+,(SP)+
5749 022472 012737 000012 000010  MOV      #12,@#10
5750 022500 104063      ERROR   63
5751
5752
5753
5754
5755
5756
5757
5758 ;*****
5759 ;*TEST 53 CHECK THAT XFC TRAPS OUT AS RESVD-INST WHEN WCS IS DISABLED
5760 ;THIS TEST CHECKS THAT THE XFC OPCODES, WHEN EXECUTED
5761 ;WITH THE WCS DISABLED, TRAPS OUT AS RESERVED INSTRUCTIONS
5762 ;TO VECTOR 10. THE FOLLOWING XFC OPCODES ARE TESTED:
5763 ; 076000-076577
5764 ; 076601-076777
5765 ; 076600 IS MED INSTRUCTION
5766 ;PREVIOUS TESTS HAVE CHECKED OUT THAT THE SAME XFC
5767 ;OP-CODES PROVIDE DISPATCH (AND TRANSFER OF CONTROL) TO
5768 ;WCS, WHEN THE WCS IS ENABLED.
5769 ;*****
022502 000004 TST53: SCOPE

```



```

5770 022504 012737 023040 003056      MOV      #TST54,NXTST ; STARTING ADDRESS OF NEXT TEST
5771 022512 005004                      CLR      R4
5772 022514 005000                      CLR      R0 ; COUNT
5773 022516 012705 076000      MOV      #076000,R5 ; INITIALIZE XFC OP CODE
5774 022522 076600                      MED
5775 022524 000022                      RDWHAMI
5776 022526 042700 000200      BIC      #BIT7,R0
5777 022532 076600                      MED ; DISABLE THE WCS
5778 022534 000222                      WRWHAMI
5779 022536 016001 003124      15:     MOV      WCSENT(R0),R1 ; GET ENTRY POINTER INTO WCS
5780 022542 042701 177000      BIC      #177000,R1 ; FORM WCS ADDRESS AND DEPOSIT
5781 022546 010177 160264      MOV      R1,@WC SAR ; BRANCH ON SELF MICRO-WORD THERE
5782 022552 005101                      COM      R1 ; NUA<8:0> BITS ARE INVERTED
5783 022554 042701 177000      BIC      #177000,R1
5784 022560 052701 030000      BIS      #BIT12+BIT13,R1 ; NULL "BUT"
5785 022564 010177 160250      MOV      R1,@WCSDR
5786
5787 022570 062777 002000 160240      ADD      #2000,@WC SAR
5788 022576 005077 160236                      CLR      @WCSDR
5789 022602 062777 002000 160226      ADD      #2000,@WC SAR
5790 022610 005077 160224                      CLR      @WCSDR
5791
5792 022614 010537 022646      25:     MOV      R5,35 ; XFC OP-CODE
5793 022620 012737 022746 000114      MOV      #55,@#114 ; WCS WATCH-DOG TIMEOUT VECTOR
5794 022626 012737 022666 000010      MOV      #45,@#10 ; RESERVD. INST. TRAP VECT
5795 022634 012777 001000 160172      MOV      #MAINT,@WC SST ; KICK ON THE WCS TIMER
5796 022642 005777 160172      TST      @WCSDR
5797
5798 022646 000000      35:     .WORD    0 ; XFC OP-CODE HERE
5799 022650 042777 001000 160156      BIC      #MAINT,@WC SST ; SHUT OFF TIMER INCASE OF ERROR
5800 022656 010537 001162                      MOV      R5,$REGO
5801 022662 104066                      ERROR    66 ; XFC OF CODE DID NOT TRAP AS RCRVD INSTR.
5802 022664 000444                      BR       85 ; CHECK IF MORE THAN 5 ERRORS
5803
5804 022666 022626      45:     CMP      (SP)+,(SP)+ ; OCCURED, AS EXPECTED
5805 022670 042777 001000 160136      75:     BIC      #MAINT,@WC SST ; SHUT OFF TIMER
5806 022676 012737 000012 000010      MOV      #12,@#10 ; NORMAL TRAP-CATCHER
5807 022704 012737 003344 000114      MOV      #BADPAR,@#114
5808 022712 005205                      INC      R5 ; NEXT XFC OF-CODE
5809 022714 032705 000077                      BIT      #77,R5 ; DONE THIS GROUP?
5810 022720 001335                      BNE     25 ; IF NOT DO IT
5811 022722 062700 000002                      ADD      #2,R0 ; SHIFT POINTER TO NEXT GROUP
5812 022726 020027 000020                      CMP      R0,#20 ; DONE ALL 8 GROUPS?
5813 022732 002026                      BGE     65 ; YES, EXIT
5814 022734 020027 000014                      CMP      R0,#14 ; IS THIS THE GROUP HAVING "MED"?
5815 022740 001276                      BNE     15
5816 022742 005205                      INC      R5 ; IF SO, SKIP THE MED OP-CODE
5817 022744 000723                      BR       25 ; DO THE REST
5818
5819
5820
5821
5822
5823
5824
5825

```

```

; ENTER HERE IF THE WCS-WATCH
; DOG TIMED OUT, INDICATING
; THAT THE XFC OP-CODE
; TRANSFERRED CONTROL TO THE
; WCS (INSTEAD OF TRAPPING
; OUT AS A RESERVED INSTR.)

```





5854  
5855  
5856  
5857  
5858  
5859  
5860  
5861  
5862  
5863  
5864  
5865  
5866  
5867  
5868  
5869  
5870  
5871  
5872  
5873  
5874  
5875  
5876  
5877  
5878  
5879  
5880  
5881  
5882  
5883  
5884  
5885  
5886  
5887  
5888  
5889  
5890  
5891  
5892  
5893  
5894  
5895  
5896  
5897  
5898  
5899  
5900  
5901  
5902  
5903  
5904  
5905  
5906  
5907  
5908  
5909

.SBTTL ---CHECK OF TMS ROM FUNCTIONS---

```

;*****
;*TEST 54      CHECK BLOCK MOVE OF GR'S TO WCS
;              THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS.
;              WHEN THIS FUNCTION IS INITIATED A BLOCK OF GR'S
;              GET TRANSFERRED FROM THE BASE MACHINE TO THE
;              WCS. THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER
;
;              R7,R6(KERNEL), R6(USER), R5,R4,R3,R2,R1,R0
;
;              THE GR'S ARE INITIALIZED TO A KNOWN STATE, BY WRITING
;              UNIQUE DATA PATTERNS IN THEM.
;
;              THE MICRO-CODE IS LOADED INTO THE WCS. USING
;              A "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
;              TO THE MICRO-CODE IN THE WCS.
;
;              THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE
;              ADDRESS (WCS) STARTING WHICH THE GR'S SHOULD BE
;              SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS THE WCS
;              HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
;              WHICH ACTUALLY DOES THE STORE OF THE GR'S. AFTER THE
;              GR'S HAVE BEEN SAVED, THE MICRO-CODE RETURNS
;              CONTROL TO THE BASE MACHINE.
;
;              THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
;              DATA FROM THE GR'S WAS STORED CORRECTLY. IF NOT,
;              AN ERROR IS REPORTED.
;
;              THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
;              A RESULT OF SOME FAULT CONDITION, THE CONTROL
;              IS NOT TRANSFERRED BACK TO THE BASES MACHINE,
;              THE WATCH DOG WILL TIME OUT AND A TRAP TO
;              114 WILL OCCUR
;
;              FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
;              TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARPAV
;              BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
;              CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
;              THE WCS- (NVA<9=0>,ARRAY ADDRESS MUX-A-PORT,
;              BUS-U-MUX-B PORT, BUS-U-LINES).

```

```

023040 000004
023042 012737 023450 003056
023050 005000
023052 012701 000111
023056 012702 000222
023062 012703 000333

```

```

;*****
TST54: SCOPE
MOV     #TST55,NXTST ;STARTING ADDRESS OF NEXT TEST
CLR     R0           ;INITIALIZE GPR'S 0-5 WITH
MOV     #111,R1     ;UNIQUE DATA PATTERNS
MOV     #222,R2
MOV     #333,R3

```

5910	023066	012704	000444		MOV	#444, R4	
5911	023072	012705	000555		MOV	#555, R5	
5912	023076	012737	023424	000114	MOV	#65, @#114	; SETUP SERVICE VECTOR FOR WCS WATCH DOG
5913							; TIMER
5914	023104	004537	037472		JSR	R5, LDWCS	; LOAD WCS WITH MICRO CODE
5915	023110	006026			6026		; CONTROL BLOCK LOADED HERE
5916	023112	060666			MGRLS		; FROM THIS MEMORY LOCATION
5917	023114	006002			UDISP		; ENTRY POINT IN WCS
5918	023116	012777	001000	157710	MOV	#MAINT, @WCSST	; SET UP MAINT BIT, TIMER WILL START WHEN
5919							; WCS HOT BOX IS SELECTED, UPON TRANSFERRING
5920							; CONTROL TO WCS
5921	023124	012737	140000	177776	MOV	#140000, @#PSW	; USER MODE
5922	023132	012706	001500		MOV	#1500, R6	; INITIALIZE USER STACK
5923	023136	005037	177776		CLR	@#PSW	; BACK TO KERNEL MODE
5924	023142	012706	001100		MOV	#1100, R6	; INITIALIZE R6-KERNEL
5925							
5926	023146	076700			XFCUDIS		; TRANSFER CONTROL TO WCS
5927							
5928	023150						; RETURN FROM WCS TO THIS POINT
5929	023150	042777	001000	157656	BIC	#MAINT, @WCSST	; SHUT OFF THE TIMER
5930	023156	012737	003344	000114	MOV	#BADPAR, @#114	; RESET UNEXPECTED PARITY HANDLER
5931	023164	012777	001000	157644	MOV	#1000, @WCSAR	; ADDRESS THE LOCATION IN WCS
5932							; STARTING WHICH THE GPR'S WERE
5933							; SAVED. (LOC 1000 IN WCS)
5934							
5935	023172	017700	157640		MOV	@WCSAR, R0	; READ WCS FOR SAVED R7
5936	023176	020027	023150		CMP	R0, #15	; WAS R7 SAVED CORRECTLY?
5937	023202	001414			BEQ	25	
5938							
5939	023204	017737	157626	001162	MOV	@WCSAR, \$REG0	
5940	023212	012737	023150	001164	MOV	#15, \$REG1	
5941	023220	010037	001166		MOV	R0, \$REG2	
5942	023224	012737	000007	001170	MOV	#7, \$REG3	
5943	023232	104040			ERROR	40	; R7 WAS NOT SAVED CORRECTLY DURING
5944							; BLOCK MOVE OF GP'S TO WCS.
5945	023234	005277	157576		INC	@WCSAR	
5946	023240	017700	157574		MOV	@WCSAR, R0	
5947	023244	020027	001100		CMP	R0, #1100	
5948	023250	001414			BEQ	35	
5949	023252	017737	157560	001162	MOV	@WCSAR, \$REG0	
5950	023260	012737	001100	001164	MOV	#1100, \$REG1	
5951	023266	010037	001166		MOV	R0, \$REG2	
5952	023272	012737	000006	001170	MOV	#6, \$REG3	
5953	023300	104040			ERROR	40	
5954	023302	005277	157530		INC	@WCSAR	
5955	023306	017700	157526		MOV	@WCSAR, R0	; READ WCS FOR SAVED R6
5956	023312	020027	001500		CMP	R0, #1500	; WAS R6 SAVED CORRECTLY?
5957	023316	001414			BEQ	45	; YES
5958							
5959	023320	017737	157512	001162	MOV	@WCSAR, \$REG0	
5960	023326	012737	001500	001164	MOV	#1500, \$REG1	
5961	023334	010037	001166		MOV	R0, \$REG2	
5962	023340	012737	000016	001170	MOV	#16, \$REG3	
5963	023346	104040			ERROR	40	; R6 WAS NOT SAVED CORRECTLY DURING
5964							; BLOCK MOVE OF GP'S TO WCS
5965	023350	012700	000555		MOV	#555, R0	



```

5966 023354 005277 157456      75:  INC      @WCSAR
5967 023360 017701 157454      MOV      @WCSDR,R1
5968 023364 020001                CMP      RO,R1
5969 023366 001412                BEQ      55
5970 023370 017737 157442 001162  MOV      @WCSAR,$REGO
5971 023376 010037 001164      MOV      RO,$REG1
5972 023402 010137 001166      MOV      R1,$REG2
5973 023406 010037 001170      MOV      RO,$REG3
5974
5975 023412 104040                ERROR    40          ;CORRECT DATA WAS NOT SAVED IN
5976                                ;WCS DURING BLOCK MOVE FUNCTION.
5977                                ;(SAVING RO-R7).
5978 023414 162700 000111      55:  SUB      #111,RO
5979 023420 100355                BPL      75
5980
5981 023422 000412                BR       TST55      ;;EXIT
5982
5983                                ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
5984                                ; A BLOCK MOVE FUNCTION (GR'S TO WCS) IN THE WCS.
5985
5986 023424 012777 000020 157402 65:  MOV      #FARDIS,@WCSST
5987 023432 022626                CMP      (SP)+,(SP)+
5988 023434 005077 157374                CLR      @WCSST
5989 023440 012737 076700 001162  MOV      #XFCUDIS,$REGO
5990 023446 104041                ERROR    41          ;WCS WATCH DOG TIMED OUT
5991                                ;UPON TRANSFERRING CONTROL FROM
5992                                ;BASE MACHINE TO WCS MICROCODE.
5993                                ;NORMALLY, AFTER UCODE EXECUTION CONTROL
5994                                ;SHOULD BE RETURNED TO BASE MACHINE.
5995
5996
5997                                ;*****
5998                                ;*TEST 55 CHECK BLOCK MOVE FROM WCS TO GR'S
5999                                ; THIS TEST CHECKS THE BLOCK LOAD GR FUNCTION OF THE WCS.
6000                                ; WHEN THIS FUNCTION IS INITIATED GR'S IN THE BASE-
6001                                ; MACHINE ARE LOADED FROM THE WCS. THE FOLLOWING REGISTERS
6002                                ; ARE LOADED IN THE GIVEN ORDER=
6003
6004                                ; R7,R6(KERNEL), R6 (USER), R5,R4,R3,R2,R1,RO
6005
6006                                ; A BLOCK OF NINE LOCATIONS IN THE WCS IS INITIALIZED
6007                                ; TO A KNOWN STATE, BY WRITING UNIQUE DATA PATTERNS
6008                                ; IN THEM.
6009
6010                                ; THE MICRO-CODE IS LOADED INTO THE WCS. USING
6011                                ; A "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
6012                                ; TO THE MICRO-CODE IN THE WCS.
6013
6014                                ; THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE
6015                                ; ADDRESS, (WCS) STARTING WHICH THE WORDS WILL BE
6016                                ; LOADED FROM THE WCS INTO THE GR'S. THEN THIS
6017                                ; MICRO-CODE SELECTS THE WCS HOT BOX AND TRANSFERS
6018                                ; CONTROL TO THE TMS-ROM WHICH ACTUALLY DOES THE
6019                                ; LOAD OF GR'S. AFTER THE GR'S HAVE BEEN LOADED,
6020                                ; THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE.
6021

```

```

6022 ; THE GR'S ARE THEN CHECKED TO MAKE SURE THAT THE
6023 ; CORRECT DATA WAS LOADED FROM THE WCS INTO THE
6024 ; GR'S. IF NOT AN ERROR IS REPORTED.
6025
6026 ; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
6027 ; A RESULT OF SOME FAULT CONDITION, THE CONTROL IS
6028 ; NOT TRANSFERRED BACK TO THE BASE MACHINE, THE
6029 ; WATCH-DOG WILL TIME OUT AND A TRAP TO 114 WILL
6030 ; OCCUR.
6031
6032 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE=
6033 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINTER CIRCUIT.
6034 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION.
6035 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
6036 ; THE WCS. (NUA(9=0), ARRAY ADDRESS MUX-A-PORT,
6037 ; BUS-U-MUX-B-PORT, BUS-U LINES).
6038
6039 ;*****

```

```

6040 023450 000004          ;*****
6041 023452 012737 024276 003056 TST55: SCOPE
6042 023460 004537 037472          MOV #TST56,NXTST ; STARTING ADDRESS OF NEXT TEST
6043 023464 006136          JSR R5,LDWCS ; LOAD WCS WITH MICRO-CODE
6044 023466 060772          6136 ; MAIN CONTROL BLOCK LOADED HERE
6045 023470 006002          MLSGR ; FROM THIS MEMORY ADDRESS
6046 ; UDISP ; ENTRY POINT INTO WCS
6047 023472 013701 003036          MOV WCSAR,R1
6048 023476 013700 003040          MOV WCSDR,R0
6049 023502 012711 002525          MOV #2525,@R1 ; ADDRESS THE WCS
6050 ; ; INITIALIZE A BLOCK (9 WORDS)
6051 023506 012710 023616          MOV #15,@R0 ; PC
6052 023512 005211          INC @R1
6053 023514 012710 001070          MOV #1070,(R0) ; R6 (KERNEL)
6054 023520 005211          INC @R1
6055 023522 012710 001060          MOV #1060,(R0) ; R6 (USER)
6056 023526 005211          INC @R1
6057 023530 012710 000015          MOV #15,(R0) ; R5
6058 023534 005211          INC @R1
6059 023536 012710 000014          MOV #14,(R0) ; R4
6060 023542 005211          INC @R1
6061 023544 012710 000013          MOV #13,(R0) ; R3
6062 023550 005211          INC @R1
6063 023552 012710 000012          MOV #12,(R0) ; R2
6064 023556 005211          INC @R1
6065 023560 012710 000011          MOV #11,(R0) ; R1
6066 023564 005211          INC @R1
6067 023566 012710 000010          MOV #10,(R0) ; R0
6068 023572 012737 024252 000114          MOV #155,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG
6069 ; ; TIMER
6070
6071 023600 012777 001000 157226          MOV #MAINT,@WCSST ; SETUP MAINT BIT, TIMER WILL START
6072 ; ; WHEN WCS HOT BOX IS SELECTED, UPON
6073 ; ; TRANSFERRING CONTROL TO WCS.
6074 023606 012706 001074          MOV #1074,R6
6075 023612 076700          XFCUDIS ; TRANSFER CONTROL TO WCS
6076 023614 000000          HALT
6077 023616          15: ; RETURN FROM WCS TO THIS POINT

```



6078								
6079	023616	042777	001000	157210		BIC	#MAINT,@WCSST	; SHUT OFF THE TIMER
6080	023624	012737	003344	000114		MOV	#BADPAR,@#114	; RESET UNEXPECTED BAD PARITY HANDLER
6081	023632	010637	003046			MOV	R6, TMP0	
6082	023636	012706	001100			MOV	#1100, R6	
6083	023642	020027	000010			CMP	RO, #10	; CORRECT DATA RESTORED IN RO FROM WCS?
6084	023646	001414				BEQ	25	
6085	023650	012737	002535	001170		MOV	#2535, \$REG3	
6086	023656	012737	000010	001164		MOV	#10, \$REG1	
6087	023664	010037	001166			MOV	RO, \$REG2	
6088	023670	012737	000000	001162		MOV	#0, \$REG0	
6089	023676	104042				ERROR	42	
6090								
6091	023700	020127	000011		25:	CMP	R1, #11	; CORRECT DATA RESTORED IN R1 FROM WCS?
6092	023704	001414				BEQ	35	
6093	023706	012737	002534	001170		MOV	#2534, \$REG3	
6094	023714	012737	000011	001164		MOV	#11, \$REG1	
6095	023722	010137	001166			MOV	R1, \$REG2	
6096	023726	012737	000001	001162		MOV	#1, \$REG0	
6097	023734	104042				ERROR	42	
6098								
6099	023736	020227	000012		35:	CMP	R2, #12	; CORRECT DATA RESTORED IN R2 FROM WCS?
6100	023742	001414				BEQ	45	
6101	023744	012737	002533	001170		MOV	#2533, \$REG3	
6102	023752	012737	000012	001164		MOV	#12, \$REG1	
6103	023760	010237	001166			MOV	R2, \$REG2	
6104	023764	012737	000002	001162		MOV	#2, \$REG0	
6105	023772	104042				ERROR	42	
6106								
6107	023774	020327	000013		45:	CMP	R3, #13	
6108	024000	001414				BEQ	55	
6109	024002	012737	002532	001170		MOV	#2532, \$REG3	
6110	024010	012737	000013	001164		MOV	#13, \$REG1	
6111	024016	010337	001166			MOV	R3, \$REG2	
6112	024022	012737	000003	001162		MOV	#3, \$REG0	
6113	024030	104042				ERROR	42	
6114								
6115	024032	020427	000014		55:	CMP	R4, #14	
6116	024036	001414				BEQ	65	
6117	024040	012737	002531	001170		MOV	#2531, \$REG3	
6118	024046	012737	000014	001164		MOV	#14, \$REG1	
6119	024054	010437	001166			MOV	R4, \$REG2	
6120	024060	012737	000004	001162		MOV	#4, \$REG0	
6121	024066	104042				ERROR	42	
6122								
6123	024070	020527	000015		65:	CMP	R5, #15	
6124	024074	001414				BEQ	75	
6125	024076	012737	002530	001170		MOV	#2530, \$REG3	
6126	024104	012737	000015	001164		MOV	#15, \$REG1	
6127	024112	010537	001166			MOV	R5, \$REG2	
6128	024116	012737	000005	001162		MOV	#5, \$REG0	
6129	024124	104042				ERROR	42	
6130								
6131	024126	012737	140000	177776	75:	MOV	#140000, @#PSW	
6132	024134	010637	003050			MOV	R6, TMP1	
6133	024140	005037	177776			CLR	@#PSW	

```

6134 024144 022737 001060 003050
6135 024152 001415
6136 024154 012737 002527 001170
6137 024162 012737 001060 001164
6138 024170 013737 003050 001166
6139 024176 012737 000016 001162
6140 024204 104042
6141 024206 023727 003046 001070 85:
6142 024214 001415
6143 024216 012737 002526 001170
6144 024224 012737 001070 001164
6145 024232 013737 003046 001166
6146 024240 012737 000006 001162
6147 024246 104042
6148
6149 024250 95:
6150 024250 000412
6151
6152 024252 012777 000020 156554 155:
6153 024260 022626
6154 024262 005077 156546
6155 024266 012737 076700 001162
6156 024274 104043
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189

```

```

CMP #1060, TMP1
BEQ 85
MOV #2527, $REG3
MOV #1060, $REG1
MOV TMP1, $REG2
MOV #16, $REG0
ERROR 42
CMP TMP0, #1070
BEQ 95
MOV #2526, $REG3
MOV #1070, $REG1
MOV TMP0, $REG2
MOV #6, $REG0
ERROR 42
BR TST56 ;EXIT
MOV #PARDIS, @WCSST
CMP (SP)+, (SP)+
CLR @WCSST
MOV #XFCUDIS, $REG0
ERROR 43

```

```

;WCS WATCH DOG TIMED OUT
;UPON TRANSFERRING CONTROL FROM
;BASE MACHINE TO WCS MICROCODE.
;NORMALLY, AFTER UCODE EXECUTION CONTROL
;SHOULD BE RETURNED TO BASE MACHINE.

```

```

;*****
;*TEST 56 CHECK BLOCK MOVE OF FLOATING POINT REGISTERS TO WCS
; THIS TEST CHECKS THE BLOCK STORE WFP FUNCTION OF THE WCS.
; WHEN THIS FUNCTION IS INITIATED A BLOCK OF WARM
; FLOATING POINT REGISTERS GET TRANSFERRED FROM THE
; BASE MACHINE TO THE WCS. THE FOLLOWING WFP REGISTER
; ARE STORED IN THE GIVEN ORDER:
; FAC 0 5 ,FAC1 5 , FAC2 5 ,FAC3 5 ,FAC0 4 ,... FAC3 0
;
; THE WFP REGISTERS ARE INITIALIZED TO A KNOWN STATE,
; BY WRITING UNIQUE DATA PATTERNS IN THEM.
; THE MICRO-CODE IS LOADED IN WCS. USING A "XFC"
; DISPATCH INSTRUCTION CONTROL IS TRANSFERRED TO THE
; MICRO-CODE IN WCS.
;
; THE MICRO-CODE SETS UP IN "D" (BASE-MACHINE), THE
; ADDRESS (WCS) STARTING WHICH THE WFP REGISTERS
; SHOULD BE STORED IN THE WCS. THEN THE MICRO-CODE
; SELECTS THE WCS HOT BOX AND TRANSFERS CONTROL TO THE
; TMS-ROM WHICH ACTUALLY DOES THE STORE OF WFP
; REGISTERS. CONTROL IS RETURNED TO THE BASE MACHINE
;
; THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
; DATA FROM THE WFP REGISTERS WAS STORED CORRECTLY. IF NOT,
; AN ERROR IS REPORTED.
;
; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF
; AS A RESULT OF SOME FAULT-CONDITION, THE CONTROL

```



```

6190 ; IS NOT TRANSFERRED BACK TO THE BASE MACHINE
6191 ; THE WATCH-DOG WILL TIME OUT AND A TRAP TO
6192 ; 114 WILL OCCUR.
6193 ;
6194 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE =
6195 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY POINTER
6196 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
6197 ; CONTROL PATH USED BY MICRO-CODE EXECUTING
6198 ; FROM THE WCS. (NUA<9:0>, ARRAY ADDRESS MUX-
6199 ; A-PORT, BUS-U-MUX-B-PORT, BUS-U-LINES.
6200 ;*****
6201 024276 000004 TST56: SCOPE
6202 024300 012737 024506 003056 MOV #TST57,NXTST ;STARTING ADDRESS OF NEXT TEST
6203 ; INITIALIZE AND WRITE PATTERNS INTO WFP REGISTERS
6204 024306 012701 000030 MOV #24,R1 ;COUNT FOR 24 FP REGISTERS
6205 024312 005037 024336 CLR 25
6206 024316 012702 003174 MOV #FAC05W,R2 ;INITIALIZE POINTER TO WRITE MED CODE
6207 024322 012703 003224 MOV #PFAC05,R3 ;INITIALIZE POINTER TO THE PATTERN
6208 ; TO BE WRITTEN IN THE FLOATING
6209 ; POINT REGISTERS;
6210 024326 112237 024336 15: MOVB (R2)+,25 ;INSERT THE MED CODE FOR
6211 ; WRITING INTO FP REGISTER
6212 024332 112300 MOVB (R3)+,R0 ;WRITE THIS PATTERN
6213 ;
6214 024334 076600 MED ;WRITE PATTERN INTO THE
6215 024336 000000 25: .WORD 0 ;FP REGISTER USING THIS MED CODE
6216 ;
6217 024340 077106 SOB R1,15 ;WRITTEN ALL FP REGISTERS?
6218 ;
6219 ; LOAD UP WCS WITH MICRO-CODE, WHICH WILL STORE
6220 ; THE WFP REGISTERS IN WCS
6221 ;
6222 024342 004537 037472 JSR R5,LDWCS ;LOAD WCS WITH MICRO-CODE
6223 024346 006046 6046 ;MAIN CONTROL BLOCK LOADED HERE
6224 024350 061076 MFPLS ;FROM THIS MEMORY ADDRESS
6225 024352 006002 UDISP ;ENTRY POINT INTO WCS
6226 024354 012737 024470 000114 MOV #55,@#114 ;VECTOR FOR WCS WATCH DOG TIMER
6227 024362 012777 001000 156444 MOV #MAINT,@WCSST ;SETUP MAINT. BIT, TIMER WILL
6228 ; START WHEN WCS HOT BOX IS SELECTED
6229 ;
6230 024370 076700 XFCUDIS ;TRANSFER CONTROL TO WCS
6231 ; AND STORE FP REGISTERS IN WCS
6232 ; STARTING AT WCS LOCATION 1200
6233 ;
6234 ; RETURN HERE FROM WCS
6235 ;
6236 024372 042777 001000 156434 BIC #MAINT,@WCSST ;START OFF TIMER
6237 024400 012737 003344 000114 MOV #BADPAR,@#114 ;RESET BAD PARITY HANDLER
6238 ;
6239 ; CHECK IF THE FP REGISTERS WERE STORED CORRECTLY IN WCS
6240 ;
6241 024406 012704 000030 MOV #24,R4 ;SET COUNT
6242 ;
6243 024412 012700 003224 MOV #PFAC05,R0 ;INITIALIZE POINTER TO THE
6244 ; PATTERN THAT WAS WRITTEN
6245 024416 012701 001200 MOV #1200,R1 ;INITIALIZE WCSADR

```

```

6246
6247 024422 112002          35:  MOVB   (R0)+,R2      ;GOOD DATA PATTERN THAT
6248                                ;WAS SAVED
6249 024424 010177 156406    MOV    R1,@WCSAR      ;WCS READ
6250 024430 017703 156404    MOV    @WCSDR,R3     ;FROM THIS WCS LOCATION
6251 024434 020203          CMP    R2,R3         ;WAS THE CORRECT DATA
6252                                ;STORED IN WCS FROM
6253                                ;THE FP REGISTER?
6254 024436 001411          BEQ    45             ;YES
6255
6256 024440 010137 001162    MOV    R1,$REG0      ;WCS LOC THAT WAS READ
6257 024444 010237 001164    MOV    R2,$REG1      ;EXPECTED DATA
6258 024450 010337 001166    MOV    R3,$REG2      ;DATA ACTUALLY READ
6259 024454 010237 001170    MOV    R2,$REG3      ;FAC REGISTER NUMBER
6260 024460 104046          ERROR  46           ;CORRECT DATA WAS NOT STORED ON BLOCK
6261                                ;MOVE OF WFP REGISTERS TO WCS
6262
6263 024462 005201          45:  INC    R1             ;NEXT WCS ADDRESS
6264 024464 077422          SOB    R4,35         ;DONE?
6265 024466 000407          BR    65
6266
6267                                ; ENTER HERE IF THE WATCH-DOG TIMED OUT
6268
6269 024470 022626          55:  CMP    (SP)+,(SP)+   ;POP THE STACK
6270 024472 012777 000020 156334  MOV    #PAPDIS,@WCSST ;CLEAR PARITY ERROR (BIT 15)
6271 024500 005077 156330    CLR    @WCSST
6272 024504 104047          ERROR  47           ;WATCH-DOG TIMED OUT ON STORING OF
6273                                ;WFP REGISTERS IN WCS
6274 024506          65:                                ;EXIT
6275
6276                                ; *****
6277                                ; *TEST 57 CHECK BLOCK MOVE FROM WCS TO WFP REGISTERS
6278                                ; THIS TEST CHECKS THE BLOCK LOAD FP FUNCTION OF THE WCS.
6279                                ; WHEN THIS FUNCTION IS INITIATED WFP REGISTERS IN THE BASE
6280                                ; MACHINE ARE LOADED FROM THE WCS. THE FOLLOWING REGISTERS
6281                                ; ARE LOADED IN THE GIVEN ORDER:
6282                                ;
6283                                ; FAC0 5 , FAC1 5 , FAC2 5 , FAC3 5 , FAC0 4 , . . . . FAC3 0
6284                                ;
6285                                ; A BLOCK OF 24 LOCATIONS IN THE WCS IS INITIALIZED
6286                                ; TO A KNOWN STATE, BY WRITING UNIQUE DATA PATTERNS
6287                                ; IN THEM.
6288                                ;
6289                                ; THE MICRO-CODE IS LOADED INTO THE WCS. USING
6290                                ; A "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
6291                                ; TO THE MICRO-CODE IN THE WCS.
6292                                ;
6293                                ; THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE
6294                                ; ADDRESS(WCS) STARTING WHICH THE WORDS WILL BE
6295                                ; LOADED FROM THE WCS INTO THE WFP-REGS. THEN THIS
6296                                ; MICRO-CODE SELECTS THE WCS HOT BOX AND TRANSFERS
6297                                ; CONTROL TO THE TMS-ROM WHICH ACTUALLY DOES THE
6298                                ; LOAD OF WFP REGS. AFTER THE WFP REGS HAVE BEEN LOADED,
6299                                ; THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE.
6300
6301                                ; THE WFP REGS ARE THEN CHECKED TO MAKE SURE THAT THE

```



```
6302 ; CORRECT DATA WAS LOADED FROM THE WCS INTO THE
6303 ; WFP REGS. IF NOT AN ERROR IS REPORTED.
6304 ;
6305 ; THIS TEST USES THE WCS WATCH-DOG-TIMERS. IF AS
6306 ; A RESULT OF SOME FAULT CONDITION, THE CONTROL IS
6307 ; NOT TRANSFERRED BACK TO THE BASE MACHINE, THE
6308 ; WATCH-DOG WILL TIME OUT AND A TRAP TO 114 WILL
6309 ; OCCUR.
6310 ;
6311 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
6312 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINTER CIRCUIT.
6313 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION.
6314 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
6315 ; THE WCS. (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
6316 ; BUS-U-MUX-B-PORT, BUS-U LINES).
6317 ;
6318 ; *****
6319 024506 000004 TST57: SCOPE
6320 024510 012737 024776 003056 MOV #TST60,NXTST ; STARTING ADDRESS OF NEXT TEST
6321 ;
6322 ; CLEAR OUT THE WFP REGISTERS IN BASE MACHINE
6323 ;
6324 024516 012702 000030 MOV #24,R2 ; INITIALIZE COUNT
6325 024522 012701 003174 MOV #FAC05W,R1 ; INITIALIZE POINTER TO WRITE MED CODE
6326 024526 005037 024542 CLR 35 ; FOR WFP REGISTERS
6327 024532 112137 024542 25: MOVB (R1)+,35 ; INSERT MED CODE FOR WRITING INTO
6328 ; FP REGISTER
6329 024536 005000 CLR R0 ; WRITE A 0 PATTERN
6330 024540 076600 MED ; WRITE A ZERO IN THE FP
6331 024542 000000 35: .WORD 0 ; REGISTER POINTED BY THIS CODE
6332 ;
6333 024544 077206 SOB R2,25 ; DONE?
6334 ;
6335 ;
6336 ; LOAD THE WCS WITH MICRO-CODE, WHICH WILL TRANSFER DATA
6337 ; FROM WCS TO WFP REGISTERS IN BASE MACHINE
6338 ;
6339 024546 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO-CODE
6340 024552 006256 6256 ; MAIN CONTROL BLOCK LOADED HERE
6341 024554 061202 ML$FP ; FROM THIS MEMORY ADDRESS
6342 024556 006002 UDISP ; ENTRY POINT INTO WCS
6343 ;
6344 ; LOAD THE WCS WITH DATA PATTERNS STARTING AT WCS LOCATION 1100
6345 ;
6346 024560 012700 000030 MOV #24,R0 ; INITIALIZE COUNT
6347 024564 012701 001100 MOV #1100,R1 ; INITIALIZE THE WCS ADDRESS STARTING
6348 ; WHICH THE PATTERNS WILL BE WRITTEN
6349 024570 012702 003224 MOV #PFAC05,R2 ; INITIALIZE POINTER TO THE PATTERN
6350 ;
6351 024574 010177 156236 15: MOV R1,@WCSAR ; ADDRESS THE WCS LOCATION
6352 024600 112203 MOVB (R2)+,R3 ; GET THE PATTERN AND FORM A
6353 024602 052703 000300 BIS #300,R3 ; NEW ONE FOR LOADING INTO WCS
6354 024606 010377 156226 MOV R3,@WCSDP ; WRITE THE PATTERN INTO WCS
6355 024612 005201 INC R1 ; NEXT WCS ADDRESS
6356 024614 077011 SOB R0,15 ; DONE?
6357 ;
```





```

6414 024752 005205          65:  INC      R5              ;NEXT WCS LOCATION
6415 024754 077131          SOB      R1,4$          ;DONE CHECKING?
6416
6417 024756 000407          BR       TST60          ;;EXIT
6418
6419
6420 ;
6421 ; ENTER HERE, IF THE WCS WATCH DOG TIMER TIMED OUT
6422 ; ON TRANSFERRING CONTROL TO WCS. THE CONTROL WAS
6423 ; TRANSFERRED TO WCS TO LOAD THE WFP REGISTERS
6424 ; FROM THE WCS.
6425
6426 024760 022626          75:  CMP      (SP)+,(SP)+    ;POP THE STACK
6427 024762 012777 000020 156044 MOV      #PARDIS,@WCSST ;CLEAR OUT ERROR BIT (15)
6428 024770 005077 156040 CLR      @WCSST
6429 024774 104051          ERROR   51             ;WCS WATCH-DOG TIMED OUT
6430 ; WHEN CONTROL WAS PASSED TO
6431 ; WCS TO LOAD THE WFP
6432 ; REGISTERS
6433
6434
6435
6436
6437
6438
6439 ;*****
6440 ;*TEST 60 CHECK BLOCK MOVE OF C-SCRATCH PAD REGISTERS TO WCS
6441 ; THIS TEST CHECKS THE BLOCK STORE CSP FUNCTION OF THE WCS.
6442 ; WHEN THIS FUNCTION IS INITIATED A BLOCK OF C-
6443 ; SCRATCH PAD REGISTERS GET TRANSFERRED FROM THE
6444 ; BASE MACHINE TO THE WCS. THE FOLLOWING CSP REGISTER
6445 ; ARE STORED IN THE GIVEN ORDER:
6446 ; CSP 0 ,CSP 1 ,CSP 2 ,... CSP 11
6447 ;
6448 ; THE CSP REGISTERS ARE INITIALIZED TO A KNOWN STATE,
6449 ; BY WRITING UNIQUE DATA PATTERNS IN THEM.
6450 ;
6451 ; THE MICRO-CODE IS LOADED IN WCS. USING A "XFC"
6452 ; DISPATCH INSTRUCTION CONTROL IS TRANSFERRED TO THE
6453 ; MICRO-CODE IN WCS.
6454 ;
6455 ; THE MICRO-CODE SETS UP IN "D" (BASE-MACHINE), THE
6456 ; ADDRESS (WCS) STARTING WHICH THE CSP REGISTERS
6457 ; SHOULD BE STORED IN THE WCS. THEN THE MICRO-CODE
6458 ; SELECTS THE WCS HOT BOX AND TRANSFERS CONTROL TO THE
6459 ; TMS-ROM WHICH ACTUALLY DOES THE STORE OF CSP
6460 ; REGISTERS. CONTROL IS RETURNED TO THE BASE MACHINE.
6461 ;
6462 ; THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
6463 ; DATA FROM THE CSP REGISTERS WAS STORED CORRECTLY. IF NOT,
6464 ; AN ERROR IS REPORTED.
6465 ;
6466 ; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF
6467 ; AS A RESULT OF SOME FAULT-CONDITION, THE CONTROL
6468 ; IS NOT TRANSFERRED BACK TO THE BASE MACHINE
6469 ; THE WATCH-DOG WILL TIME OUT AND A TRAP TO

```

```

6470 ; 114 WILL OCCUR.
6471 ;
6472 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE =
6473 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY POINTER
6474 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
6475 ; CONTROL PATH USED BY MICRO-CODE EXECUTING
6476 ; FROM THE WCS. (NUA<9:0>, ARRAY ADDRESS MUX-
6477 ; A-PORT, BUS-U-MUX-B-PORT, BUS-U-LINES.
6478 ;
6479 ;*****
6480 024776 000004 TST60: SCOPE
6481 025000 012737 025216 003056 MOV #TST61,NXTST ;STARTING ADDRESS OF NEXT TEST
6482 ; INITIALIZE AND WRITE PATTERNS INTO CSP REGISTERS
6483 025006 012701 000014 MOV #12.,R1 ;COUNT FOR 24 FP REGISTERS
6484 025012 005037 025036 CLR 25
6485 025016 012702 003254 MOV #CSP00W,R2 ;INITIALIZE POINTER TO WRITE MED CODE
6486 025022 012703 000200 MOV #200,R3 ;INITIALIZE PATTERN TO BE WRITTEN
6487 025026 112237 025036 15: MOVB (R2)+,25 ;INSERT THE MED CODE FOR
6488 ; WRITING INTO CSP REGISTER
6489 025032 010300 MOV R3,R0 ;WRITE THIS PATTERN
6490 ;
6491 025034 076600 MED ;WRITE PATTERN INTO THE
6492 025036 000000 25: .WORD 0 ;CSP REGISTER USING THIS MED CODE
6493 ;
6494 025040 005203 INC R3
6495 025042 077107 SOB R1,15 ;WRITTEN ALL CSP REGISTERS?
6496 ;
6497 ; LOAD UP WCS WITH MICRO-CODE, WHICH WILL STORE
6498 ; THE CSP REGISTERS IN WCS
6499 ;
6500 025044 004537 037472 JSR R5,LDWCS ;LOAD WCS WITH MICRO-CODE
6501 025050 006066 6066 ;MAIN CONTROL BLOCK LOADED HERE
6502 025052 061306 MCCLS ;FROM THIS MEMORY ADDRESS.
6503 025054 006002 UDISP ;ENTRY POINT INTO WCS
6504 025056 012737 025200 000114 MOV #55,@#114 ;VECTOR FOR WCS WATCH DOG TIMER
6505 025064 012777 001000 155742 MOV #MAINT,@WCSST ;SETUP MAINT. BIT, TIMER WILL
6506 ; START WHEN WCS HOT BOX IS SELECTED
6507 ;
6508 025072 076700 XFCUDIS ;TRANSFER CONTROL TO WCS
6509 ; AND STORE CSP REGISTERS IN WCS
6510 ; STARTING AT WCS LOCATION 1200
6511 ;
6512 ; RETURN HERE FROM WCS
6513 ;
6514 025074 042777 001000 155732 BIC #MAINT,@WCSST ;START OFF TIMER
6515 025102 012737 003344 000114 MOV #BADPAR,@#114 ;RESET BAD PARITY HANDLER
6516 ;
6517 ; CHECK IF THE CSP REGISTERS WERE STORED CORRECTLY IN WCS
6518 ;
6519 025110 012704 000014 MOV #12.,R4 ;SET COUNT
6520 025114 005005 CLR R5 ;INITIALIZE CSP REGISTER NUMBER
6521 ;
6522 025116 012700 000200 MOV #200,R0 ;INITIALIZE PATTERN TTHAT WAS WRITTEN
6523 025122 012701 001040 MOV #1040,R1 ;INITIALIZE WCSADP
6524 ;
6525 025126 010002 35: MOV R0,R2 ;GOOD DATA PATTERN THAT

```



```

6526 ; WAS SAVED
6527 025130 010177 155702 MOV R1, @WCSAR ; WCS READ
6528 025134 017703 155700 MOV @WCSDR, R3 ; FROM THIS WCS LOCATION
6529 025140 020203 CMP R2, R3 ; WAS THE CORRECT DATA
6530 ; STORED IN WCS FROM
6531 ; THE FP REGISTER?
6532 025142 001411 BEQ 45 ; YES
6533
6534 025144 010137 001162 MOV R1, $REG0 ; WCS LOC THAT WAS READ
6535 025150 010237 001164 MOV R2, $REG1 ; EXPECTED DATA
6536 025154 010337 001166 MOV R3, $REG2 ; DATA ACTUALLY READ
6537 025160 010537 001170 MOV R5, $REG3 ; CSP REGISTER NUMBER
6538 025164 104052 ERROR 52 ; CORRECT DATA WAS NOT STORED ON BLOCK
6539 ; MOVE OF CSP REGISTERS TO WCS
6540 025166 005200 45: INC R0 ; NEXT DATA PATTERN
6541 025170 005205 INC R5
6542 025172 005201 INC R1 ; NEXT WCS ADDRESS
6543
6544 025174 077424 SOB R4, 35 ; DONE?
6545 025176 000407 BR TST61 ; EXIT
6546
6547 ; ENTER HERE IF THE WATCH-DOG TIMED OUT
6548

```

```

6549 025200 022626 55: CMP (SP)+, (SP)+ ; POP THE STACK
6550 025202 012777 000020 155624 MOV #PAPDIS, @WCSST ; CLEAR PARITY ERROR (BIT 15)
6551 025210 005077 155620 CLR @WCSST
6552 025214 104053 ERROR 53 ; WATCH-DOG TIMED OUT ON STORING OF
6553 ; CSP REGISTERS IN WCS
6554
6555

```

```

6556 ; *****
6557 ; *TEST 61 CHECK BLOCK MOVE FROM WCS TO C-SCRATCH PAD, CSP
6558 ; THIS TEST CHECKS THE BLOCK LOAD CSP FUNCTION OF THE WCS.
6559 ; WHEN THIS FUNCTION IS INITIATED CSP IN THE BASE
6560 ; MACHINE ARE LOADED FROM THE WCS. THE FOLLOWING REGISTER
6561 ; ARE LOADED IN THE GIVEN ORDER:
6562 ;
6563 ; CSP 0 , CSP 1 , CSP 2 ..... CSP 11
6564 ;
6565 ; A BLOCK OF 12 LOCATIONS IN THE WCS IS INITIALIZED
6566 ; TO A KNOWN STATE, BY WRITING UNIQUE DATA PATTERN
6567 ; IN THEM.
6568 ;
6569 ; THE MICRO-CODE IS LOADED INTO THE WCS. USING
6570 ; A "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
6571 ; TO THE MICRO-CODE IN THE WCS.
6572 ;
6573 ; THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE
6574 ; ADDRESS(WCS) STARTING WHICH THE WORDS WILL BE
6575 ; LOADED FROM THE WCS INTO THE CSP. THEN THIS
6576 ; MICRO-CODE SELECTS THE WCS HOT BOX AND TRANSFERS
6577 ; CONTROL TO THE TMS-ROM WHICH ACTUALLY DOES THE
6578 ; LOAD OF CSP. AFTER THE CSP HAVE BEEN LOADED,
6579 ; THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE
6580 ;
6581 ; THE CSP ARE THEN CHECKED TO MAKE SURE THAT THE

```

6582  
6583  
6584  
6585  
6586  
6587  
6588  
6589  
6590  
6591  
6592  
6593  
6594  
6595  
6596  
6597  
6598  
6599  
6600  
6601  
6602  
6603  
6604  
6605  
6606  
6607  
6608  
6609  
6610  
6611  
6612  
6613  
6614  
6615  
6616  
6617  
6618  
6619  
6620  
6621  
6622  
6623  
6624  
6625  
6626  
6627  
6628  
6629  
6630  
6631  
6632  
6633  
6634  
6635  
6636  
6637

025216 000004  
025220 012737 025474 003056  
  
025226 012702 000014  
025232 012701 003254  
025236 005037 025252  
  
025242 112137 025252  
  
025246 005000  
025250 076600  
025252 000000  
  
025254 077206  
  
025256 004537 037472  
025262 006476  
025264 061412  
025266 006002  
  
025270 012700 000014  
025274 012701 001020  
  
025300 012702 000100  
  
025304 010177 15. 26  
025310 010277 155524  
025314 005202  
025316 005201

; CORRECT DATA WAS LOADED FROM THE WCS INTO THE  
; CSP. IF NOT AN ERROR IS REPORTED.  
  
; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
; A RESULT OF SOME FAULT CONDITION, THE CONTROL IS  
; NOT TRANSFERRED BACK TO THE BASE MACHINE, THE  
; WATCH-DOG WILL TIME OUT AND A TRAP TO 114 WILL  
; OCCUR.  
  
; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:  
; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINTER CIRCUIT.  
; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION.  
; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
; THE WCS. (NUR<9:0>, ARRAY ADDRESS MUX-A-PORT,  
; BUS-U-MUX-B-PORT, BUS-U LINES).  
  
; \*\*\*\*\*  
TST61: SCOPE  
MOV #TST62,NXTST ; STARTING ADDRESS OF NEXT TEST  
  
; CLEAR OUT CSP 0 TO CSP 13 IN BASE MACHINE  
  
MOV #12,R2 ; INITIALIZE COUNT  
MOV #CSP00W,R1 ; INITIALIZE POINTER TO WRITE MED CODE  
CLR 35 ; FOR CSP'S  
  
MOVB (R1)+,35 ; INSERT MED CODE FOR WRITING INTO  
; CSP  
  
CLR R0 ; WRITE A 0 PATTERN  
MED ; WRITE A 0 IN THE CSP  
WORD 0 ; POINTED BY THIS CODE  
  
SOB R2,25 ; DONE?  
  
; LOAD THE WCS WITH MICRO-CODE WHICH WILL TRANSFER DATA  
; FROM WCS TO CSP REGISTERS IN THE BASE MACHINE  
  
JSR R5,LOWCS ; LOAD WCS WITH MICRO-CODE  
6476 ; MAIN CONTROL BLOCK LOADED HERE  
MLSCS ; FROM THIS MEMORY ADDRESS.  
UDISP ; ENTRY POINT INTO WCS  
  
; LOAD THE WCS WITH DATA PATTERNS STARTING AT WCS LOCATION 1020  
  
MOV #12,R0 ; INITIALIZE COUNT  
MOV #1020,R1 ; INITIALIZE THE WCS ADDRESS  
; STARTING WHICH THE PATTERNS WILL BE WRITTEN  
MOV #100,R2 ; INITIALIZE DATA PATTERN TO BE WRITTEN  
  
15. MOV R1,@WCSAP ; ADDRESS THE WCS LOCATION  
MOV R2,@WCSOP ; WRITE THE PATTERN INTO WCS LOC  
INC R2  
INC R1 ; NEXT WCS LOCATION



```

6638 025320 077007          SOB      R0, 15          ; DONE?
6639
6640 025322 012737 025456 000114  MOV      #75, @#114      ; SETUP VECTOR FOR WCS WATCH-DOG
6641 025330 012777 001000 155476  MOV      #MAINT, @WCSST  ; SET MAINT BIT, TIMER WILL
6642                                     ; START WHEN WCS IS SELECTED
6643 025336 076700          XFCUDIS  ; TRANSFER CONTROL TO WCS AND
6644                                     ; LOAD CSP REGISTERS FROM
6645                                     ; WCS (STARTING AT LOC 1040)
6646
6647                                     ; RETURN HERE FROM WCS
6648
6649 025340 042777 001000 155466  BIC      #MAINT, @WCSST  ; SHUT OFF TIMER
6650 025346 012737 003344 000114  MOV      #BADPAR, @#114  ; RESET BAD PARITY HANDLER
6651
6652                                     ; CHECK IF THE CSP REGISTERS WERE LOADED CORRECTLY
6653                                     ; FROM WCS
6654
6655 025354 012701 000014  MOV      #12, R1          ; COUNT FOR 12 REGISTERS
6656 025360 012702 003254  MOV      #CSP00W, R2      ; INITIALIZE POINTER TO MED CODES
6657 025364 012704 000100  MOV      #100, R4         ; INITIALIZE DATA PATTERN
6658 025370 005037 025416  CLR      59
6659 025374 005003  CLR      R3
6660 025376 012705 001040  MOV      #1040, R5        ; START OF WCS LOC FROM WHICH
6661                                     ; DATA WAS LOADED INTO CSP
6662
6663
6664 025402 112237 025416 49  MOV      (R2)+, 59        ; GET THE MED CODE AND FORM
6665 025406 042737 000200 025416  BIC      #200, 59        ; THE RIGHT CODE FOR READING
6666
6667 025414 076600  MED      WORD 0          ; READ THE WFP REGISTER POINTED
6668 025416 000000 59                                     ; BY THIS MED CODE
6669
6670 025420 020004  CMP      R0, R4          ; CHECK IF THE PATTERN LOADED INTO
6671                                     ; CSP REGISTER IS AS EXPECTED
6672 025422 001411  BEQ      69             ; IF YES, OK
6673 025424 010337 001162  MOV      R3, $REG0       ; SAVE CSP REGISTER NUMBER WHICH
6674                                     ; DID NOT HAVE THE CORRECT DATA
6675 025430 010437 001164  MOV      R4, $REG1       ; SAVE EXPECTED DATA
6676 025434 010037 001166  MOV      R0, $REG2       ; SAVE DATA RECEIVED
6677 025440 010537 001170  MOV      R5, $REG3       ; SAVE WCS LOC FROM WHERE DATA
6678                                     ; WAS READ
6679 025444 104054  ERROR   54          ; CORRECT DATA WAS NOT FOUND
6680                                     ; IN THE BASE MACHINE CSP REGISTERS
6681                                     ; AFTER HAVING DONE A
6682                                     ; BLOCK MOVE (LOAD) OF CSP
6683                                     ; FROM THE WCS. THE CSP REGISTER
6684                                     ; NUMBER IS GIVEN IN THE ERROR MESSAGE.
6685                                     ; THE MESSAGE ALSO GIVES THE WCS LOC
6686                                     ; FROM WHICH THE CSP WAS LOADED.
6687                                     ; THE EXPECTED AND RECEIVED DATA
6688                                     ; IN CSP.
6689
6690 025446 005205 69  INC      R5          ; NEXT WCS LOCATION
6691 025450 005204  INC      R4          ; NEXT DATA PATTERN
6692 025452 077125  SOB      R1, 49        ; DONE?
6693

```

6694 025454 000407

BR TST62 ; ;EXIT

6695

6696

6697

6698

6699

6700 025456 022626

75:

CMP (SP)+,(SP)+ ;POP THE STACK

6701 025460 012777 000020 155346

MOV #PARDIS,@WCSST ;CLEAR OUT ERROR BIT (15)

6702 025466 005077 155342

CLR @WCSST

6703

6704 025472 104055

ERROR 55 ;WCS WATCH-DOG TIMED OUT

6705

6706

6707

6708

6709 025474

85:

6710

6711

6712

6713

6714

6715

6716

6717

6718

6719

6720

6721

6722

6723

6724

6725

6726

6727

6728

6729

6730

6731

6732

6733

6734

6735

6736

6737

6738

6739

6740

6741

6742

6743

6744

6745

6746

6747

6748

6749

```

;*****
; *TEST 62 CHECK BLOCK MOVE OF USER-SCRATCH PAD REGISTERS TO WCS
; THIS TEST CHECKS THE BLOCK STORE USP FUNCTION OF THE WCS.
; WHEN THIS FUNCTION IS INITIATED A BLOCK OF USER-
; SCRATCH PAD REGISTERS GET TRANSFERRED FROM THE
; BASE MACHINE TO THE WCS. THE FOLLOWING USP REGISTER
; ARE STORED IN THE GIVEN ORDER:
; WCSA 0 ,WCSB 0 ,WCSB 1
;
; THE USP REGISTERS ARE INITIALIZED TO A KNOWN STATE,
; BY WRITING UNIQUE DATA PATTERNS IN THEM.
;
; THE MICRO-CODE IS LOADED IN WCS. USING A "XFC"
; DISPATCH INSTRUCTION CONTROL IS TRANSFERRED TO THE
; MICRO-CODE IN WCS.
;
; THE MICRO-CODE SETS UP IN "D" (BASE-MACHINE), THE
; ADDRESS (WCS) STARTING WHICH THE USP REGISTERS
; SHOULD BE STORED IN THE WCS. THEN THE MICRO-CODE
; SELECTS THE WCS HOT BOX AND TRANSFERS CONTROL TO THE
; TMS-ROM WHICH ACTUALLY DOES THE STORE OF USP
; REGISTERS. CONTROL IS RETURNED TO THE BASE MACHINE.
;
; THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
; DATA FROM THE USP REGISTERS WAS STORED CORRECTLY. IF NOT,
; AN ERROR IS REPORTED.
;
; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF
; AS A RESULT OF SOME FAULT-CONDITION, THE CONTROL
; IS NOT TRANSFERRED BACK TO THE BASE MACHINE
; THE WATCH-DOG WILL TIME OUT AND A TRAP TO
; 114 WILL OCCUR.
;
; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE =
; TMS-CROM SECTION OF WCS INCLUDING ENTRY POINTER
; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
; CONTROL PATH USED BY MICRO-CODE EXECUTING
; FROM THE WCS. (NUA<9: 0>), ARRAY ADDRESS MUX-
; A-PORT, BUS-U-MUX-B-PORT, BUS-U-LINES

```



```

6750
6751
6752 025474 000004
6753 025476 012737 025700 003056
6754
6755
6756 025504 012701 000003
6757 025510 012702 003270
6758 025514 011237 025524
6759
6760 025520 012200
6761
6762 025522 076600
6763 025524 000000
6764
6765 025526 077106
6766
6767
6768
6769
6770 025530 004537 037472
6771 025534 006106
6772 025536 061516
6773 025540 006002
6774 025542 012737 025662 000114
6775 025550 012777 001000 155256
6776
6777
6778 025556 076700
6779
6780
6781
6782
6783
6784 025560 042777 001000 155246
6785 025566 012737 003344 000114
6786
6787
6788
6789 025574 012704 000003
6790 025600 012700 003270
6791 025604 012701 001010
6792 025610 012002
6793 025612 010177 155220
6794 025616 017703 155213
6795 025622 020203
6796
6797
6798 025624 001413
6799
6800 025626 010137 001162
6801 025632 010237 001164
6802 025636 010337 001166
6803 025642 042702 177400
6804 025646 010237 001170
6805 025652 104056

```

```

;*****
TST62: SCOPE
MOV #TST63,NXTST ;STARTING ADDRESS OF NEXT TEST
; INITIALIZE AND WRITE PATTERNS INTO USER SCRATCH PAD (USP)
; REGISTERS
MOV #3,R1 ;COUNT FOR 3 USP REGISTERS
MOV #WCSAOW,R2 ;INITIALIZE POINTER TO WRITE MED CODE
15: MOV (R2),25 ;INSERT THE MED CODE FOR
;WRITING INTO USP REGISTER
MOV (R2)+,R0 ;FORM THE PATTERN
;
MED ;WRITE PATTERN INTO THE
;WORD 0 ;USP REGISTER USING THIS MED CODE
25: SOB R1,15 ;WRITTEN ALL USP REGISTERS?
;
; LOAD UP WCS WITH MICRO-CODE, WHICH WILL STORE
; THE USP REGISTERS IN WCS
;
JSR R5,LDWCS ;LOAD WCS WITH MICRO-CODE
6106 ;STARTING AT THIS WCS ADDRESS
MWSLS
UDISP ;FROM THIS MEMORY ADDRESS.
MOV #55,@#114 ;VECTOR FOR WCS WATCH DOG TIMER
MOV #MAINT,@WCSST ;SETUP MAINT. BIT, TIMER WILL
;START WHEN WCS HOT BOX IS SELECTED
;
XFCUDIS ;TRANSFER CONTROL TO WCS
;AND STORE USP REGISTERS IN WCS
;STARTING AT WCS LOCATION 1200
;
;RETURN HERE FROM WCS
;
BIC #MAINT,@WCSST ;START OFF TIMER
MOV #BADPAR,@#114 ;RESET BAD PARITY HANDLER
;
CHECK IF THE USP REGISTERS WERE STORED CORRECTLY IN WCS
;
MOV #3,R4 ;SET COUNT
MOV #WCSAOW,R0 ;INITIALIZE POINTER TO MED WRITE-CODES
MOV #1010,R1 ;INITIALIZE WCSADR
35: MOV (R0)+,R2 ;GOOD DATA PATTERN THAT
MOV R1,@WCSAR ;WCS READ
MOV @WCSDR,R3 ;FROM THIS WCS LOCATION
CMP R2,R3 ;WAS THE CORRECT DATA
;STORED IN WCS FROM
;THE FP REGISTER?
;YES
BEQ 45
;
MOV R1,$REG0 ;WCS LOC THAT WAS READ
MOV R2,$REG1 ;EXPECTED DATA
MOV R3,$REG2 ;DATA ACTUALLY READ
BIC #177400,R2
MOV R2,$REG3 ;USP REGISTER (MED READ) CODE
ERROR 56 ;CORRECT DATA WAS NOT STORED ON BLOCK

```

```

6806 ;MOVE OF USP REGISTERS TO WCS
6807 025654 005201 45: INC R1 ;NEXT WCS ADDRESS
6808
6809 025656 077424 SOB R4,35 ;DONE?
6810 025660 000407 BR 65
6811
6812 ; ENTER HERE IF THE WATCH-DOG TIMED OUT
6813
6814 025662 022626 55: CMP (SP)+,(SP)+ ;POP THE STACK
6815 025664 012777 000020 155142 MOV #PARDIS,@WCSST ;CLEAR PARITY ERROR (BIT 15)
6816 025672 005077 155136 CLR @WCSST
6817 025676 104057 ERROR 57 ;WATCH-DOG TIMED OUT ON STORING OF
6818 ;USP REGISTERS IN WCS
6819 025700 65: ;EXIT
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861

```

\*\*\*\*\*

\*TEST 63 CHECK BLOCK MOVE FROM WCS TO USER-SCRATCH PAD, USP  
THIS TEST CHECKS THE BLOCK LOAD USP FUNCTION OF THE WCS.  
WHEN THIS FUNCTION IS INITIATED USP IN THE BASE  
MACHINE ARE LOADED FROM THE WCS. THE FOLLOWING REGISTER  
ARE LOADED IN THE GIVEN ORDER:

WCSA 0 ,WCSB 0 ,WCSB 1

A BLOCK OF 12 LOCATIONS IN THE WCS IS INITIALIZED  
TO A KNOWN STATE, BY WRITING UNIQUE DATA PATTERN  
IN THEM.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
A "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE  
ADDRESS(WCS) STARTING WHICH THE WORDS WILL BE  
LOADED FROM THE WCS INTO THE USP. THEN THIS  
MICRO-CODE SELECTS THE WCS HOT BOX AND TRANSFERS  
CONTROL TO THE TMS-ROM WHICH ACTUALLY DOES THE  
LOAD OF USP. AFTER THE USP HAVE BEEN LOADED,  
THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE

THE USP ARE THEN CHECKED TO MAKE SURE THAT THE  
CORRECT DATA WAS LOADED FROM THE WCS INTO THE  
USP. IF NOT AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL IS  
NOT TRANSFERRED BACK TO THE BASE MACHINE, THE  
WATCH-DOG WILL TIME OUT AND A TRAP TO 114 WILL  
OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:  
TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINTER CIRCUIT.  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION.  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS. (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT.



```
6862 ; BUS-U-MUX-B-PORT, BUS-U LINES).
6863 ;
6864 ; *****
6865 025700 000004 TST63: SCOPE
6866 025702 012737 026142 003056 MOV #TST64, NXTST ; STARTING ADDRESS OF NEXT TEST
6867 ;
6868 ; CLEAR OUT WCSA 0 TO WCSB 1 IN BASE MACHINE
6869 ;
6870 025710 012702 000003 MOV #3, R2 ; INITIALIZE COUNT
6871 025714 012701 003270 MOV #WCSA0W, R1 ; INITIALIZE POINTER TO WRITE MED CODE
6872 ;
6873 025720 012137 025730 25: MOV (R1)+, 35 ; INSERT MED CODE FOR WRITING INTO
6874 ; USP
6875 025724 005000 CLR R0 ; WRITE A 0 PATTERN
6876 025726 076600 MED ; WRITE A 0 IN THE USP
6877 025730 000000 35: .WORD 0 ; POINTED BY THIS CODE
6878 ;
6879 025732 077206 SOB R2, 25 ; DONE?
6880 ;
6881 ;
6882 ; LOAD THE WCS WITH MICRO-CODE WHICH WILL TRANSFER DATA
6883 ; FROM WCS TO USP REGISTERS IN THE BASE MACHINE
6884 ;
6885 025734 004537 037472 JSR R5, LDWCS ; LOAD WCS WITH MICRO-CODE
6886 025740 006716 6716 ; STARTING AT THIS WCS ADDRESS
6887 025742 061622 MLSWS ; FROM THIS MEMORY ADDRESS.
6888 025744 006002 UDISP ; THIS IS THE ENTRY POINT INTO WCS
6889 ;
6890 ; LOAD THE WCS WITH DATA PATTERNS STARTING AT WCS LOCATION 1004
6891 ;
6892 025746 012700 000003 MOV #3, R0 ; INITIALIZE COUNT
6893 025752 012701 001004 MOV #1004, R1 ; INITIALIZE THE WCS ADDRESS
6894 ; STARTING WHICH THE PATTERNS WILL BE WRITTEN
6895 025756 012702 003270 MOV #WCSA0W, R2 ; INITIALIZE POINTER TO MED WRITE CODES
6896 ;
6897 025762 010177 155050 15: MOV R1, @WCSAR ; ADDRESS THE WCS LOCATION
6898 025766 012277 155046 MOV (R2)+, @WCS0R ; WRITE THE PATTERN INTO WCS LOC
6899 025772 042777 177600 155040 BIC #177600, @WCS0R
6900 026000 005201 INC R1 ; NEXT WCS LOCATION
6901 026002 077011 SOB R0, 15 ; DONE?
6902 ;
6903 026004 012737 026124 000114 MOV #75, @#114 ; SETUP VECTOR FOR WCS WATCH-DOG
6904 026012 012777 001000 155014 MOV #MAINT, @WCSST ; SET MAINT BIT, TIMER WILL
6905 ; START WHEN WCS IS SELECTED
6906 026020 076700 XFCUDIS ; TRANSFER CONTROL TO WCS AND
6907 ; LOAD USP REGISTERS FROM
6908 ; WCS (STARTING AT LOC 1004)
6909 ;
6910 ; RETURN HERE FROM WCS
6911 ;
6912 026022 042777 001000 155004 BIC #MAINT, @WCSST ; SHUT OFF TIMER
6913 026030 012737 003344 000114 MOV #BADPAR, @#114 ; RESET BAD PARITY HANDLER
6914 ;
6915 ; CHECK IF THE USP REGISTERS WERE LOADED CORRECTLY
6916 ; FROM WCS
6917 ;
```

```

6918 026036 012701 000003      MOV      #3, R1          ;COUNT FOR 3 REGISTERS
6919 026042 012702 003270      MOV      #WCSAOW, R2    ;INITIALIZE POINTER TO MED CODES
6920 026046 012705 001004      MOV      #1004, R5      ;START OF WCS LOC FROM WHICH
6921                                     ;DATA WAS LOADED INTO USP
6922
6923
6924 026052 012204      45:     MOV      (R2)+, R4          ;GET THE MED CODE AND FORM
6925 026054 042704 177600      BIC      #177600, R4    ;THE RIGHT CODE FOR READING
6926 026060 010437 026066      MOV      R4, 55
6927 026064 076600      MED
6928 026066 000000      55:     .WORD   0          ;READ THE WFP REGISTER POINTED
6929                                     ;BY THIS MED CODE
6930 026070 020004      CMP      R0, R4        ;CHECK IF THE PATTERN LOADED INTO
6931                                     ;USP REGISTER IS AS EXPECTED
6932 026072 001411      BEQ      65            ;IF YES, OK
6933                                     ;SAVE USP REG (RD MED CODE)
6934 026074 010437 001170      MOV      R4, $REG3     ;WHICH DID NOT HAVE THE CORRECT DATA
6935 026100 010437 001164      MOV      R4, $REG1     ;SAVE EXPECTED DATA
6936 026104 010037 001166      MOV      R0, $REG2     ;SAVE DATA RECEIVED
6937 026110 010537 001162      MOV      R5, $REG0     ;SAVE WCS LOC FROM WHERE DATA
6938                                     ;WAS READ
6939 026114 104060      ERROR   60            ;CORRECT DATA WAS NOT FOUND
6940                                     ;IN THE BASE
6941                                     ;MACHINE, AFTER HAVING DONE A
6942                                     ;BLOCK MOVE (LOAD) OF USP
6943                                     ;FROM THE WCS. THE USP REGISTER
6944                                     ;NUMBER IS GIVEN IN THE ERROR MESSAGE.
6945                                     ;THE MESSAGE ALSO GIVES THE WCS LOC
6946                                     ;FROM WHICH THE USP WAS LOADED,
6947                                     ;THE EXPECTED AND RECEIVED DATA
6948                                     ;IN USP.
6949
6950 026116 005205      65:     INC      R5            ;NEXT WCS LOCATION
6951 026120 077124      SOB      R1, 45        ;DONE?
6952
6953 026122 000407      BR      TST64          ;EXIT
6954
6955 ;
6956 ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT ON
6957 ; TRANSFERRING CONTROL TO WCS. THE CONTROL WAS TRANSFERRED
6958 ; TO WCS TO LOAD THE USP REGISTERS FROM THE WCS.
6959 026124 022626      75:     CMP      (SP)+, (SP)+    ;POP THE STACK
6960 026126 012777 000020 154700      MOV      #PARDIS, @WCSST ;CLEAR OUT ERROR BIT (15)
6961 026134 005077 154674      CLR      @WCSST
6962
6963 026140 104061      ERROR   61            ;WCS WATCH-DOG TIMED OUT
6964                                     ;WHEN CONTROL WAS PASSED
6965                                     ;TO WCS TO LOAD THE USP
6966                                     ;REGISTERS.
6967
6968
6969 ;*****
6970 ;*TEST 64 CHECK BLOCK STORE OF LOAD-WRITE-2 TO WCS
6971 ;
6972 ; THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:
6973

```



```

6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018 026142 000004
7019 026144 012737 026400 003056
7020
7021 026152 004537 037472
7022 026156 006560
7023 026160 057454
7024 026162 006002
7025
7026 101010
7027 010101
7028 001111
7029

```

```

; 'LOAD WRITE 2' -> WCS.ARRAY
;
; WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
; GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.
;
; THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:
;
; R3, R4 (SETUP VIA MICROCODE, ANY INTERNAL
; REGISTERS COULD BE CHOSEN)
;
; THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING
; UNIQUE DATA PATTERNS IN THEM.
;
; THE MICRO-CODE IS LOADED INTO THE WCS. USING
; AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
; TO THE MICRO-CODE IN THE WCS.
;
; FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),
; THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS
; SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS
; THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
; WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE
; REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS
; CONTROL TO THE BASE MACHINE.
;
; THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
; DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,
; AN ERROR IS REPORTED.
;
; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
; A RESULT OF SOME FAULT CONDITION, THE CONTROL
; IS NOT TRANSFERRED BACK TO THE BASE MACHINE,
; THE WATCH DOG WILL TIME OUT AND A TRAP TO
; 114 WILL OCCUR.
;
; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
;
; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
; BUS-U-MUX-B PORT, BUS-U-LINES).
;
; *****
; TST64: SCOPE
; MOV #TST65,NXTST ; STARTING ADDRESS OF NEXT TEST
;
; JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
; 6560 ; CONTROL BLOCK LOADED HERE
; MLDWR2 ; FROM THIS MEMORY LOCATION
; UDISP ; ENTRY POINT IN WCS
;
; PAT. A= 101010 ; DATA PATTERN FOR TEST
; PAT. B= 010101 ;
; ADDR. 1=1111 ;
;

```

```

7030 026164 012703 101010      MOV      #PAT. A, R3      ; PATTERN # 1
7031 026170 012704 010101      MOV      #PAT. B, R4      ; PATTERN # 2
7032 026174 012702 001111      MOV      #ADDR. 1, R2     ; R2 IS LOCAL STORE ADDRESS HOLDER
7033 026200 005000              CLR      R0               ; ZAP ELSE
7034 026202 005001              CLR      R1
7035 026204 005005              CLR      R5
7036
7037 026206 012737 026362 000114  MOV      #39$, @#114      ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7038 026214 012777 001000 154612  MOV      #MAINT, @WCSST   ; SET UP MAINT BIT, TIMER WILL START WHEN
7039                                ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7040
7041 026222 076700              XFCUDIS                   ; TRANSFER CONTROL TO WCS
7042 026224                      15:                          ; SHOULD RETURN FROM WCS TO THIS POINT
7043
7044 026224 042777 001000 154602  BIC      #MAINT, @WCSST   ; SHUT OFF THE TIMER
7045 026232 012737 003344 000114  MOV      #BADPAR, @#114   ; RESET UNEXPECTED PARITY HANDLER
7046
7047                                ; CHECK FIRST DATA
7048 026240 012777 001111 154570  MOV      #ADDR. 1, @WCSAR  ; ADDRESS THE WCS AT EXPEC'D LOC
7049 026246 027727 154566 101010  CMP      @WCSDR, #PAT. A  ; IS IT VALUE THAR WAS IN R3 ??
7050 026254 001415              BEQ      115              ; BR IF OK
7051 026256 012737 000003 001162  MOV      #3, $REG0
7052 026264 012737 101010 001164  MOV      #PAT. A, $REG1
7053 026272 017737 154542 001166  MOV      @WCSDR, $REG2
7054 026300 017737 154532 001170  MOV      @WCSAR, $REG3
7055 026306 104076              ERROR    76              ; NOPE
7056
7057                                ; CHECK SECOND DATA
7058 026310 012777 001112 154520 115:  MOV      #ADDR. 1+1, @WCSAR ; ADDRESS THE WCS AT EXPEC'D LOC
7059 026316 027727 154516 010101  CMP      @WCSDR, #PAT. B  ; IS IT VALUE THAR WAS IN R4 ??
7060 026324 001415              BEQ      125              ; BR IF OK
7061 026326 012737 000004 001162  MOV      #4, $REG0
7062 026334 012737 010101 001164  MOV      #PAT. B, $REG1
7063 026342 017737 154472 001166  MOV      @WCSDR, $REG2
7064 026350 017737 154462 001170  MOV      @WCSAR, $REG3
7065 026356 104076              ERROR    76              ; NOPE
7066
7067 026360                      125:
7068 026360 000407              BR      TST65             ; EXIT
7069
7070
7071                                ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7072                                ; A BLOCK MOVE FUNCTION IN THE WCS.
7073 026362 012777 000020 154444 39$:  MOV      #PARDIS, @WCSST
7074 026370 022626              CMP      (SP)+, (SP)+
7075 026372 005077 154436              CLR      @WCSST
7076 026376 104077              ERROR    77
7077                                ; WCS WATCH DOG TIMED OUT
7078                                ; UPON TRANSFERRING CONTROL FROM
7079                                ; BASE MACHINE TO WCS MICROCODE.
7080                                ; NORMALLY, AFTER UCODE EXECUTION CONTROL
7081                                ; SHOULD BE RETURNED TO BASE MACHINE.
7082
7083
7084
7085

```

\*\*\*\*\*  
\*TEST 65 CHECK BLOCK LOAD OF LOAD-READ-2 FROM WCS



7086  
7087  
7088  
7089  
7090  
7091  
7092  
7093  
7094  
7095  
7096  
7097  
7098  
7099  
7100  
7101  
7102  
7103  
7104  
7105  
7106  
7107  
7108  
7109  
7110  
7111  
7112  
7113  
7114  
7115  
7116  
7117  
7118  
7119  
7120  
7121  
7122  
7123  
7124  
7125  
7126  
7127  
7128  
7129  
7130  
7131  
7132  
7133  
7134  
7135  
7136  
7137  
7138  
7139  
7140  
7141

THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:

WCS.ARRAY -> 'LOAD READ 2'

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:

R3, R4 (RESULTS STORED HERE)

THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN IT.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL STORE ADDRESS (-1) STARTING FROM WHICH THE REGISTERS  
SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE REGISTERS ARE THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS WAS LOADED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,  
BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

TST65: SCOPE  
MOV #TST66.NXTST ; STARTING ADDRESS OF NEXT TEST

JSR R3,LDWCS ; LOAD WCS WITH MICRO CODE  
6570 ; CONTROL BLOCK LOADED HERE  
MLDRD2 ; FROM THIS MEMORY LOCATION  
UDISP ; ENTRY POINT IN WCS

PAT. A= 050505 ; DATA PATTERN FOR TEST  
PAT. B= 105050 ;

026400 000004  
026402 012737 026616 003056  
026410 004537 037472  
026414 006570  
026416 057406  
026420 006002  
050505  
105050

```

7142          001222          ADDR. 1=1222          ;
7143          ;
7144 026422 005003          CLR      R3          ; ZAP BEFORE TEST
7145 026424 005004          CLR      R4          ;
7146 026426 005000          CLR      R0          ;
7147 026430 005001          CLR      R1          ;
7148 026432 005005          CLR      R5          ;
7149 026434 012702 001222  MOV      #ADDR. 1, R2          ; R2 IS LOCAL STORE ADDRESS HOLDER
7150 026440 010277 154372  MOV      R2, @WCSAR          ; SETUP LS WITH DATA
7151 026444 012777 050505 154366  MOV      #PAT. A, @WCSDR          ; STORE # 1
7152 026452 005277 154360          INC      @WCSAR          ;
7153 026456 012777 105050 154354  MOV      #PAT. B, @WCSDR          ; STORE # 2
7154          ;
7155 026464 012737 026600 000114  MOV      #395, @#114          ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7156 026472 012777 001000 154334  MOV      #MAINT, @WCSST          ; SET UP MAINT BIT, TIMER WILL START WHEN
7157          ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7158          ;
7159 026500 076700          XFCUDIS          ; TRANSFER CONTROL TO WCS
7160 026502          15:          ; SHOULD RETURN FROM WCS TO THIS POINT
7161          ;
7162 026502 042777 001000 154324  BIC      #MAINT, @WCSST          ; SHUT OFF THE TIMER
7163 026510 012737 003344 000114  MOV      #BADPAR, @#114          ; RESET UNEXPECTED PARITY HANDLER
7164          ;
7165          ; CHECK FIRST DATA
7166 026516 020327 050505          CMP      R3, #PAT. A          ; IS IT VALUE THAR WAS IN R3 ??
7167 026522 001411          BEQ      115          ; BR IF OK
7168 026524 012737 000003 001162  MOV      #3, $REG0          ;
7169 026532 012737 050505 001164  MOV      #PAT. A, $REG1          ;
7170 026540 010337 001166          MOV      R3, $REG2          ;
7171 026544 104100          ERROR    100          ; NOPE
7172          ;
7173          ; CHECK SECOND DATA
7174 026546 020427 105050          115:  CMP      R4, #PAT. B          ; IS IT VALUE THAT WAS IN R4 ??
7175 026552 001411          BEQ      125          ; BR IF OK
7176 026554 012737 000004 001162  MOV      #4, $REG0          ;
7177 026562 012737 105050 001164  MOV      #PAT. B, $REG1          ;
7178 026570 010437 001166          MOV      R4, $REG2          ;
7179 026574 104100          ERROR    100          ; NOPE
7180          ;
7181 026576          125:          BR      TST66          ; EXIT
7182 026576 000407          ;
7183          ;
7184          ;
7185          ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7186          ; A BLOCK MOVE FUNCTION IN THE WCS.
7187 026600 012777 000020 154226 395:  MOV      #PARDIS, @WCSST          ;
7188 026606 022626          CMP      (SP)+, (SP)+          ;
7189 026610 005077 154220          CLR      @WCSST          ;
7190 026614 104101          ERROR    101          ;
7191          ; WCS WATCH DOG TIMED OUT
7192          ; UPON TRANSFERRING CONTROL FROM
7193          ; BASE MACHINE TO WCS MICROCODE.
7194          ; NORMALLY, AFTER UCODE EXECUTION CONTROL
7195          ; SHOULD BE RETURNED TO BASE MACHINE.
7196
7197

```



7198  
7199  
7200  
7201  
7202  
7203  
7204  
7205  
7206  
7207  
7208  
7209  
7210  
7211  
7212  
7213  
7214  
7215  
7216  
7217  
7218  
7219  
7220  
7221  
7222  
7223  
7224  
7225  
7226  
7227  
7228  
7229  
7230  
7231  
7232  
7233  
7234  
7235  
7236  
7237  
7238  
7239  
7240  
7241  
7242  
7243  
7244  
7245  
7246  
7247  
7248  
7249  
7250  
7251  
7252  
7253

\*\*\*\*\*  
\*TEST 66 CHECK BLOCK STORE OF SET-STORE, WRITE-2 TO WCS

THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:

'SET STORE, WRITE 2' -> WCS.ARRAY

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.

THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:

R3, R4 (SETUP VIA MICROCODE, ANY INTERNAL  
REGISTERS COULD BE CHOSEN)

THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN THEM.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS  
SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,  
BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

026616 000004  
026620 012737 027054 003056  
026626 004537 037472  
026632 006430  
026634 057704  
026636 006002

TST66: SCOPE  
MOV #TST67,NXTST ; STARTING ADDRESS OF NEXT TEST  
JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE  
6430 ; CONTROL BLOCK LOADED HERE  
MWR2 ; FROM THIS MEMORY LOCATION  
UDISP ; ENTRY POINT IN WCS

```

7254
7255      125252      PAT. A= 125252 ; DATA PATTERN FOR TEST
7256      052525      PAT. B= 052525 ;
7257      001234      ADDR. 1= 1234 ;
7258
7259 026640 012703 125252      MOV      #PAT. A, R3      ; PATTERN # 1
7260 026644 012704 052525      MOV      #PAT. B, R4      ; PATTERN # 2
7261 026650 012702 001234      MOV      #ADDR. 1, R2     ; R2 IS LOCAL STORE ADDRESS HOLDER
7262 026654 005000      CLR      R0
7263 026656 005001      CLR      R1
7264 026660 005005      CLR      R5
7265
7266 026662 012737 027036 000114      MOV      #39%, @#114     ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7267 026670 012777 001000 154136      MOV      #MAINT, @WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
7268                                     ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7269
7270 026676 076700      XFCUDIS      ; TRANSFER CONTROL TO WCS
7271 026700      15:      ; SHOULD RETURN FROM WCS TO THIS POINT
7272
7273 026700 042777 001000 154126      BIC      #MAINT, @WCSST ; SHUT OFF THE TIMER
7274 026706 012737 003344 000114      MOV      #BADPAR, @#114 ; RESET UNEXPECTED PARITY HANDLER
7275
7276                                     ; CHECK FIRST DATA
7277 026714 012777 001234 154114      MOV      #ADDR. 1, @WCSAR ; ADDRESS THE WCS AT EXPECT'D LOC
7278 026722 027727 154112 125252      CMP      @WCSDR, #PAT. A ; IS IT VALUE THAR WAS IN R3 ??
7279 026730 001415      BEQ      11%      ; BR IF OK
7280 026732 012737 000003 001162      MOV      #3, $REG0
7281 026740 012737 125252 001164      MOV      #PAT. A, $REG1
7282 026746 017737 154066 001166      MOV      @WCSDR, $REG2
7283 026754 017737 154056 001170      MOV      @WCSAR, $REG3
7284 026762 104102      ERROR    102      ; NOPE
7285
7286                                     ; CHECK SECOND DATA
7287 026764 012777 001235 154044 11%:      MOV      #ADDR. 1+1, @WCSAR ; ADDRESS THE WCS AT EXPECT'D LOC
7288 026772 027727 154042 052525      CMP      @WCSDR, #PAT. B ; IS IT VALUE THAR WAS IN R4 ??
7289 027000 001415      BEQ      12%      ; BR IF OK
7290 027002 012737 000004 001162      MOV      #4, $REG0
7291 027010 012737 052525 001164      MOV      #PAT. B, $REG1
7292 027016 017737 154016 001166      MOV      @WCSDR, $REG2
7293 027024 017737 154006 001170      MOV      @WCSAR, $REG3
7294 027032 104102      ERROR    102      ; NOPE
7295
7296 027034      12%:      BP      TST67      ; EXIT
7297 027034 000407
7298
7299
7300                                     ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7301                                     ; A BLOCK MOVE FUNCTION IN THE WCS.
7302 027036 012777 000020 153770 39%:      MOV      #PARDIS, @WCSST
7303 027044 022626      CMP      (SP)+, (SP)+
7304 027046 005077 153762      CLR      @WCSST
7305 027052 104103      ERROR    103
7306                                     ; WCS WATCH DOG TIMED OUT
7307                                     ; UPON TRANSFERRING CONTROL FROM
7308                                     ; BASE MACHINE TO WCS MICROCODE.
7309                                     ; NORMALLY, AFTER UCODE EXECUTION CONTROL

```



7310  
7311  
7312  
7313  
7314  
7315  
7316  
7317  
7318  
7319  
7320  
7321  
7322  
7323  
7324  
7325  
7326  
7327  
7328  
7329  
7330  
7331  
7332  
7333  
7334  
7335  
7336  
7337  
7338  
7339  
7340  
7341  
7342  
7343  
7344  
7345  
7346  
7347  
7348  
7349  
7350  
7351  
7352  
7353  
7354  
7355  
7356  
7357  
7358  
7359  
7360  
7361  
7362  
7363  
7364  
7365

; SHOULD BE RETURNED TO BASE MACHINE.

\*\*\*\*\*  
\*TEST 67 CHECK BLOCK LOAD OF SET-LOAD, INC-READ-2 FROM WCS

THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:

WCS.ARRAY -> 'SET LOAD, INC READ 2'

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:

R3, R4 (RESULTS STORED HERE)

THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN IT.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL STORE ADDRESS (-1) STARTING FROM WHICH THE REGISTERS  
SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE REGISTERS ARE THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS WAS LOADED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUR<9:0>, ARRAY ADDRESS MUX-A-PORT,  
BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

027054 000004  
027056 012737 027272 003056  
027064 004537 037472  
027070 006410

TST67: SCOPE  
MOV #TST70,NXTST ; STARTING ADDRESS OF NEXT TEST  
JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE  
6410 ; CONTROL BLOCK LOADED HERE

```

PDP-11/60 WCS DIAGNOSTIC          MACY11 30(1046) 25-OCT-77 20:12 PAGE 136          SEQ 0138
DQKUAAR.P11 25-OCT-77 20:03      T67          CHECK BLOCK LOAD OF SET-LOAD, INC-READ-2 FROM WCS          SEQ 0149
                                     G 12
7366 027072 057614          MICRD2          ; FROM THIS MEMORY LOCATION
7367 027074 006002          UDISP          ; ENTRY POINT IN WCS
7368
7369          070707          PAT. A= 070707 ; DATA PATTERN FOR TEST
7370          107070          PAT. B= 107070 ;
7371          001432          ADDR. 1= 1432  ;
7372
7373 027076 005003          CLR R3          ; ZAP BEFORE TEST
7374 027100 005004          CLR R4          ;
7375 027102 005000          CLR R0          ;
7376 027104 005001          CLR R1          ;
7377 027106 005005          CLR R5          ;
7378 027110 012702 001432          MOV #ADDR. 1, R2 ; R2 IS LOCAL STORE ADDRESS HOLDER
7379 027114 010277 153716          MOV R2, @WCSAR  ; SETUP LS WITH DATA
7380 027120 012777 070707 153712          MOV #PAT. A, @WCSDR ; STORE # 1
7381 027126 005277 153704          INC @WCSAR      ;
7382 027132 012777 107070 153700          MOV #PAT. B, @WCSDR ; STORE # 2
7383
7384 027140 012737 027254 000114          MOV #395, @#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7385 027146 012777 001000 153660          MOV #MAINT, @WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
7386          ; WCS HOT BOX IS SELECTED. UPON TRANSFERRING CONTROL TO WCS
7387
7388 027154 076700          XFCUDIS        ; TRANSFER CONTROL TO WCS
7389 027156          15:          ; SHOULD RETURN FROM WCS TO THIS POINT
7390
7391 027156 042777 001000 153650          BIC #MAINT, @WCSST ; SHUT OFF THE TIMER
7392 027164 012737 003344 000114          MOV #BADPAR, @#114 ; RESET UNEXPECTED PARITY HANDLER
7393
7394          ; CHECK FIRST DATA
7395 027172 020327 070707          CMP R3, #PAT. A ; IS IT VALUE THAT WAS IN R3 ??
7396 027176 001411          BEQ 115        ; BR IF OK
7397 027200 012737 000003 001162          MOV #3, $REG0
7398 027206 012737 070707 001164          MOV #PAT. A, $REG1
7399 027214 010337 001166          MOV R3, $REG2
7400 027220 104104          ERROR 104      ; NOPE
7401
7402          ; CHECK SECOND DATA
7403 027222 020427 107070          115:          CMP R4, #PAT. B ; IS IT VALUE THAT WAS IN R4 ??
7404 027226 001411          BEQ 125        ; BR IF OK
7405 027230 012737 000004 001162          MOV #4, $REG0
7406 027236 012737 107070 001164          MOV #PAT. B, $REG1
7407 027244 010437 001166          MOV R4, $REG2
7408 027250 104104          ERROR 104      ; NOPE
7409
7410 027252          125:
7411 027252 000407          BR TST70      ; EXIT
7412
7413
7414          ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7415          ; A BLOCK MOVE FUNCTION IN THE WCS.
7416 027254 012777 000020 153552 395:          MOV #PARDIS, @WCSST
7417 027262 022626          CMP (SP)+, (SP)+
7418 027264 005077 153544          CLR @WCSST
7419 027270 104105          ERROR 105
7420          ; WCS WATCH DOG TIMED OUT
7421          ; UPON TRANSFERRING CONTROL FROM

```



7422  
7423  
7424  
7425  
7426  
7427  
7428  
7429  
7430  
7431  
7432  
7433  
7434  
7435  
7436  
7437  
7438  
7439  
7440  
7441  
7442  
7443  
7444  
7445  
7446  
7447  
7448  
7449  
7450  
7451  
7452  
7453  
7454  
7455  
7456  
7457  
7458  
7459  
7460  
7461  
7462  
7463  
7464  
7465  
7466  
7467  
7468  
7469  
7470  
7471  
7472  
7473  
7474  
7475  
7476  
7477

;BASE MACHINE TO WCS MICROCODE.  
;NORMALLY, AFTER UCODE EXECUTION CONTROL  
;SHOULD BE RETURNED TO BASE MACHINE.

\*\*\*\*\*

\*TEST 70 CHECK BLOCK STORE OF WRITE-INDIRECT TO WCS

THIS TEST CHECKS THE STORE FUNCTION OF THE WCS:

'WRITE-INDIRECT FROM WCSA 0 ' -> WCS.ARRAY

WHEN THIS FUNCTION IS INITIATED A SINGLE REGISTER  
GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.

THE FOLLOWING REGISTER IS STORED IN THE GIVEN ORDER:

- 1) 'LS.ADDR' (FROM R2) IS USED TO FETCH NEW  
'LS.ADDR'
- 2) A 'WCSA 0 ' (ALWAYS) IS STORED AT NEW  
'LS.ADDR' IN LS

THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN THEM.

THE MICRO-CODE IS STORED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTER  
SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE.  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9: 0>, ARRAY ADDRESS MUX-A-PORT,  
BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

```

7478 027272 000004          TST70: SCOPE
7479 027274 012737 027524 003056 MOV      #TST71,NXTST      ; STARTING ADDRESS OF NEXT TEST
7480                                     ;
7481 027302 004537 037472 JSR      R5,LDWCS          ; LOAD WCS WITH MICRO CODE
7482 027306 006540          6540          ; CONTROL BLOCK STORED HERE
7483 027310 057554          MWRIND        ; FROM THIS MEMORY LOCATION
7484 027312 006002          UDISP         ; ENTRY POINT IN WCS
7485                                     ;
7486          012345          PAT. A= 012345      ; DATA PATTERN FOR TEST
7487          001234          ADDR. 1= 1234      ; FIRST ADDRESS IN LS
7488          001342          ADDR. 2= 1342      ; SECOND " " "
7489                                     ;
7490 027314 012700 012345 MOV      #PAT. A,R0        ; GET PATTERN
7491 027320 076600 000220 MED      ,220            ; INTO WCSA 0
7492 027324 005000          CLR      R0              ; ZAP GPR'S
7493 027326 005001          CLR      R1
7494 027330 005003          CLR      R3
7495 027332 005004          CLR      R4
7496 027334 005005          CLR      R5
7497 027336 012702 001234 MOV      #ADDR. 1,R2        ; 1ST WCS ADDR POINTS TO SECOND
7498 027342 010277 153470 MOV      R2,@WCSAR         ; ADDRESS WCS AT 1ST
7499 027346 012777 001342 153464 MOV      #ADDR. 2,@WCSAR   ; STORE 2ND ADDR, DATA GOES AT HERE
7500 027354 005077 153456 CLR      @WCSAR           ; POINT AWAY
7501                                     ;
7502 027360 012737 027506 000114 MOV      #39%,@#114        ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7503 027366 012777 001000 153440 MOV      #MAINT,@WCSST     ; SET UP MAINT BIT, TIMER WILL START WHEN
7504                                     ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7505                                     ;
7506 027374 076700          XFCDDIS          ; TRANSFER CONTROL TO WCS
7507 027376          15:          ; SHOULD RETURN FROM WCS TO THIS POINT
7508                                     ;
7509 027376 042777 001000 153430 BIC      #MAINT,@WCSST     ; SHUT OFF THE TIMER
7510 027404 012737 003344 000114 MOV      #BADPAR,@#114     ; RESET UNEXPECTED PARITY HANDLER
7511                                     ;
7512          ; CHECK DATA WAS STORED AT LOCATION ADDR. 2, NOT ADDR. 1
7513 027412 012777 001342 153416 MOV      #ADDR. 2,@WCSAR   ; ADDR 2ND LOCATION IN LS
7514 027420 027727 153414 012345 CMP      @WCSDR,#PAT. A    ; DATA MATCH ??
7515 027426 001426          BEQ      12%          ; BR IF OK
7516 027430 012700 001234 MOV      #ADDR. 1,R0
7517 027434 010037 001162 MOV      R0,$REG0
7518 027440 010077 153372 MOV      R0,@WCSAR
7519 027444 017737 153370 001164 MOV      @WCSDR,$REG1
7520 027452 012701 001342 MOV      #ADDR. 2,R1
7521 027456 010137 001166 MOV      R1,$REG2
7522 027462 010177 153350 MOV      R1,@WCSAR
7523 027466 017737 153346 001172 MOV      @WCSDR,$REG4
7524 027474 012737 012345 001170 MOV      #PAT. A,$REG3
7525 027502 104106          ERROR 106          ; NOPE
7526                                     ;
7527 027504          12%:          BR      TST71          ; EXIT
7528 027504 000407
7529
7530
7531          ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7532          ; A BLOCK MOVE FUNCTION IN THE WCS.
7533 027506 012777 000020 153320 39% MOV      #PARDIS,@WCSST

```



7534 027514 022626  
7535 027516 005077 153312  
7536 027522 104107  
7537  
7538  
7539  
7540  
7541  
7542  
7543  
7544  
7545  
7546  
7547  
7548  
7549  
7550  
7551  
7552  
7553  
7554  
7555  
7556  
7557  
7558  
7559  
7560  
7561  
7562  
7563  
7564  
7565  
7566  
7567  
7568  
7569  
7570  
7571  
7572  
7573  
7574  
7575  
7576  
7577  
7578  
7579  
7580  
7581  
7582  
7583  
7584  
7585  
7586  
7587  
7588  
7589

CMP (SP)+,(SP)+ ;  
CLR @WCSST ;  
ERROR 107 ;  
;WCS WATCH DOG TIMED OUT  
;UPON TRANSFERRING CONTROL FROM  
;BASE MACHINE TO WCS MICROCODE.  
;NORMALLY, AFTER UCODE EXECUTION CONTROL  
;SHOULD BE RETURNED TO BASE MACHINE.

\*\*\*\*\*

\*TEST 71 CHECK BLOCK LOAD OF READ-INDIRECT FROM WCS

THIS TEST CHECKS THE LOAD FUNCTION OF THE WCS:

WCS.ARRAY -> 'READ-INDIRECT TO "D"'

WHEN THIS FUNCTION IS INITIATED A SINGLE REGISTER  
GET TRANSFERRED FROM THE WCS TO THE BASE MACHINE.  
"D" IS COPIED INTO GPR R3 AFTER THE TMS FUNCTION.

THE FOLLOWING REGISTER IS LOADED IN THE GIVEN ORDER:

- 1) 'LS.ADDR' (FROM R2) IS USED TO FETCH NEW  
'LS.ADDR'
- 2) R3 IS LOADED (FROM "D") WITH DATA READ  
FROM LS AT NEW 'LS.ADDR'

THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN THEM.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL SFROMRE ADDRESS (-1) STARTING TO WHICH THE REGISTER  
SHOULD BE LOADED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN LOADED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE REGISTER IS THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS WAS LOAD CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY

```
7590 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
7591 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
7592 ; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
7593 ; BUS-U-MUX-B PORT, BUS-U-LINES).
7594 ;
7595 ;*****
7596 027524 000004 TST71: SCOPE
7597 027526 012737 027750 003056 MOV #TST72,NXTST ; STARTING ADDRESS OF NEXT TEST
7598 ;
7599 027534 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
7600 027540 006550 6550 ; CONTROL BLOCK LOAD HERE
7601 027542 057514 MRDIND ; FROM THIS MEMORY LOCATION
7602 027544 006002 UDISP ; ENTRY POINT IN WCS
7603 ;
7604 054321 PAT. A= 054321 ; DATA PATTERN FOR TEST
7605 001222 ADDR. 1= 1222 ; FIRST ADDRESS IN LS
7606 001333 ADDR. 2= 1333 ; SECOND " " "
7607 ;
7608 027546 005000 CLR R0 ; ZAP GPR'S
7609 027550 005001 CLR R1
7610 027552 005003 CLR R3
7611 027554 005004 CLR R4
7612 027556 005005 CLR R5
7613 027560 012702 001222 MOV #ADDR. 1,R2 ; 1ST WCS ADDR POINTS TO SECOND
7614 027564 010277 153246 MOV R2,@WCSAR ; ADDRESS WCS AT 1ST
7615 027570 012777 001333 153242 MOV #ADDR. 2,@WCSAR ; LOAD 2ND ADDR, DATA GOES AT HERE
7616 027576 012777 001333 153232 MOV #ADDR. 2,@WCSAR ; 2ND WCS ADDR POINTS AT DATA
7617 027604 012777 054321 153226 MOV #PAT. A,@WCSAR ; STORE DATA
7618 027612 005077 153220 CLR @WCSAR ; POINT AWAY
7619 ;
7620 027616 012737 027732 000114 MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7621 027624 012777 001000 153202 MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
7622 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7623 ;
7624 027632 076700 XFCUDIS ; TRANSFER CONTROL TO WCS
7625 027634 15: ; SHOULD RETURN FROM WCS TO THIS POINT
7626 ;
7627 027634 042777 001000 153172 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
7628 027642 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
7629 ;
7630 ; CHECK DATA WAS LOADED FROM LOCATION ADDR. 2, NOT ADDR. 1
7631 027650 020327 054321 CMP R3,#PAT. A ; DATA MATCH ??
7632 027654 001425 BEQ 125 ; BR IF OK
7633 027656 012700 001222 MOV #ADDR. 1,R0
7634 027662 010037 001162 MOV R0,$REG0
7635 027666 010077 153144 MOV R0,@WCSAR
7636 027672 017737 153142 001164 MOV @WCSAR,$REG1
7637 027700 012701 001333 MOV #ADDR. 2,R1
7638 027704 010137 001166 MOV R1,$REG2
7639 027710 010177 153122 MOV R1,@WCSAR
7640 027714 017737 153120 001170 MOV @WCSAR,$REG3
7641 027722 010337 001172 MOV R3,$REG4
7642 027726 104110 ERROR 110 ; NOPE
7643 ;
7644 027730 125: BR TST72 ; EXIT
7645 027730 000407
```



```

7646
7647
7648
7649
7650 027732 012777 000020 153074 395:
7651 027740 022626
7652 027742 005077 153066
7653 027746 104111
7654
7655
7656
7657
7658
7659
7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701

```

```

;
;
; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
; A BLOCK MOVE FUNCTION IN THE WCS.
MOV #PARDIS, @WCSST
CMP (SP)+, (SP)+
CLR @WCSST
ERROR 111
; WCS WATCH DOG TIMED OUT
; UPON TRANSFERRING CONTROL FROM
; BASE MACHINE TO WCS MICROCODE.
; NORMALLY, AFTER UCODE EXECUTION CONTROL
; SHOULD BE RETURNED TO BASE MACHINE.

```

```

; *****

```

```

; *TEST 72 CHECK BLOCK STORE OF WFP.AC 0:1 TO WCS

```

```

; THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:

```

```

; WFP.AC 0:1 -> WCS.ARRAY

```

```

; WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
; GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.

```

```

; THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:

```

```

; WFP.AC0 0 , WFP.AC1 0 , WFP.AC2 0 , WFP.AC3 0 ,
; WFP.AC0 1 , WFP.AC1 1 , WFP.AC2 1 , WFP.AC3 1

```

```

; THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING
; UNIQUE DATA PATTERNS IN THEM.

```

```

; THE MICRO-CODE IS LOADED INTO THE WCS. USING
; AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
; TO THE MICRO-CODE IN THE WCS.

```

```

; FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),
; THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS
; SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS
; THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
; WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE
; REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS
; CONTROL TO THE BASE MACHINE.

```

```

; THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
; DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,
; AN ERROR IS REPORTED.

```

```

; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
; A RESULT OF SOME FAULT CONDITION, THE CONTROL
; IS NOT TRANSFERRED BACK TO THE BASE MACHINE,
; THE WATCH DOG WILL TIME OUT AND A TRAP TO
; 114 WILL OCCUR.

```

```

; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

```

```

7702 ;
7703 ;
7704 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
7705 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
7706 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
7707 ; THE WCS- (NUR<9:0>, ARRAY ADDRESS MUX-A-PORT,
7708 ; BUS-U-MUX-B PORT, BUS-U-LINES).
7709 ; *****
7710 027750 000004 TST72: SCOPE
7711 027752 012737 030220 003056 MOV #TST73,NXTST ; STARTING ADDRESS OF NEXT TEST
7712 027760 005037 001162 CLR $REGO ; ZAP FOR PRINTOUT
7713 ;
7714 027764 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
7715 027770 006610 6610 ; CONTROL BLOCK LOADED HERE
7716 027772 060026 MFO1LS ; FROM THIS MEMORY LOCATION
7717 027774 006002 UDISP ; ENTRY POINT IN WCS
7718 ;
7719 ; CLEAR -ALL- WFP AC'S
7720 027776 012705 003144 MOV #FPOSXX,R5 ; TABLE OF MED CODES
7721 030002 012704 000030 MOV #24,R4 ;
7722 030006 112537 030024 115: MOV (R5)+,125 ; GET A CODE
7723 030012 152737 000200 030024 BISB #200,125 ; MAKE IT 'WRITE'
7724 030020 005000 CLR R0 ; WRITE ZEROES
7725 030022 076600 MED ; DO IT
7726 030024 000000 125: .WORD 0 ;
7727 030026 077411 SOB R4,115 ; LOOP ON ALL
7728 ;
7729 ; SETUP WFP AC'S 0:1 WITH DATA (= MED CODE)
7730 030030 012705 003144 MOV #FPO1XX,R5 ; TABLE OF CODES
7731 030034 012704 000010 MOV #8,R4 ;
7732 030040 112500 135: MOV (R5)+,R0 ; GET MED CODE
7733 030042 052700 125200 BIS #125200,R0 ; DATA = 125MED
7734 030046 110037 030054 MOV R0,145 ; STORE WRITE MED CODE
7735 030052 076600 MED ; WRITE THE DATA
7736 030054 000000 145: .WORD 0 ;
7737 030056 077410 SOB R4,135 ; LOOP
7738 ;
7739 030060 012702 001234 MOV #1234,R2 ; R2 IS LOCAL STORE ADDRESS HOLDER
7740 ;
7741 030064 012737 030202 000114 MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7742 030072 012777 001000 152734 MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
7743 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7744 ;
7745 030100 076700 XFCUDIS ; TRANSFER CONTROL TO WCS
7746 030102 15: ; SHOULD RETURN FROM WCS TO THIS POINT
7747 ;
7748 030102 042777 001000 152724 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
7749 030110 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
7750 ;
7751 030116 012705 003144 MOV #FPO1XX,R5 ; PTR TO MED CODE TABLE, ALSO DATA
7752 030122 012704 000010 MOV #8,R4 ; # REGISTERS TO SAVE
7753 ;
7754 030126 010277 152704 205: MOV R2,@WCSAR ; POINT AT DATA IN WCS
7755 030132 017703 152702 MOV @WCSDR,R3 ; GET WCS DATA
7756 ;
7757 030136 112501 MOV (R5)+,R1 ; GET DATA FROM TABLE

```



```

7758 030140 052701 125200      BIS      #125200,R1      ;
7759                                ;
7760 030144 020103      CMP      R1,R3      ; DATA MATCH ???
7761 030146 001412      BEQ      21$      ; BR IF OK ...
7762 030150 116537 177777 001162  MOVB    -1(R5), $REG0 ; WFP AC MED CODE
7763 030156 010237 001164      MOV      R2,$REG1    ; LS ADDR
7764 030162 010137 001166      MOV      R1,$REG2    ; EXP'D DATA
7765 030166 010337 001170      MOV      R3,$REG3    ; RCV'D DATA
7766 030172 104112      ERROR    112      ; TELL ABOUT ERROR
7767                                ;
7768 030174 005202      21$: INC      R2      ; NEXT LS ADDR
7769 030176 077425      SOB      R4,20$    ; LOOP FOR ALL
7770                                ;
7771 030200 000407      BR      TST73      ; EXIT
7772                                ;
7773                                ;
7774                                ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7775                                ; A BLOCK MOVE FUNCTION IN THE WCS.
7776 030202 012777 000020 152624 39$: MOV     #PARDIS,@WCSST ;
7777 030210 022626      CMP     (SP)+,(SP)+ ;
7778 030212 005077 152616      CLR     @WCSST      ;
7779 030216 104113      ERROR    113      ;
7780                                ; WCS WATCH DOG TIMED OUT
7781                                ; UPON TRANSFERRING CONTROL FROM
7782                                ; BASE MACHINE TO WCS MICROCODE.
7783                                ; NORMALLY, AFTER UCODE EXECUTION CONTROL
7784                                ; SHOULD BE RETURNED TO BASE MACHINE.
7785
7786
7787 *****
7788 *TEST 73 CHECK BLOCK STORE OF WFP.AC 2:3 TO WCS
7789
7790 THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:
7791
7792 WFP.AC 2:3 -> WCS.ARRAY
7793
7794 WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
7795 GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.
7796
7797 THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:
7798
7799 WFP.AC0 2 , WFP.AC1 2 , WFP.AC2 2 , WFP.AC3 2 ,
7800 WFP.AC0 3 , WFP.AC1 3 , WFP.AC2 3 , WFP.AC3 3
7801
7802 THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING
7803 UNIQUE DATA PATTERNS IN THEM.
7804
7805 THE MICRO-CODE IS LOADED INTO THE WCS. USING
7806 AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
7807 TO THE MICRO-CODE IN THE WCS.
7808
7809 FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),
7810 THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS
7811 SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS
7812 THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
7813 WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE

```



```

7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836 030220 000004
7837 030222 012737 030464 003056
7838
7839 030230 004537 037472
7840 030234 006630
7841 030236 060126
7842 030240 006002
7843
7844
7845 030242 012705 003144
7846 030246 012704 000030
7847 030252 112537 030270 115:
7848 030256 152737 000200 030270
7849 030264 005000
7850 030266 076600
7851 030270 000000 125:
7852 030272 077411
7853
7854
7855 030274 012705 003154
7856 030300 012704 000010
7857 030304 112500 135:
7858 030306 052700 125200
7859 030312 110037 030320
7860 030316 076600
7861 030320 000000 145:
7862 030322 077410
7863
7864 030324 012702 001222
7865
7866 030330 012737 030446 000114
7867 030336 012777 001000 152470
7868
7869

```

REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE.

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT, AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS A RESULT OF SOME FAULT CONDITION, THE CONTROL IS NOT TRANSFERRED BACK TO THE BASE MACHINE, THE WATCH DOG WILL TIME OUT AND A TRAP TO 114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION CONTROL PATH USED BY MICRO-CODE EXECUTING FROM THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT, BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

```

TST73: SCOPE
MOV #TST74,NXTST ;STARTING ADDRESS OF NEXT TEST
;
JSR R5,LDWCS ;LOAD WCS WITH MICRO CODE
6630 ;CONTROL BLOCK LOADED HERE
MF23LS ;FROM THIS MEMORY LOCATION
UDISP ;ENTRY POINT IN WCS
;
;CLEAR -ALL- WFP AC'S
MOV #FP05XX,R5 ;TABLE OF MED CODES
MOV #24,R4 ;
MOVB (R5)+,125 ;GET A CODE
BISB #200,125 ;MAKE IT 'WRITE'
CLR R0 ;WRITE ZEROES
MED ;DO IT
;
125: .WORD 0
SOB R4,115 ;LOOP ON ALL
;
;SETUP WFP AC'S 2:3 WITH DATA (= MED CODE)
MOV #FP23XX,R5 ;TABLE OF CODES
MOV #8,R4 ;
135: MOVB (R5)+,R0 ;GET MED CODE
BIS #125200,R0 ;DATA = 125MED
MOVB R0,145 ;STORE WRITE MED CODE
MED ;WRITE THE DATA
;
145: .WORD 0
SOB R4,135 ;LOOP
;
MOV #1222,R2 ;R2 IS LOCAL STORE ADDRESS HOLDER
;
MOV #395,@#114 ;SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
MOV #MAINT,@WCSST ;SET UP MAINT BIT. TIMER WILL START WHEN
;WCS HOT BOX IS SELECTED. UPON TRANSFERRING CONTROL TO WCS
;

```



```

7870 030344 076700          XFCUDIS          ;TRANSFER CONTROL TO WCS
7871 030346          15:          ;SHOULD RETURN FROM WCS TO THIS POINT
7872          ;
7873 030346 042777 001000 152460      BIC      #MAINT,@WCSST      ;SHUT OFF THE TIMER
7874 030354 012737 003344 000114      MOV      #BADPAR,@#114     ;RESET UNEXPECTED PARITY HANDLER
7875          ;
7876 030362 012705 003154          MOV      #FP23XX,R5        ;PTR TO MED CODE TABLE, ALSO DATA
7877 030366 012704 000010          MOV      #8.,R4           ;# REGISTERS TO SAVE
7878          ;
7879 030372 010277 152440      205:     MOV      R2,@WCSAR        ;POINT AT DATA IN WCS
7880 030376 017703 152436          MOV      @WCSDR,R3        ;GET WCS DATA
7881          ;
7882 030402 112501          MOVVB    (R5)+,R1          ;GET DATA FROM TABLE
7883 030404 052701 125200          BIS      #125200,R1       ;
7884          ;
7885 030410 020103          CMP      R1,R3           ;DATA MATCH ???
7886 030412 001412          BEQ      215             ;BR IF OK ...
7887 030414 116537 177777 001162      MOVVB    -1(R5),SREG0     ;WFP AC MED CODE
7888 030422 010237 001164          MOV      R2,SREG1        ;LS ADDR
7889 030426 010137 001166          MOV      R1,SREG2        ;EXP'D DATA
7890 030432 010337 001170          MOV      R3,SREG3        ;RCV'D DATA
7891 030436 104114          ERROR   114             ;TELL ABOUT ERROR
7892          ;
7893 030440          215:     INC      R2                ;NEXT LS ADDR
7894 030442 077425          SOB      R4,205         ;LOOP FOR ALL
7895          ;
7896 030444 000407          BR       TST74           ;EXIT
7897          ;
7898          ;
7899          ;ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
7900          ;A BLOCK MOVE FUNCTION IN THE WCS.
7901 030446 012777 000020 152360      395:     MOV      #PARDIS,@WCSST
7902 030454 022626          CMP      (SP)+,(SP)+
7903 030456 005077 152352          CLR      @WCSST
7904 030462 104115          ERROR   115
7905          ;WCS WATCH DOG TIMED OUT
7906          ;UPON TRANSFERRING CONTROL FROM
7907          ;BASE MACHINE TO WCS MICROCODE.
7908          ;NORMALLY, AFTER UCODE EXECUTION CONTROL
7909          ;SHOULD BE RETURNED TO BASE MACHINE.
7910          ;
7911          ;*****
7912          ;*TEST 74      CHECK BLOCK STORE OF WFP.AC 4:5 TO WCS
7913          ;
7914          ;
7915          ;THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:
7916          ;
7917          ;      WFP.AC 4:5 -> WCS.ARRAY
7918          ;
7919          ;WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
7920          ;GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.
7921          ;
7922          ;THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:
7923          ;
7924          ;      WFP.AC0 4 , WFP.AC1 4 , WFP.AC2 4 , WFP.AC3 4 ,
7925          ;      WFP.AC0 5 , WFP.AC1 5 , WFP.AC2 5 , WFP.AC3 5

```

```

7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961 030464 000004
7962 030466 012737 030730 003056
7963
7964 030474 004537 037472
7965 030500 006650
7966 030502 060226
7967 030504 006002
7968
7969
7970 030506 012705 003144
7971 030512 012704 000030
7972 030516 112537 030534
7973 030522 152737 000200 030534
7974 030530 005000
7975 030532 076600
7976 030534 000000
7977 030536 077411
7978
7979
7980 030540 012705 003164
7981 030544 012704 000010

```

THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING UNIQUE DATA PATTERNS IN THEM.

THE MICRO-CODE IS LOADED INTO THE WCS. USING AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE.

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT, AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS A RESULT OF SOME FAULT CONDITION, THE CONTROL IS NOT TRANSFERRED BACK TO THE BASE MACHINE, THE WATCH DOG WILL TIME OUT AND A TRAP TO 114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION CONTROL PATH USED BY MICRO-CODE EXECUTING FROM THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT, BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

```

TST74: SCOPE
MOV #TST75,NXTST ; STARTING ADDRESS OF NEXT TEST
;
JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
6650 ; CONTROL BLOCK LOADED HERE
MF45LS ; FROM THIS MEMORY LOCATION
UDISP ; ENTRY POINT IN WCS
;
; CLEAR -ALL- WFP AC'S
MOV #FP05XX,R5 ; TABLE OF MED CODES
MOV #24,R4 ;
11$: MOVB (R5)+,12$ ; GET A CODE
BISB #200,12$ ; MAKE IT 'WRITE'
CLR R0 ; WRITE ZEROES
MED ; DO IT
;
12$: .WORD 0 ;
SOB R4,11$ ; LOOP ON ALL
;
; SETUP WFP AC'S 4:5 WITH DATA (= MED CODE)
MOV #FP45XX,R5 ; TABLE OF CODES
MOV #8,R4 ;

```



```

7982 030550 112500      135:  MOVB   (R5)+,R0      ;GET MED CODE
7983 030552 052700 125200  BIS    #125200,R0    ;DATA = 125MED
7984 030556 110037 030564  MOVB   R0,R4        ;STORE WRITE MED CODE
7985 030562 076600      MED    ;WRITE THE DATA
7986 030564 000000      145:  .WORD  0           ;
7987 030566 077410      SOB    R4,135       ;LOOP
7988                                     ;
7989 030570 012702 001432  MOV    #1432,R2     ;R2 IS LOCAL STORE ADDRESS HOLDER
7990                                     ;
7991 030574 012737 030712 000114  MOV    #395,@#114   ;SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
7992 030602 012777 001000 152224  MOV    #MAINT,@WCSST ;SET UP MAINT BIT, TIMER WILL START WHEN
7993                                     ;WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
7994                                     ;
7995 030610 076700      XFCUDIS ;TRANSFER CONTROL TO WCS
7996 030612      15:    ;SHOULD RETURN FROM WCS TO THIS POINT
7997                                     ;
7998 030612 042777 001000 152214  BIC    #MAINT,@WCSST ;SHUT OFF THE TIMER
7999 030620 012737 003344 000114  MOV    #BADPAR,@#114 ;RESET UNEXPECTED PARITY HANDLER
8000                                     ;
8001 030626 012705 003164      MOV    #FP45XX,R5   ;PTR TO MED CODE TABLE ALSO DATA
8002 030632 012704 000010      MOV    #S,R4        ;# REGISTERS TO SAVE
8003                                     ;
8004 030636 010277 152174      205:  MOV    R2,@WCSAR    ;POINT AT DATA IN WCS
8005 030642 017703 152172      MOV    @WCSDR,R3    ;GET WCS DATA
8006                                     ;
8007 030646 112501      MOVB   (R5)+,R1     ;GET DATA FROM TABLE
8008 030650 052701 125200      BIS    #125200,R1   ;
8009                                     ;
8010 030654 020103      CMP    R1,R3        ;DATA MATCH ???
8011 030656 001412      BEQ    215          ;BR IF OK
8012 030660 116537 177777 001162  MOVB   -1(R5),%REG0 ;WFP AC MED CODE
8013 030666 010237 001164      MOV    R2,%REG1     ;LS ADDR
8014 030672 010137 001166      MOV    R1,%REG2     ;EXP'D DATA
8015 030676 010337 001170      MOV    R3,%REG3     ;RCV'D DATA
8016 030702 104116      ERROR  116         ;TELL ABOUT ERROR
8017                                     ;
8018 030704 005202      215:  INC    R2           ;NEXT LS ADDR
8019 030706 077425      SOB    R4,205      ;LOOP FOR ALL
8020                                     ;
8021 030710 000407      BR     TST75        ;EXIT
8022                                     ;
8023                                     ;
8024                                     ;ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
8025                                     ;A BLOCK MOVE FUNCTION IN THE WCS.
8026 030712 012777 000020 152114  395:  MOV    #PARDIS,@WCSST ;
8027 030720 022626      CMP    (SP)+,(SP)+  ;
8028 030722 005077 152106      CLR    @WCSST       ;
8029 030726 104117      ERROR  117         ;
8030                                     ;WCS WATCH DOG TIMED OUT
8031                                     ;UPON TRANSFERRING CONTROL FROM
8032                                     ;BASE MACHINE TO WCS MICROCODE.
8033                                     ;NORMALLY, AFTER UCODE EXECUTION CONTROL
8034                                     ;SHOULD BE RETURNED TO BASE MACHINE.
8035
8036
8037

```

\*\*\*\*\*

8038  
8039  
8040  
8041  
8042  
8043  
8044  
8045  
8046  
8047  
8048  
8049  
8050  
8051  
8052  
8053  
8054  
8055  
8056  
8057  
8058  
8059  
8060  
8061  
8062  
8063  
8064  
8065  
8066  
8067  
8068  
8069  
8070  
8071  
8072  
8073  
8074  
8075  
8076  
8077  
8078  
8079  
8080  
8081  
8082  
8083  
8084  
8085  
8086  
8087  
8088  
8089  
8090  
8091  
8092  
8093

```

: *TEST 75      CHECK BLOCK LOAD OF WFP.AC 0:1 FROM WCS
:
: THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:
:
:   WCS.ARRAY -> WFP.AC 0:1
:
: WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
: GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.
:
: THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:
:
:   WFP.AC0 0 , WFP.AC1 0 , WFP.AC2 0 , WFP.AC3 0
:   WFP.AC0 1 , WFP.AC1 1 , WFP.AC2 1 , WFP.AC3 1
:
: THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING
: UNIQUE DATA PATTERNS IN IT.
:
: THE MICRO-CODE IS LOADED INTO THE WCS. USING
: AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
: TO THE MICRO-CODE IN THE WCS.
:
: FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),
: THE LOCAL LOAD ADDRESS (-1) STARTING TO WHICH THE REGISTERS
: SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS
: THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
: WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE
: REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS
: CONTROL TO THE BASE MACHINE.
:
: THE REGARETERS IS THEN CHECKED TO MAKE SURE THAT THE
: DATA FROM THE WCS ARRAY WAS LOADED CORRECTLY. IF NOT,
: AN ERROR IS REPORTED.
:
: THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
: A RESULT OF SOME FAULT CONDITION, THE CONTROL
: IS NOT TRANSFERRED BACK TO THE BASE MACHINE,
: THE WATCH DOG WILL TIME OUT AND A TRAP TO
: 114 WILL OCCUR.
:
: FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
:
:   TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
:   BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
:   CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
:   THE WCS- (NUA<9: 0>, ARRAY ADDRESS MUX-A-PORT,
:   BUS-U-MUX-B PORT, BUS-U-LINES).

```

```

: *****
: TST75: SCOPE
: MOV #TST76,NXTST ; STARTING ADDRESS OF NEXT TEST
:
: JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
: 6600 ; CONTROL BLOCK LOADED HERE
: MLSF01 ; FROM THIS MEMORY LOCATION
: UDISP ; ENTRY POINT IN WCS
:

```

```

030730 000004
030732 012737 031176 003056
030740 004537 037472
030744 006600
030746 057766
030750 006002

```



```

8094 ;CLEAR -ALL- WFP AC'S
8095 030752 012705 003144 MOV #FPO5XX,R5 ;TABLE OF MED CODES
8096 030756 012704 000030 MOV #24.,R4 ;
8097 030762 112537 031000 MOV (R5)+,12% ;GET A CODE
8098 030766 152737 000200 031000 11%: BIS #200,12% ;MAKE IT 'WRITE'
8099 030774 005000 CLR R0 ;WRITE ZEROES
8100 030776 076600 MED ;DO IT
8101 031000 000000 .WORD 0 ;
8102 031002 077411 SOB R4,11% ;LOOP ON ALL
8103 ;
8104 ;SETUP LS ARRAY WITH DATA TO BE LOADED = MED CODE OF WFP REGISTER
8105 031004 012705 003144 MOV #FPO1XX,R5 ;TABLE OF CODES
8106 031010 012704 000010 MOV #8.,R4 ;
8107 031014 012702 001233 MOV #1233,R2 ;LS ARRAY ADDRESS
8108 031020 010277 152012 MOV R2,@WCSAR ;
8109 031024 112500 13%: MOV (R5)+,R0 ;GET MED CODE
8110 031026 052700 052000 BIS #052000,R0 ;DATA = 052MED
8111 031032 010077 152002 MOV R0,@WCSDR ;STORE INTO ARRAY
8112 031036 005277 151774 INC @WCSAR ;NEXT
8113 031042 077410 SOB R4,13% ;LOOP
8114 ;
8115 031044 012737 031160 000114 MOV #39%,@#114 ;SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
8116 031052 012777 001000 151754 MOV #MAINT,@WCSST ;SET UP MAINT BIT, TIMER WILL START WHEN
8117 ;WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
8118 ;
8119 031060 076700 XFCUDIS ;TRANSFER CONTROL TO WCS
8120 031062 1%: ;SHOULD RETURN FROM WCS TO THIS POINT
8121 ;
8122 031062 042777 001000 151744 BIC #MAINT,@WCSST ;SHUT OFF THE TIMER
8123 031070 012737 003344 000114 MOV #BADPAR,@#114 ;RESET UNEXPECTED PARITY HANDLER
8124 ;
8125 031076 012705 003144 MOV #FPO1XX,R5 ;PTR TO MED CODE TABLE, ALSO DATA
8126 031102 012704 000010 MOV #8.,R4 ;# REGISTERS TO RESTORE
8127 031106 012702 001233 MOV #1233,R2 ;LS ARRAY ADDRESS
8128 ;
8129 031112 112501 20%: MOV (R5)+,R1 ;GET DATA FROM TABLE
8130 031114 052701 052000 BIS #052000,R1 ;
8131 031120 110137 031126 MOV R1,22% ;AS READ MED CODE
8132 031124 076600 MED ;GET WFP AC DATA
8133 031126 000000 22%: .WORD 0 ;
8134 031130 020100 CMP R1,R0 ;DATA MATCH ???
8135 031132 001410 BEQ 21% ;BR IF OK
8136 031134 116537 177777 001162 MOVB -1(R5),%REG0 ;WFP AC MED CODE
8137 031142 010137 001164 MOV R1,%REG1 ;EXP'D DATA
8138 031146 010037 001166 MOV R0,%REG2 ;RCV'D DATA
8139 031152 104120 ERROR 120 ;TELL ABOUT ERROR
8140 ;
8141 031154 077422 21%: SOB R4,20% ;LOOP FOR ALL
8142 ;
8143 031156 000407 BR TST76 ;EXIT
8144 ;
8145 ;
8146 ;ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
8147 ;A BLOCK MOVE FUNCTION IN THE WCS.
8148 031160 012777 000020 151646 39%: MOV #PARDIS,@WCSST ;
8149 031166 022626 CMP (SP)+,(SP)+ ;

```

8150 031170 005077 151640  
8151 031174 104121  
8152  
8153  
8154  
8155  
8156  
8157  
8158  
8159  
8160  
8161  
8162  
8163  
8164  
8165  
8166  
8167  
8168  
8169  
8170  
8171  
8172  
8173  
8174  
8175  
8176  
8177  
8178  
8179  
8180  
8181  
8182  
8183  
8184  
8185  
8186  
8187  
8188  
8189  
8190  
8191  
8192  
8193  
8194  
8195  
8196  
8197  
8198  
8199  
8200  
8201  
8202  
8203  
8204  
8205

CLR @WCSST ;  
ERROR 121 ;  
;WCS WATCH DOG TIMED OUT  
;UPON TRANSFERRING CONTROL FROM  
;BASE MACHINE TO WCS MICROCODE.  
;NORMALLY, AFTER UCODE EXECUTION CONTROL  
;SHOULD BE RETURNED TO BASE MACHINE.

\*\*\*\*\*  
\*TEST 76 CHECK BLOCK LOAD OF WFP.AC 2:3 FROM WCS

THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:

WCS.ARRAY -> WFP.AC 2:3

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:

WFP.AC0 2 , WFP.AC1 2 , WFP.AC2 2 , WFP.AC3 2 ,  
WFP.AC0 3 , WFP.AC1 3 , WFP.AC2 3 , WFP.AC3 3

THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN IT.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL LOAD ADDRESS (-1) STARTING TO WHICH THE REGISTERS  
SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE REGARETERS IS THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS ARRAY WAS LOADED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9:0> ARRAY ADDRESS MUX-A-PORT.  
BUS-U-MUX-B PORT, BUS-U-LINES).



```

8206
8207 ;*****
8208 031176 000004 TST76: SCOPE
8209 031200 012737 031444 003056 MOV #TST77,NXTST ; STARTING ADDRESS OF NEXT TEST
8210
8211 031206 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
8212 031212 006620 6620 ; CONTROL BLOCK LOADED HERE
8213 031214 060066 ML SF23 ; FROM THIS MEMORY LOCATION
8214 031216 006002 UDISP ; ENTRY POINT IN WCS
8215
8216 ; CLEAR -ALL- WFP AC'S
8217 031220 012705 003144 MOV #FP05XX,R5 ; TABLE OF MED CODES
8218 031224 012704 000030 MOV #24,R4 ;
8219 031230 112537 031246 115: MOVB (R5)+,125 ; GET A CODE
8220 031234 152737 000200 031246 BISB #200,125 ; MAKE IT 'WRITE'
8221 031242 005000 CLR R0 ; WRITE ZEROES
8222 031244 076600 MED ; DO IT
8223 031246 000000 125: .WORD 0 ;
8224 031250 077411 SOB R4,115 ; LOOP ON ALL
8225
8226 ; SETUP LS ARRAY WITH DATA TO BE LOADED = MED CODE OF WFP REGISTER
8227 031252 012705 003154 MOV #FP23XX,R5 ; TABLE OF CODES
8228 031256 012704 000010 MOV #8,R4 ;
8229 031262 012702 001212 MOV #1212,R2 ; LS ARRAY ADDRESS
8230 031266 010277 151544 MOV R2,@WCSAR ;
8231 031272 112500 135: MOVB (R5)+,R0 ; GET MED CODE
8232 031274 052700 052000 BIS #052000,R0 ; DATA = 052MED
8233 031300 010077 151534 MOV R0,@WCSDR ; STORE INTO ARRAY
8234 031304 005277 151526 INC @WCSAR ; NEXT
8235 031310 077410 SOB R4,135 ; LOOP
8236
8237 031312 012737 031426 000114 MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
8238 031320 012777 001000 151506 MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
8239 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
8240
8241 031326 076700 XFCUDIS ; TRANSFER CONTROL TO WCS
8242 031330 15: ; SHOULD RETURN FROM WCS TO THIS POINT
8243
8244 031330 042777 001000 151476 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
8245 031336 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
8246
8247 031344 012705 003154 MOV #FP23XX,R5 ; PTR TO MED CODE TABLE, ALSO DATA
8248 031350 012704 000010 MOV #8,R4 ; # REGISTERS TO RESTORE
8249 031354 012702 001212 MOV #1212,R2 ; LS ARRAY ADDRESS
8250
8251 031360 112501 205: MOVB (R5)+,R1 ; GET DATA FROM TABLE
8252 031362 052701 052000 BIS #052000,R1 ;
8253 031366 110137 031374 MOVB R1,225 ; AS READ MED CODE
8254 031372 076600 MED ; GET WFP AC DATA
8255 031374 000000 225: .WORD 0 ;
8256 031376 020100 CMP R1,R0 ; DATA MATCH ???
8257 031400 001410 BEQ 215 ; BR IF OK
8258 031402 116537 177777 001162 MOVB -1(R5),SREG0 ; WFP AC MED CODE
8259 031410 010137 001164 MOV R1,SREG1 ; EXP'D DATA
8260 031414 010037 001166 MOV R0,SREG2 ; RCV'D DATA
8261 031420 104122 ERROR 122 ; TELL ABOUT ERROR
    
```

```

8262
8263 031422 077422          215: SOB      R4, 205          ; LOOP FOR ALL
8264
8265 031424 000407          BR       TST77          ;; EXIT
8266
8267
8268 ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
8269 ; A BLOCK MOVE FUNCTION IN THE WCS.
8270 031426 012777 000020 151400 395: MOV      #PARDIS, @WCSST
8271 031434 022626          CMP      (SP)+, (SP)+
8272 031436 005077 151372          CLR      @WCSST
8273 031442 104123          ERROR   123
8274 ; WCS WATCH DOG TIMED OUT
8275 ; UPON TRANSFERRING CONTROL FROM
8276 ; BASE MACHINE TO WCS MICROCODE.
8277 ; NORMALLY, AFTER UCODE EXECUTION CONTROL
8278 ; SHOULD BE RETURNED TO BASE MACHINE.
8279
8280

```

```

8281 ; *****
8282 ; *TEST 77      CHECK BLOCK LOAD OF WFP.AC 4: 5 FROM WCS
8283
8284 ; THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:
8285
8286 ;           WCS.ARRAY -> WFP.AC 4: 5
8287
8288 ; WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
8289 ; GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.
8290
8291 ; THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:
8292
8293 ;           WFP.AC0 4 , WFP.AC1 4 , WFP.AC2 4 , WFP.AC3 4 ,
8294 ;           WFP.AC0 5 , WFP.AC1 5 , WFP.AC2 5 , WFP.AC3 5
8295
8296 ; THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING
8297 ; UNIQUE DATA PATTERNS IN IT.
8298
8299 ; THE MICRO-CODE IS LOADED INTO THE WCS. USING
8300 ; AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
8301 ; TO THE MICRO-CODE IN THE WCS.
8302
8303 ; FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),
8304 ; THE LOCAL LOAD ADDRESS (-1) STARTING TO WHICH THE REGISTERS
8305 ; SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS
8306 ; THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
8307 ; WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE
8308 ; REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS
8309 ; CONTROL TO THE BASE MACHINE.
8310
8311 ; THE REGARETERS IS THEN CHECKED TO MAKE SURE THAT THE
8312 ; DATA FROM THE WCS ARRAY WAS LOADED CORRECTLY. IF NOT,
8313 ; AN ERROR IS REPORTED.
8314
8315 ; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
8316 ; A RESULT OF SOME FAULT CONDITION, THE CONTROL
8317 ; IS NOT TRANSFERRED BACK TO THE BASE MACHINE.

```



```
8318 ; THE WATCH DOG WILL TIME OUT AND A TRAP TO
8319 ; 114 WILL OCCUR.
8320 ;
8321 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
8322 ;
8323 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
8324 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
8325 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
8326 ; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
8327 ; BUS-U-MUX-B PORT, BUS-U-LINES).
8328 ;
8329 ; *****
8330 031444 000004 TST77: SCOPE
8331 031446 012737 031712 003056 MOV #TST100,NXTST ; STARTING ADDRESS OF NEXT TEST
8332 ;
8333 031454 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
8334 031460 006640 6640 ; CONTROL BLOCK LOADED HERE
8335 031462 060166 ML$F45 ; FROM THIS MEMORY LOCATION
8336 031464 006002 UDISP ; ENTRY POINT IN WCS
8337 ;
8338 ; CLEAR -ALL- WFP AC'S
8339 031466 012705 003144 MOV #FP05XX,R5 ; TABLE OF MED CODES
8340 031472 012704 000030 MOV #24,R4 ;
8341 031476 112537 031514 11$: MOV#B (R5)+,125 ; GET A CODE
8342 031502 152737 000200 031514 BISB #200,125 ; MAKE IT 'WRITE'
8343 031510 005000 CLR R0 ; WRITE ZEROES
8344 031512 076600 MED ; DO IT
8345 031514 000000 12$: .WORD 0 ;
8346 031516 077411 SOB R4,115 ; LOOP ON ALL
8347 ;
8348 ; SETUP LS ARRAY WITH DATA TO BE LOADED = MED CODE OF WFP REGISTER
8349 031520 012705 003164 MOV #FP45XX,R5 ; TABLE OF CODES
8350 031524 012704 000010 MOV #8,R4 ;
8351 031530 012702 001414 MOV #1414,R2 ; LS ARRAY ADDRESS
8352 031534 010277 151276 MOV R2,@WCSAR ;
8353 031540 112500 13$: MOV#B (R5)+,R0 ; GET MED CODE
8354 031542 052700 052000 BIS #052000,R0 ; DATA = 052MED
8355 031546 010077 151266 MOV R0,@WCSOR ; STORE INTO ARRAY
8356 031552 005277 151260 INC @WCSAR ; NEXT
8357 031556 077410 SOB R4,135 ; LOOP
8358 ;
8359 031560 012737 031674 000114 MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
8360 031566 012777 001000 151240 MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
8361 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
8362 ;
8363 031574 076700 15$: XFCUDIS ; TRANSFER CONTROL TO WCS
8364 031576 ; SHOULD RETURN FROM WCS TO THIS POINT
8365 ;
8366 031576 042777 001000 151230 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
8367 031604 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
8368 ;
8369 031612 012705 003164 MOV #FP45XX,R5 ; PTR TO MED CODE TABLE, ALSO DATA
8370 031616 012704 000010 MOV #8,R4 ; # REGISTERS TO RESTOPE
8371 031622 012702 001414 MOV #1414,R2 ; LS ARRAY ADDRESS
8372 ;
8373 031626 112501 20$: MOV#B (R5)+,R1 ; GET DATA FROM TABLE
```

```

8374 031630 052701 052000 BIS #052000,R1 ;
8375 031634 110137 031642 MOVB R1,22$ ;AS READ MED CODE
8376 031640 076600 MED ;GET WFP AC DATA
8377 031642 000000 22$: .WORD 0 ;
8378 031644 020100 CMP R1,R0 ;DATA MATCH ???
8379 031646 001410 BEQ 21$ ;BR IF OK ...
8380 031650 116537 177777 001162 MOVB -1(R5),SREG0 ;WFP AC MED CODE
8381 031656 010137 001164 MOV R1,SREG1 ;EXP'D DATA
8382 031662 010037 001166 MOV R0,SREG2 ;RCV'D DATA
8383 031666 104124 ERROR 124 ;TELL ABOUT ERROR
8384 ;
8385 031670 077422 21$: SOB R4,20$ ;LOOP FOR ALL
8386 ;
8387 031672 000407 BR TST100 ;;EXIT
8388 ;
8389 ;
8390 ;ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
8391 ;A BLOCK MOVE FUNCTION IN THE WCS.
8392 031674 012777 000020 151132 39$: MOV #PARDIS,@WCSST ;
8393 031702 022626 CMP (SP)+,(SP)+ ;
8394 031704 005077 151124 CLR @WCSST ;
8395 031710 104125 ERROR 125 ;
8396 ;WCS WATCH DOG TIMED OUT
8397 ;UPON TRANSFERRING CONTROL FROM
8398 ;BASE MACHINE TO WCS MICROCODE.
8399 ;NORMALLY, AFTER UCODE EXECUTION CONTROL
8400 ;SHOULD BE RETURNED TO BASE MACHINE.
8401
8402
8403
8404
8405

```

```

8406 *****
8407 *TEST 100 CHECK BLOCK STORE OF ASPAD 00:37 TO WCS
8408 ;
8409 ; THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:
8410 ;
8411 ; ASPAD 00:37 -> WCS.ARRAY
8412 ;
8413 ; WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
8414 ; GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.
8415 ;
8416 ; THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:
8417 ;
8418 ; ASPLO 00:17 , ASPHI 00:17
8419 ;
8420 ; THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING
8421 ; UNIQUE DATA PATTERNS IN THEM.
8422 ;
8423 ; THE MICRO-CODE IS LOADED INTO THE WCS. USING
8424 ; AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED
8425 ; TO THE MICRO-CODE IN THE WCS.
8426 ;
8427 ; FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),
8428 ; THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS
8429 ; SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS

```



```

8430 ; THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM
8431 ; WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE
8432 ; REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS
8433 ; CONTROL TO THE BASE MACHINE.
8434 ;
8435 ; THE WCS IS THEN CHECKED TO MAKE SURE THAT THE
8436 ; DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,
8437 ; AN ERROR IS REPORTED.
8438 ;
8439 ; THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS
8440 ; A RESULT OF SOME FAULT CONDITION, THE CONTROL
8441 ; IS NOT TRANSFERRED BACK TO THE BASE MACHINE,
8442 ; THE WATCH DOG WILL TIME OUT AND A TRAP TO
8443 ; 114 WILL OCCUR.
8444 ;
8445 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
8446 ;
8447 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
8448 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
8449 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
8450 ; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
8451 ; BUS-U-MUX-B PORT, BUS-U-LINES).
8452 ;
8453 ; *****
8454 031712 000004 TST100: SCOPE
8455 031714 012737 032314 003056 MOV #TST101,NXTST ; STARTING ADDRESS OF NEXT TEST
8456 ;
8457 001234 ADDR. 1= 1234 ; LS ADDR FOR ASPAD STORE
8458 052000 PAT. A= 052000 ; BASE PATTERN FOR DATA
8459 ;
8460 031722 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
8461 031726 006670 6670 ; CONTROL BLOCK LOADED HERE
8462 031730 060326 MASPLS ; FROM THIS MEMORY LOCATION
8463 031732 006002 UDISP ; ENTRY POINT IN WCS
8464 ;
8465 031734 012737 032276 000114 MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
8466 031742 012777 001000 151064 MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
8467 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
8468 ;
8469 ;
8470 ; INIT ASPAD REGISTERS WITH DATA, DATA = '052MED'
8471 031750 012702 031776 MOV #405,R2 ; PTR TO MED TABLE FOR THIS TEST
8472 031754 112200 115: MOVB (R2)+,R0 ; GET A MED CODE
8473 031756 100425 BMI 135 ; END OF TABLE ??
8474 031760 052700 052200 BIS #PAT.A+BIT7,R0 ; MAKE DATA & WRITE MED CODE
8475 031764 110037 031772 MOVB R0,125 ; STORE WRITE-MED CODE
8476 031770 076600 MED ;
8477 031772 000000 125: .WORD 0 ;
8478 031774 000767 BR 115 ;
8479 ;
8480 ; TABLE FOR INIT:
8481 031776 001 003 004 405: .BYTE 01,03,04,05 ; EXCLUDE: R0(00),R2(02),KSP(06),PC(07)
8482 032001 005 ;
8483 032002 010 011 012 .BYTE 10,11,12,13,14,15,16,17 ;
8484 032005 013 014 015 ;
8485 032010 016 017 ;

```

```

8486 032012 020 021 023 .BYTE 20,21,23,24,25,27 ;EXCLUDE: WHAMI(22), CNSL.SW(26)
8487 032015 024 025 027
8488 032020 030 031 032 .BYTE 30,31,32,33,34,35,36,37 ;
8489 032023 033 034 035
8490 032026 036 037
8491 032030 377 .BYTE -1 ; END
8492 032032 032032 .EVEN
8493
8494 032032 012702 001234 135: MOV #ADDR.1,R2 ;R2 IS LS ADDR PTR
8495
8496 032036 010637 032212 MOV SP,415+(2*006) ;GET KSP INTO RESULT TABLE
8497
8498 032042 076600 000022 MED ,022 ;GET WHAMI INTO RESULT TABLE
8499 032046 010037 032242 MOV RO,415+(2*022) ;
8500
8501 032052 076600 000026 MED ,026 ;GET CNSL.SW INTO RESULT TABLE
8502 032056 010037 032252 MOV RO,415+(2*026) ;
8503
8504 032062 012700 052200 MOV #PAT.A+BIT7+00,RO ;ETUP RO W/ 052MED PATTERN
8505
8506 032066 076700 XFCUDIS ;TRANSFER CONTROL TO WCS
8507 032070 15: ;SHOULD RETURN FROM WCS TO THIS POINT
8508
8509 032070 042777 001000 150736 BIC #MAINT,@WCSST ;SHUT OFF THE TIMER
8510 032076 012737 003344 000114 MOV #BADPAR,@#114 ;RESET UNEXPECTED PARITY HANDLER
8511
8512 ;CHECK RESULTS STORED INTO WCS ARRAY
8513 032104 012705 032176 MOV #415,R5 ;PTR TO RESULT TABLE
8514 032110 012704 000040 MOV #32,R4 ;32 REGISTERS SHOULD HAVE BEEN STORED
8515 032114 005037 001162 CLR $REG0 ;HOLDS MED CODE OF ASPAD LOC
8516 032120 012777 001234 150710 MOV #ADDR.1,@WCSAR ;POINT TO WHERE DATA SHOULD BE
8517
8518 032126 017737 150706 001170 205: MOV @WCSDR,$REG3 ;GET DATA FROM LS (RECEIVED)
8519 032134 012537 001166 MOV (R5)+,$REG2 ;GET DATA FROM TABLE (EXPECTED)
8520 032140 001410 BEQ 215 ;IF DATA WASNT FILLED IN, ITS INDETERMINATE
8521 032142 023737 001166 001170 CMP $REG2,$REG3 ;E-DATA = R-DATA ??
8522 032150 001404 BEQ 215 ;BR IF A-OK
8523 032152 017737 150660 001164 MOV @WCSAR,$REG1 ;GET WCS LOC
8524 032160 104126 ERROR 126 ;DATA ERR ? !!
8525
8526 032162 005237 001162 215: INC $REG0 ;NEXT MED CODE
8527 032166 005277 150644 INC @WCSAR ;ADDRESS NEXT LS LOC
8528 032172 077423 SOB R4,205 ;LOOP
8529
8530 032174 000447 BR TST101 ;EXIT
8531
8532
8533 ;---RESULT TABLE---
8534 032176 052200 415: .WORD PAT.A+BIT7+00 ;R0
8535 032200 052201 .WORD PAT.A+BIT7+01 ;R1
8536 032202 001234 .WORD ADDR.1 ;R2 = LS ADDR
8537 032204 052203 .WORD PAT.A+BIT7+03 ;R3
8538 032206 052204 .WORD PAT.A+BIT7+04 ;R4
8539 032210 052205 .WORD PAT.A+BIT7+05 ;R5
8540 032212 123456 .WORD 123456 ;KSP = SP STORED HERE FOR COMPARE
8541 032214 032070 .WORD 15 ;PC = RETURN PC HERE

```



```

8542 032216 052210 .WORD PAT. A+BIT7+10 ;FAC3 0
8543 032220 052211 .WORD PAT. A+BIT7+11 ;FAC3 1
8544 032222 052212 .WORD PAT. A+BIT7+12 ;FAC3 2
8545 032224 052213 .WORD PAT. A+BIT7+13 ;FAC3 3
8546 032226 052214 .WORD PAT. A+BIT7+14 ;FAC3 4
8547 032230 052215 .WORD PAT. A+BIT7+15 ;FAC3 5
8548 032232 052216 .WORD PAT. A+BIT7+16 ;USER SP
8549 032234 052217 .WORD PAT. A+BIT7+17 ;FDST3
8550 032236 052220 .WORD PAT. A+BIT7+20 ;WCSA 0
8551 032240 052221 .WORD PAT. A+BIT7+21 ;WCSA 1
8552 032242 123456 .WORD 123456 ;WHAMI STORED HERE
8553 032244 052223 .WORD PAT. A+BIT7+23 ;CNLS.TMPSW
8554 032246 000000 .WORD 000000 ;R SRC X,1 **INDETERMINATE**
8555 032250 000000 .WORD 000000 ;R DST X,1 **INDETERMINATE**
8556 032252 123456 .WORD 123456 ;CNLS.SW STORED HERE
8557 032254 052227 .WORD PAT. A+BIT7+27 ;CNLS.CADR
8558 032256 052230 .WORD PAT. A+BIT7+30 ;FAC1 0
8559 032260 052231 .WORD PAT. A+BIT7+31 ;FAC1 1
8560 032262 052232 .WORD PAT. A+BIT7+32 ;FAC1 2
8561 032264 052233 .WORD PAT. A+BIT7+33 ;FAC1 3
8562 032266 052234 .WORD PAT. A+BIT7+34 ;FAC1 4
8563 032270 052235 .WORD PAT. A+BIT7+35 ;FAC1 5
8564 032272 052236 .WORD PAT. A+BIT7+36 ;FPSHI/FEC
8565 032274 052237 .WORD PAT. A+BIT7+37 ;FDST1/R IR
8566 . ; END OF TABLE
8567 .
8568 .

```

```

8569 . ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
8570 . ; A BLOCK MOVE FUNCTION IN THE WCS.
8571 032276 012777 000020 150530 395: MOV #PARDIS,@WCSST ;
8572 032304 022626 CMP (SP)+,(SP)+ ;
8573 032306 005077 150522 CLR @WCSST ;
8574 032312 104127 ERROR 127 ; !!! WATCH DOG TIMEOUT !!!
8575 . ; WCS WATCH DOG TIMED OUT
8576 . ; UPON TRANSFERRING CONTROL FROM
8577 . ; BASE MACHINE TO WCS MICROCODE.
8578 . ; NORMALLY, AFTER UCODE EXECUTION CONTROL
8579 . ; SHOULD BE RETURNED TO BASE MACHINE.
8580 .
8581 .

```

```

8582 . ; *****
8583 . ; *TEST 101 CHECK BLOCK STORE OF BSPAD 00:37 TO WCS
8584 .
8585 . ; THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:
8586 .
8587 . ; BSPAD 00:37 -> WCS.ARRAY
8588 .
8589 . ; WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
8590 . ; GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.
8591 .
8592 . ; THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:
8593 .
8594 . ; BSPLO 00:17 , BSPHI 00:17
8595 .
8596 . ; THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING
8597 . ; UNIQUE DATA PATTERNS IN THEM.

```



8598  
8599  
8600  
8601  
8602  
8603  
8604  
8605  
8606  
8607  
8608  
8609  
8610  
8611  
8612  
8613  
8614  
8615  
8616  
8617  
8618  
8619  
8620  
8621  
8622  
8623  
8624  
8625  
8626  
8627  
8628  
8629  
8630  
8631  
8632  
8633  
8634  
8635  
8636  
8637  
8638  
8639  
8640  
8641  
8642  
8643  
8644  
8645  
8646  
8648  
8649  
8650  
8651  
8652  
8653

032314 000004  
032316 012737 032710 003056  
001433  
124000  
032324 004537 037472  
032330 006710  
032332 060426  
032334 006002  
032336 012737 032672 000114  
032344 012777 001000 150462  
032352 012702 032400  
032356 112200  
032360 100425  
032362 052700 124200  
032366 110037 032374  
032372 076600  
032374 000000

THE MICRO-CODE IS LOADED INTO THE WCS. USING AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE), THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS CONTROL TO THE BASE MACHINE.

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT, AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS A RESULT OF SOME FAULT CONDITION, THE CONTROL IS NOT TRANSFERRED BACK TO THE BASE MACHINE, THE WATCH DOG WILL TIME OUT AND A TRAP TO 114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT, BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*  
TST101: SCOPE  
MOV #TST102,NXTST ; STARTING ADDRESS OF NEXT TEST  
ADDR. 1= 1433 ; LS ADDR FOR BSPAD STORE  
PAT B= 124000 ; BASE PATTERN FOR DATA  
JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE  
6710 ; CONTROL BLOCK LOADED HERE  
MBSPLS ; FROM THIS MEMORY LOCATION  
UDISP ; ENTRY POINT IN WCS  
MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER  
MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN  
; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS  
; INIT BSPAD REGISTERS WITH DATA, DATA = '124MED'  
MOV #405,R2 ; PTR TO MED TABLE FOR THIS TEST  
115: MOV (R2)+,R0 ; GET A MED CODE  
BMI 135 ; END OF TABLE ??  
BIS #PAT.B+BIT7,R0 ; MAKE DATA & WRITE MED CODE  
MOV R0,125 ; STORE WRITE-MED CODE  
MED  
125: .WORD 0



```

8654 032376 000767 BR 115 ;
8655 ;
8656 ; TABLE FOR INIT:
8657 032400 041 043 044 405: . BYTE 41,43,44,45 ; EXCLUDE: RO(40),R2(42),KSP(46),PC(47)
8658 032403 045
8659 032404 050 051 052 . BYTE 50,51,52,53,54,55,56,57 ;
8660 032407 053 054 055
8661 032412 056 057
8662 032414 060 061 062 . BYTE 60,61,62,64,65,66 ; EXCLUDE: R ZERO (63), CNSL.CNTL(67)
8663 032417 064 065 066
8664 032422 070 071 072 . BYTE 70,71,72,73,74,75,76,77 ;
8665 032425 073 074 075
8666 032430 076 077
8667 032432 377 . BYTE -1 ; END
8668 032434 . EVEN
8669 ;
8670 032434 012702 001433 135: MOV #ADDR. 1,R2 ; R2 IS LS ADDR PTR
8671 ;
8672 032440 010637 032606 MOV SP,415+(2*006) ; GET KSP INTO RESULT TABLE
8673 ;
8674 032444 076600 000067 MED ,067 ; GET CNSL.CNTL INTO RESULT TABLE
8675 032450 010037 032650 MOV RO,415+(2*027) ;
8676 ;
8677 032454 012700 124240 MOV #PAT. B+BIT7+40,RO ; SETUP RO W/ 124MED PATTERN
8678 ;
8679 032460 076700 XFCUDIS ; TRANSFER CONTROL TO WCS
8680 032462 15: ; SHOULD RETURN FROM WCS TO THIS POINT
8681 ;
8682 032462 042777 001000 150344 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
8683 032470 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
8684 ;
8685 ; CHECK RESULTS STORED INTO WCS ARRAY
8686 032476 012705 032572 MOV #415,R5 ; PTR TO RESULT TABLE
8687 032502 012704 000040 MOV #32,R4 ; 32. REGISTERS SHOULD HAVE BEEN STORED
8688 032506 012737 000040 001162 MOV #040,$REG0 ; HOLDS MED CODE OF BSPAD LOC
8689 032514 012777 001433 150314 MOV #ADDR. 1,@WCSAR ; POINT TO WHERE DATA SHOULD BE
8690 ;
8691 032522 017737 150312 001170 205: MOV @WCSDR,$REG3 ; GET DATA FROM LS (RECEIVED)
8692 032530 012537 001166 MOV (R5)+,$REG2 ; GET DATA FROM TABLE (EXPECTED)
8693 032534 001410 BEQ 215 ; IF DATA WASNT FILLED IN, ITS INDETERMINATE
8694 032536 023737 001166 001170 CMP $REG2,$REG3 ; E-DATA = R-DATA ??
8695 032544 001404 BEQ 215 ; BR IF A-OK
8696 032546 017737 150264 001164 MOV @WCSAR,$REG1 ; GET WCS LOC
8697 032554 104130 ERROR 130 ; DATA ERROR !!
8698 ;
8699 032556 005237 001162 215: INC $REG0 ; NEXT MED CODE
8700 032562 005277 150250 INC @WCSAR ; ADDRESS NEXT LS LOC
8701 032566 077423 SOB R4,205 ; LOOP
8702 ;
8703 032570 000447 BR TST102 ; EXIT
8704 ;
8705 ;
8706 ; ---RESULT TABLE---
8707 032572 124240 415: . WORD PAT. B+BIT7+40 ; RO
8708 032574 124241 . WORD PAT. B+BIT7+41 ; R1
8709 032576 001433 . WORD ADDR. 1 ; R2 = LS ADDR

```

```

8710 032600 124243 .WORD PAT.B+BIT7+43 ;R3
8711 032602 124244 .WORD PAT.B+BIT7+44 ;R4
8712 032604 124245 .WORD PAT.B+BIT7+45 ;R5
8713 032606 123456 .WORD 123456 ;KSP = SP STORED HERE FOR COMPARE
8714 032610 032462 .WORD 15 ;PC = RETURN PC HERE
8715 032612 124250 .WORD PAT.B+BIT7+50 ;FAC2 0
8716 032614 124251 .WORD PAT.B+BIT7+51 ;FAC2 1
8717 032616 124252 .WORD PAT.B+BIT7+52 ;FAC2 2
8718 032620 124253 .WORD PAT.B+BIT7+53 ;FAC2 3
8719 032622 124254 .WORD PAT.B+BIT7+54 ;FAC2 4
8720 032624 124255 .WORD PAT.B+BIT7+55 ;FAC2 5
8721 032626 124256 .WORD PAT.B+BIT7+56 ;USER SP
8722 032630 124257 .WORD PAT.B+BIT7+57 ;FDST2
8723 032632 124260 .WORD PAT.B+BIT7+60 ;WCSB 0
8724 032634 124261 .WORD PAT.B+BIT7+61 ;WCSB 1
8725 032636 124262 .WORD PAT.B+BIT7+62 ;R VECT HERE - HOPEFULLY NO INTR ??
8726 032640 000000 .WORD 000000 ;R ZERO
8727 032642 000000 .WORD 000000 ;R SRC X,1 **INDETERMINATE**
8728 032644 000000 .WORD 000000 ;R DST X,1 **INDETERMINATE**
8729 032646 124266 .WORD PAT.B+BIT7+66 ;FPA
8730 032650 123456 .WORD 123456 ;CNLS.CNTL STORED HERE
8731 032652 124270 .WORD PAT.B+BIT7+70 ;FAC0 0
8732 032654 124271 .WORD PAT.B+BIT7+71 ;FAC0 1
8733 032656 124272 .WORD PAT.B+BIT7+72 ;FAC0 2
8734 032660 124273 .WORD PAT.B+BIT7+73 ;FAC0 3
8735 032662 124274 .WORD PAT.B+BIT7+74 ;FAC0 4
8736 032664 124275 .WORD PAT.B+BIT7+75 ;FAC0 5
8737 032666 124276 .WORD PAT.B+BIT7+76 ;FEA
8738 032670 124277 .WORD PAT.B+BIT7+77 ;FDST0
8739 . END OF TABLE

```

```

8740
8741
8742 ;ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
8743 ;A BLOCK MOVE FUNCTION IN THE WCS.
8744 032672 012777 000020 150134 395. MOV #PARDIS,@WCSST
8745 032700 022626 CMP (SP)+,(SP)+
8746 032702 005077 150126 CLR @WCSST
8747 032706 104131 ERROR 131 ;!!! WATCH DOG TIMEOUT !!!
8748 ;WCS WATCH DOG TIMED OUT
8749 ;UPON TRANSFERRING CONTROL FROM
8750 ;BASE MACHINE TO WCS MICROCODE.
8751 ;NORMALLY, AFTER UCODE EXECUTION CONTROL
8752 ;SHOULD BE RETURNED TO BASE MACHINE.

```

```

8753
8754
8755 *****
8756 *TEST 102 CHECK BLOCK STORE OF CSPAD 17:00 TO WCS
8757
8758 THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:
8759
8760 CSPAD 17:00 -> WCS.ARRAY
8761
8762 WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS
8763 GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS
8764
8765 THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:

```



8766  
8767  
8768  
8769  
8770  
8771  
8772  
8773  
8774  
8775  
8776  
8777  
8778  
8779  
8780  
8781  
8782  
8783  
8784  
8785  
8786  
8787  
8788  
8789  
8790  
8791  
8792  
8793  
8794  
8795  
8796  
8797  
8798  
8799  
8800  
8801  
8802  
8803  
8804  
8805  
8806  
8807  
8808  
8809  
8810  
8811  
8812  
8813  
8814  
8815  
8816  
8817  
8818  
8819  
8820  
8821

CSP 17 (#1), CSP 16 (#0), CSP 15 (MD), CSP 14 (#2),  
CSP 13, ..., CSP 00

THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN THEM.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS  
SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE STORE OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,  
BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

```

TST102: SCOPE
MOV      #TST103,NXTST      : STARTING ADDRESS OF NEXT TEST
ADDR. 1= 1212              : LS ADDR FOR CSPAD STORE
PAT. C= 070000            : BASE PATTERN FOR DATA
MED      .RDFLAG           : READ FLAGS
BIS      #BIT11,R0         : SET CSP CNST INVALID
MED      .WRFLAG
JSR      R5,LDWCS          : LOAD WCS WITH MICRO CODE
6740
MCSPLS
UDISP
MOV      #395,@#114        : SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
MOV      #MAINT,@WCSST     : SET UP MAINT BIT. TIMER WILL START WHEN
WCS HOT BOX IS SELECTED. UPON TRANSFERRING CONTROL TO WCS

```

```
8822
8823
8824 ; INIT CSPAD REGISTERS WITH DATA, DATA = '070MED'
8825 032762 012702 033010 MOV #405,R2 ; PTR TO MED TABLE FOR THIS TEST
8826 032766 112200 115: MOV (R2)+,R0 ; GET A MED CODE
8827 032770 100416 BMI 135 ; END OF TABLE ??
8828 032772 052700 070200 BIS #PAT.C+BIT7,R0 ; MAKE DATA & WRITE MED CODE
8829 032776 110037 033004 MOV R0,125 ; STORE WRITE-MED CODE
8830 033002 076600 MED
8831 033004 000000 125: .WORD 0
8832 033006 000767 BR 115
8833
8834 ; TABLE FOR INIT:
8835 033010 100 101 102 405: .BYTE 100,101,102,103,104,105,106,107
8836 033013 103 104 105
8837 033016 106 107
8838 033020 110 111 112 .BYTE 110,111,112,113 ; EXCLUDE: CSP 14:17 BASE CNST & MD
8839 033023 113
8840 033024 377 .BYTE -1 ; END
8841 033026 .EVEN
8842
8843 033026 012702 001212 135: MOV #ADDR.1,R2 ; R2 IS LS ADDR PTR
8844
8845 033032 076700 XFCUDIS ; TRANSFER CONTROL TO WCS
8846 033034 15: ; SHOULD RETURN FROM WCS TO THIS POINT
8847
8848 033034 042777 001000 147772 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
8849 033042 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
8850
8851 ; CHECK RESULTS STORED INTO WCS ARRAY
8852 033050 012705 033142 MOV #415,R5 ; PTR TO RESULT TABLE
8853 033054 012704 000020 MOV #16,R4 ; 16. REGISTERS SHOULD HAVE BEEN STORED
8854 033060 012737 000117 001162 MOV #117,$REG0 ; HOLDS MED CODE OF CSPAD LOC
8855 ; NOTE: STORED IN REVERSE ORDER
8856 033066 012777 001212 147742 MOV #ADDR.1,@WCSAR ; POINT TO WHERE DATA SHOULD BE
8857
8858 033074 017737 147740 001170 205: MOV @WCSDR,$REG3 ; GET DATA FROM LS (RECEIVED)
8859 033102 012537 001166 MOV (R5)+,$REG2 ; GET DATA FROM TABLE (EXPECTED)
8860 033106 023737 001166 001170 CMP $REG2,$REG3 ; E-DATA = R-DATA ??
8861 033114 001404 BEQ 215 ; BR IF A-OK
8862 033116 017737 147714 001164 MOV @WCSAR,$REG1 ; GET WCS LOC
8863 033124 104132 ERROR 132 ; DATA ERROR !!
8864
8865 033126 005337 001162 215: DEC $REG0 ; NEXT MED CODE
8866 033132 005277 147700 INC @WCSAR ; ADDRESS NEXT LS LOC
8867 033136 077422 SOB R4,205 ; LOOP
8868
8869 033140 000427 BR TST103 ; EXIT
8870
8871
8872 ; ---RESULT TABLE---
8873 033142 000001 415: .WORD 000001 ; CSP 17 = # 1
8874 033144 000000 .WORD 000000 ; CSP 16 = # 0
8875 033146 076700 .WORD XFCUDIS ; CSP 15 = MD
8876 033150 000002 .WORD 000002 ; CSP 14 = # 2
8877 033152 070313 .WORD PAT.C+BIT7+113 ; CSP 13 = PATTERN
```



8878 033154 070312  
8879 033156 070311  
8880 033160 070310  
8881 033162 070307  
8882 033164 070306  
8883 033166 070305  
8884 033170 070304  
8885 033172 070303  
8886 033174 070302  
8887 033176 070301  
8888 033200 070300

WORD PAT. C+BIT7+112 ;CSP 12 = PATTERN  
WORD PAT. C+BIT7+111 ;CSP 11 = PATTERN  
WORD PAT. C+BIT7+110 ;CSP 10 = PATTERN  
WORD PAT. C+BIT7+107 ;CSP 07 = PATTERN  
WORD PAT. C+BIT7+106 ;CSP 06 = PATTERN  
WORD PAT. C+BIT7+105 ;CSP 05 = PATTERN  
WORD PAT. C+BIT7+104 ;CSP 04 = PATTERN  
WORD PAT. C+BIT7+103 ;CSP 03 = PATTERN  
WORD PAT. C+BIT7+102 ;CSP 02 = PATTERN  
WORD PAT. C+BIT7+101 ;CSP 01 = PATTERN  
WORD PAT. C+BIT7+100 ;CSP 00 = PATTERN

8889  
8890  
8891

ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING  
A BLOCK MOVE FUNCTION IN THE WCS.

8892  
8893 033202 012777 000020 147624 395  
8894 033210 022626  
8895 033212 005077 147616  
8896 033216 104133

MOV #PARDIS, @WCSST ;  
CMP (SP)+, (SP)+ ;  
CLR @WCSST ;  
ERROR 133 ; !!! WATCH DOG TIMEOUT !!!

8897  
8898  
8899  
8900  
8901  
8902  
8903  
8904  
8905  
8906  
8907  
8908

WCS WATCH DOG TIMED OUT  
UPON TRANSFERRING CONTROL FROM  
BASE MACHINE TO WCS MICROCODE.  
NORMALLY, AFTER UCODE EXECUTION CONTROL  
SHOULD BE RETURNED TO BASE MACHINE.

8909  
8910  
8911  
8912  
8913  
8914  
8915  
8916  
8917  
8918  
8919  
8920  
8921  
8922  
8923  
8924  
8925  
8926  
8927  
8928  
8929  
8930  
8931  
8932  
8933

\*\*\*\*\*  
\*TEST 103 CHECK BLOCK LOAD OF ASPAD 00:37 FROM WCS

THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:  
WCS ARRAY -> ASPAD 00:37

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:  
ASPL0 00:17 , ASPH1 00:17

THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS TO IT.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL LOAD ADDRESS (-1) STARTING FROM WHICH THE REGISTERS  
SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM

8934  
8935  
8936  
8937  
8938  
8939  
8940  
8941  
8942  
8943  
8944  
8945  
8946  
8947  
8948  
8949  
8950  
8951  
8952  
8953  
8954  
8955  
8956  
8957  
8958  
8959  
8960  
8961  
8962  
8963  
8964  
8965  
8966  
8967  
8968  
8969  
8970  
8971  
8972  
8973  
8974  
8975  
8976  
8977  
8978  
8979  
8980  
8981  
8982  
8983  
8984  
8985  
8986  
8987  
8988  
8989

033220 000004  
033222 012737 034100 003056  
  
001555  
111000  
  
033230 004537 037472  
033234 006660  
033236 060266  
033240 006002  
  
033242 012777 001555 147566  
033250 012700 111000  
033254 012701 000040  
  
033260 010077 147554 305  
033264 005277 147546  
033270 005200  
033272 077106  
  
033274 024646  
  
033276 076600 000022  
033302 012777 001577 147526  
033310 010077 147524  
033314 010037 033774  
  
033320 076600 000026  
033324 012777 001603 147504  
033332 010077 147502  
033336 010037 034014

WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.  
  
THE REGISTERS ARE THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS WAS LOADED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.  
  
THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.  
  
FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:  
  
TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT  
BUS-U-MUX-B PORT, BUS-U-LINES).  
  
\*\*\*\*\*  
TST103: SCOPE  
MOV #TST104,NXTST ; STARTING ADDRESS OF NEXT TEST  
  
ADDR. 1= 1555 ; LS ADDR FOR ASPAD LOAD  
PAT. A= 111000 ; BASE PATTERN FOR DATA  
  
JSR P5,LDWCS ; LOAD WCS WITH MICRO CODE  
6660 ; CONTROL BLOCK LOADED HERE  
ML\$ASP ; FROM THIS MEMORY LOCATION  
UDISP ; ENTRY POINT IN WCS  
  
; INIT WCS AREA WITH '111MED' DATA PATTERN IN ALL LOCN'S  
MOV #ADDR. 1,@WCSAR ; POINT AT AREA  
MOV #PAT. A+000,R0 ; DATA PATTERN BASE  
MOV #32,R1 ; THIS MANY REGISTERS  
  
305: MOV R0,@WCSAR ; STORE A PATTERN  
INC @WCSAR ; NEXT LOCN  
INC R0 ; NEXT PATTERN  
SOB R1,305 ; LOOP  
  
CMP -(SP),-(SP) ; GET SOME STACK  
  
; INIT SOME SPECIAL LOCN'S  
MED ,022 ; GET CURRENT WHAMI  
MOV #ADDR. 1+022,@WCSAR ; FIND WHERE IT GOES  
MOV R0,@WCSAR ; STORE IT  
MOV R0,435+2 ; ALSO IN TABLE, BELOW  
  
MED ,026 ; GET CURRENT CNSL. SW  
MOV #ADDR. 1+026,@WCSAR ; FIND WHERE IT GOES  
MOV R0,@WCSAR ; STORE IT  
MOV R0,445+2 ; ALSO IN TABLE, BELOW



```

8990
8991 033342 012777 001563 147466      MOV    #ADDR. 1+006, @WCSAR      ; FIND WHERE KSP GOES
8992 033350 010677 147464              MOV    SP, @WCSDR                ; STORE IT
8993 033354 010637 033714              MOV    SP, 425+2                 ; ALSO IN TABLE, BELOW
8994
8995 033360 012777 001564 147450      MOV    #ADDR. 1+007, @WCSAR      ; FIND WHERE PC GOES
8996 033366 012777 033502 147444      MOV    #15, @WCSDR               ; STORE IT
8997
8998                                     ; INIT ASPAD REGISTERS WITH ALL ZEROES
8999 033374 012702 033424              MOV    #405, R2                  ; PTR TO MED TABLE FOR THIS TEST
9000 033400 112200                    115:  MOVB   (R2)+, R0                 ; GET A MED CODE
9001 033402 100426                    BMI    135                       ; END OF TABLE ??
9002 033404 152700 000200              BLSB  #BIT7, R0                  ; MAKE DATA & WRITE MED CODE
9003 033410 110037 033420              MOVB  R0, 125                    ; LOAD WRITE-MED CODE
9004 033414 005000                    CLR   R0                          ; WRITE ZEROES
9005 033416 076600                    MED
9006 033420 000000                    125:  .WORD  0
9007 033422 000766                    BR    115
9008
9009                                     ; TABLE FOR INIT:
9010 033424      001      003      004  405:  .BYTE  01, 03, 04, 05           ; EXCLUDE:  R0(00), R2(02), KSP(06), PC(07)
9011 033427      005
9012 033430      010      011      012      .BYTE  10, 11, 12, 13, 14, 15, 16, 17
9013 033433      013      014      015
9014 033436      016      017
9015 033440      020      021      023      .BYTE  20, 21, 23, 24, 25, 27           ; EXCLUDE:  WHAMI(22), CNSL SW(26)
9016 033443      024      025      027
9017 033446      030      031      032      .BYTE  30, 31, 32, 33, 34, 35, 36, 37
9018 033451      033      034      035
9019 033454      036      037
9020 033456      377      .BYTE  -1                          ; END
9021      033460      EVEN
9022
9023 033460 012702 001555                    135:  MOV    #ADDR. 1, R2              ; R2 IS LS ADDR PTR
9024
9025 033464 012737 034062 000114          MOV    #395, @#114              ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
9026 033472 012777 001000 147334          MOV    #MAINT, @WCSST           ; SET UP MAINT BIT, TIMER WILL START WHEN
9027                                     ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL FROM WCS
9028
9029 033500 076700                    XFCUDIS                          ; TRANSFER CONTROL FROM WCS
9030 033502                    15:                                     ; SHOULD RETURN FROM WCS TO THIS POINT
9031
9032 033502 042777 001000 147324          BIC   #MAINT, @WCSST           ; SHUT OFF THE TIMER
9033 033510 012737 003344 000114          MOV    #BADPAR, @#114          ; RESET UNEXPECTED PARITY HANDLER
9034
9035                                     ; COPY A-SIDE GPR'S TO BOTH A&B-SIDES
9036 033516 010000          MOV    R0, R0
9037 033520 010101          MOV    R1, R1
9038 033522 010202          MOV    R2, R2
9039 033524 010303          MOV    R3, R3
9040 033526 010404          MOV    R4, R4
9041 033530 010505          MOV    R5, R5
9042 033532 010606          MOV    SP, SP
9043
9044                                     ; CHECK RESULTS LOADED INTO ASP REGISTERS
9045 033534 012737 033662 001202          MOV    #415, $TMP0            ; PTR TO RESULT TABLE

```

```

9046 033542 012737 000040 001204      MOV    #32, $TMP1      ; THIS MANY REGISTERS
9047 033550 012737 001555 001164      MOV    #ADDR. 1, $REG1 ; WCS LOC USED
9048
9049 033556 017737 145420 001162 205:    MOV    @$TMP0, $REG0   ; GET MED/READ CODE
9050 033564 062737 000002 001202      ADD    #2, $TMP0      ; AUTO-INCR, THE HARD WAY
9051 033572 113737 001162 033624      MOV    $REG0, 215     ; STORE FOR MED
9052 033600 017737 145376 001166      MOV    @$TMP0, $REG2  ; GET DATA FROM TABLE (EXPECTED)
9053 033606 062737 000002 001202      ADD    #2, $TMP0      ; AUTO-INCR, THE HARD WAY
9054 033614 005737 001162      TST    $REG0          ; IGNORE THIS ENTRY ??
9055 033620 100411      BMI    225           ; YES
9056 033622 076600      MED
9057 033624 000000      215:    .WORD 0              ; GET CSP DATA
9058
9059 033626 010037 001170      MOV    R0, $REG3      ; STORE R-DATA
9060 033632 023737 001170 001166      CMP    $REG3, $REG2   ; E-DATA = R-DATA ??
9061 033640 001401      BEQ    225           ; BR IF A-OK
9062 033642 104134      ERROR 134           ; DATA ERROR !!
9063
9064 033644 005237 001164      225:    INC    $REG1          ; BUMP WCS LOC CNTR
9065 033650 005337 001204      DEC    $TMP1          ; LOOP
9066 033654 003340      BGT    205
9067
9068 033656 022626      CMP    (SP)+, (SP)+   ; RELEASE EXTRA STACK
9069 033660 000507      BR     TST104        ; EXIT
9070
9071
9072

```

---RESULT TABLE---

```

9073 033662 000000 111000      415:    .WORD 000, PAT. A+00 ; R0
9074 033666 000001 111001      .WORD 001, PAT. A+01 ; R1
9075 033672 000002 111002      .WORD 002, PAT. A+02 ; R2
9076 033676 000003 111003      .WORD 003, PAT. A+03 ; R3
9077 033702 000004 111004      .WORD 004, PAT. A+04 ; R4
9078 033706 000005 111005      .WORD 005, PAT. A+05 ; R5
9079 033712 000006 123456      425:    .WORD 006, 123456    ; KSP = SP LOADED HERE FOR COMPARE
9080 033716 177777 033502      .WORD -1, 15         ; PC = RETURN PC HERE
9081 033722 000010 111010      .WORD 010, PAT. A+10 ; FAC3 0
9082 033726 000011 111011      .WORD 011, PAT. A+11 ; FAC3 1
9083 033732 000012 111012      .WORD 012, PAT. A+12 ; FAC3 2
9084 033736 000013 111013      .WORD 013, PAT. A+13 ; FAC3 3
9085 033742 000014 111014      .WORD 014, PAT. A+14 ; FAC3 4
9086 033746 000015 111015      .WORD 015, PAT. A+15 ; FAC3 5
9087 033752 000016 111016      .WORD 016, PAT. A+16 ; USER SP
9088 033756 000017 111017      .WORD 017, PAT. A+17 ; FDST3
9089 033762 000020 111020      .WORD 020, PAT. A+20 ; WCSA 0
9090 033766 000021 111021      .WORD 021, PAT. A+21 ; WCSA 1
9091 033772 000022 123456      435:    .WORD 022, 123456    ; WHAMI LOADED HERE
9092 033776 000023 111023      .WORD 023, PAT. A+23 ; CNSL. TMP SW
9093 034002 177777 000000      .WORD -1, 000000    ; R SRC X, I      **INDETERMINATE**
9094 034006 177777 000000      .WORD -1, 000000    ; R DST X, I      **INDETERMINATE**
9095 034012 000026 123456      445:    .WORD 026, 123456    ; CNSL. SW LOADED HERE
9096 034016 000027 111027      .WORD 027, PAT. A+27 ; CNSL. CADR
9097 034022 000030 111030      .WORD 030, PAT. A+30 ; FAC1 0
9098 034026 000031 111031      .WORD 031, PAT. A+31 ; FAC1 1
9099 034032 000032 111032      .WORD 032, PAT. A+32 ; FAC1 2
9100 034036 000033 111033      .WORD 033, PAT. A+33 ; FAC1 3
9101 034042 000034 111034      .WORD 034, PAT. A+34 ; FAC1 4

```



9102 034046 000035 111035  
9103 034052 000036 111036  
9104 034056 000037 111037  
9105  
9106  
9107  
9108  
9109  
9110 034062 012777 000020 146744 395:  
9111 034070 022626  
9112 034072 005077 146736  
9113 034076 104135  
9114  
9115  
9116  
9117  
9118  
9119  
9120  
9121  
9122  
9123  
9124  
9125  
9126  
9127  
9128  
9129  
9130  
9131  
9132  
9133  
9134  
9135  
9136  
9137  
9138  
9139  
9140  
9141  
9142  
9143  
9144  
9145  
9146  
9147  
9148  
9149  
9150  
9151  
9152  
9153  
9154  
9155  
9156  
9157

WORD 035, PAT. A+35 ; FAC1 5  
WORD 036, PAT. A+36 ; FPSHI/FEC  
WORD 037, PAT. A+37 ; FDST1/R IR  
; END OF TABLE  
;  
;  
; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING  
; A BLOCK MOVE FUNCTION IN THE WCS.  
MOV #PARDIS, @WCSST ;  
CMP (SP)+, (SP)+ ;  
CLR @WCSST ;  
ERROR 135 ; !!! WATCH DOG TIMEOUT !!!  
; WCS WATCH DOG TIMED OUT  
; UPON TRANSFERRING CONTROL FROM  
; BASE MACHINE FROM WCS MICROCODE.  
; NORMALLY, AFTER UCODE EXECUTION CONTROL  
; SHOULD BE RETURNED TO BASE MACHINE.

\*\*\*\*\*  
\*TEST 104 CHECK BLOCK LOAD OF BSPAD 00:37 FROM WCS

THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:

WCS. ARRAY -> BSPAD 00:37

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:

BSPLO 00:17 , BSPHI 00:17

THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS TO IT.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL LOAD ADDRESS (-1) STARTING FROM WHICH THE REGISTERS  
SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE REGISTERS ARE THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS WAS LOADED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO

```

9158      ; 114 WILL OCCUR.
9159      ;
9160      ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
9161      ;
9162      ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
9163      ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
9164      ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
9165      ; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
9166      ; BUS-U-MUX-B PORT, BUS-U-LINES).
9167      ;
9168      ; *****
9169 034100 000004 TST104: SCOPE
9170 034102 012737 034762 003056 MOV #TST105,NXTST ; STARTING ADDRESS OF NEXT TEST
9171      ;
9172      ; ADDR. 1= 1232 ; LS ADDR FOR BSPAD LOAD
9173      ; 050000 ; BASE PATTERN FOR DATA
9174      ;
9175 034110 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
9176 034114 006700 6700 ; CONTROL BLOCK LOADED HERE
9177 034116 060366 MLSBSP ; FROM THIS MEMORY LOCATION
9178 034120 006002 UDISP ; ENTRY POINT IN WCS
9179      ;
9180 034122 024646 CMP -(SP),-(SP) ; GET SOME STACK
9181      ;
9182      ; INIT WCS APEA WITH '050MED' DATA PATTERN IN ALL LOCN'S
9183 034124 012777 001232 146704 MOV #ADDR. 1,@WCSAR ; POINT AT AREA
9184 034132 012700 050040 MOV #PAT. B+040,R0 ; DATA PATTERN BASE
9185 034136 012701 000040 MOV #32,R1 ; THIS MANY REGISTERS
9186      ;
9187 034142 010077 146672 305: MOV R0,@WCSDR ; STORE A PATTERN
9188 034146 005277 146664 INC @WCSAR ; NEXT LOCN
9189 034152 005200 INC R0 ; NEXT PATTERN
9190 034154 077106 SOB R1,305 ; LOOP
9191      ;
9192      ;
9193      ; INIT SOME SPECIAL LOCN'S:
9194 034156 012777 001255 146652 MOV #ADDR. 1+023,@WCSAR ; FIND WHERE R ZERO GOES
9195 034164 005077 146650 CLR @WCSDR ; STORE ZEROES THERE ...
9196      ;
9197 034170 076600 000067 MED ,067 ; GET CURRENT CNSL. CNTL
9198 034174 012777 001261 146634 MOV #ADDR. 1+027,@WCSAR ; FIND WHERE IT GOES
9199 034202 010077 146632 MOV R0,@WCSDR ; STORE IT
9200 034206 010037 034702 MOV R0,435+2 ; ALSO IN TABLE, BELOW
9201      ;
9202 034212 012777 001240 146616 MOV #ADDR. 1+006,@WCSAR ; FIND WHERE KSP GOES
9203 034220 010677 146614 MOV SP,@WCSDR ; STORE IT
9204 034224 010637 034576 MOV SP,425+2 ; ALSO IN TABLE, BELOW
9205      ;
9206 034230 012777 001241 146600 MOV #ADDR. 1+007,@WCSAR ; FIND WHERE PC GOES
9207 034236 012777 034352 146574 MOV #15,@WCSDR ; STORE IT
9208      ;
9209      ; INIT BSPAD REGISTERS WITH ALL ZEROES
9210 034244 012702 034274 MOV #405,R2 ; PTR TO MED TABLE FOR THIS TEST
9211 034250 112200 115: MOVB (R2)+,R0 ; GET A MED CODE
9212 034252 100426 BMI 135 ; END OF TABLE ??
9213 034254 152700 000200 BISB #BIT7,R0 ; MAKE DATA & WRITE MED CODE

```



9214	034260	110037	034270			MOV	RO, 125		; LOAD WRITE-MED CODE
9215	034264	005000				CLR	RO		; WRITE ZEROES
9216	034266	076600				MED			
9217	034270	000000			125:	WORD	0		
9218	034272	000766				BR	115		
9219									
9220									
9221	034274	041	043	044	405:	; TABLE FOR INIT:			
9222	034277	045				. BYTE	41, 43, 44, 45		; EXCLUDE: RO(40), R2(42), KSP(46), PC(47)
9223	034300	050	051	052		. BYTE	50, 51, 52, 53, 54, 55, 56, 57		
9224	034303	053	054	055					
9225	034306	056	057						
9226	034310	060	061	062		. BYTE	60, 61, 62, 64, 65, 66		; EXCLUDE: R ZERO (63), CNSL. CNTL(67)
9227	034313	064	065	066					
9228	034316	070	071	072		. BYTE	70, 71, 72, 73, 74, 75, 76, 77		
9229	034321	073	074	075					
9230	034324	076	077						
9231	034326	377				. BYTE	-1		; END
9232		034330				. EVEN			
9233									
9234	034330	012702	001232		135:	MOV	#ADDR. 1, R2		; R2 IS LS ADDR PTR
9235									
9236	034334	012737	034744	000114		MOV	#395, @#114		; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
9237	034342	012777	001000	146464		MOV	#MAINT, @WCSST		; SET UP MAINT BIT, TIMER WILL START WHEN
9238									; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL FROM WCS
9239									
9240	034350	076700				XFCUDIS			; TRANSFER CONTROL FROM WCS
9241	034352				15:				; SHOULD RETURN FROM WCS TO THIS POINT
9242									
9243	034352	042777	001000	146454		BIC	#MAINT, @WCSST		; SHUT OFF THE TIMER
9244	034360	012737	003344	000114		MOV	#BADPAR, @#114		; RESET UNEXPECTED PARITY HANDLER
9245									
9246									
9247	034366	010016							; COPY B-SIDE GPR'S TO BOTH A&B-SIDES
9248	034370	011600				MOV	P0, (SP)		
9249	034372	010116				MOV	(SP), R0		
9250	034374	011601				MOV	R1, (SP)		
9251	034376	010216				MOV	(SP), R1		
9252	034400	011602				MOV	R2, (SP)		
9253	034402	010316				MOV	(SP), R2		
9254	034404	011603				MOV	R3, (SP)		
9255	034406	010416				MOV	(SP), R3		
9256	034410	011604				MOV	R4, (SP)		
9257	034412	010516				MOV	(SP), R4		
9258	034414	011605				MOV	R5, (SP)		
9259						MOV	(SP), R5		
9260									; CHECK RESULTS LOADED INTO BSP REGISTERS
9261	034416	012737	034544	001202		MOV	#415, \$TMP0		; PTR TO RESULT TABLE
9262	034424	012737	000040	001204		MOV	#32, \$TMP1		; THIS MANY REGISTERS
9263	034432	012737	001232	001164		MOV	#ADDR. 1, \$REG1		; WCS LOC USED
9264									
9265	034440	017737	144536	001162	205:	MOV	@\$TMP0, \$REG0		; GET MED/READ CODE
9266	034446	062737	000002	001202		ADD	#2, \$TMP0		; AUTO-INCR. THE HARD WAY
9267	034454	113737	001162	034506		MOV	\$REG0, 215		; STORE FOR MED
9268	034462	017737	144514	001166		MOV	@\$TMP0, \$REG2		; GET DATA FROM TABLE (EXPECTED)
9269	034470	062737	000002	001202		ADD	#2, \$TMP0		; AUTO-INCR. THE HARD WAY







```

9326 034744 012777 000020 146062 395: MOV #PARDIS, @WCSST ;
9327 034752 022626 CMP (SP)+, (SP)+ ;
9328 034754 005077 146054 CLR @WCSST ;
9329 034760 104137 ERROR 137 ;!!! WATCH DOG TIMEOUT !!!
9330 ;WCS WATCH DOG TIMED OUT
9331 ;UPON TRANSFERRING CONTROL FROM
9332 ;BASE MACHINE FROM WCS MICROCODE.
9333 ;NORMALLY, AFTER UCODE EXECUTION CONTROL
9334 ;SHOULD BE RETURNED TO BASE MACHINE.

```

```

9335
9336
9337 ;*****
9338 ;*TEST 105 CHECK BLOCK LOAD OF CSPAD 17:00 FROM WCS
9339

```

9340 THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:

9341 WCS. BRRAY -> CSPAD 17:40

9342 WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
9343 GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

9344 THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:

9345 CSP 17 (#1), CSP 16 (#0), CSP 15 (MD), CSP 14 (#2),  
9346 CSP 13 , . . . , CSP 04

9347 THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
9348 UNIQUE DATA PATTERNS TO IT.

9349 THE MICRO-CODE IS LOADED INTO THE WCS. USING  
9350 AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
9351 TO THE MICRO-CODE IN THE WCS.

9352 FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
9353 THE LOCAL LOAD ADDRESS (-1) STARTING FROM WHICH THE REGISTERS  
9354 SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
9355 THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
9356 WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
9357 REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
9358 CONTROL TO THE BASE MACHINE.

9359 THE REGISTERS ARE THEN CHECKED TO MAKE SURE THAT THE  
9360 DATA FROM THE WCS WAS LOADED CORRECTLY. IF NOT,  
9361 AN ERROR IS REPORTED.

9362 THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
9363 A RESULT OF SOME FAULT CONDITION, THE CONTROL  
9364 IS NOT TRANSFERRED BACK TO THE BASE MACHINE.  
9365 THE WATCH DOG WILL TIME OUT AND A TRAP TO  
9366 114 WILL OCCUR.

9367 FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

9368 TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
9369 BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
9370 CONTROL PATH USED BY MICRO-CODE EXECUTING FROM

9371  
9372  
9373  
9374  
9375  
9376  
9377  
9378  
9379  
9380  
9381



```

9382 ; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
9383 ; BUS-U-MUX-B PORT, BUS-U-LINES).
9384 ;
9385 ;*****
9386 034762 000004 TST105: SCOPE
9387 034764 012737 035440 003056 MOV #TST106,NXTST ;STARTING ADDRESS OF NEXT TEST
9388 ;
9389 ; ADDR. 1= 1503 ;LS ADDR FOR CSPAD LOAD
9390 ; 134000 ;BASE PATTERN FOR DATA
9391 ;
9392 034772 076600 000144 MED ,RDFLAG ;READ FLAGS
9393 034776 052700 004000 BIS #BIT11,R0 ;SET CSP CNST INVALID
9394 035002 076600 000344 MED ,WRFLAG ;
9395 ;
9396 035006 004537 037472 JSR R5,LDWCS ;LOAD WCS WITH MICRO CODE
9397 035012 006730 6730 ;CONTROL BLOCK LOADED HERE
9398 035014 060466 MLSCSP ;FROM THIS MEMORY LOCATION
9399 035016 006002 UDISP ;ENTRY POINT IN WCS
9400 ;
9401 ; INIT WCS LOC'S WITH DATA TO BE LOADED
9402 035020 012777 001503 146010 MOV #ADDR. 1,@WCSAR ;POINT TO AREA
9403 035026 012777 000001 146004 MOV #000001,@WCSAR ;CSP 17 = # 1
9404 035034 005277 145776 INC @WCSAR ;NEXT
9405 035040 012777 000000 145772 MOV #000000,@WCSAR ;CSP 16 = # 0
9406 035046 005277 145764 INC @WCSAR ;NEXT
9407 035052 012777 123456 145760 MOV #123456,@WCSAR ;CSP 15 = MD
9408 035060 005277 145752 INC @WCSAR ;NEXT
9409 035064 012777 000002 145746 MOV #000002,@WCSAR ;CSP 14 = # 2
9410 035072 005277 145740 INC @WCSAR ;NEXT
9411 ;
9412 035076 012704 000014 MOV #12.,R4 ;12. MORE, CSP 13:00
9413 035102 012700 134113 MOV #PAT. C+113,R0 ;DATA PATTERN ...
9414 ;
9415 035106 010077 145726 105: MOV R0,@WCSAR ;DATA PATTERN
9416 035112 005277 145720 INC @WCSAR ;NEXT LOC
9417 035116 005300 DEC R0 ;NEXT CODE
9418 035120 077406 SOB R4,105 ;LOOP
9419 ;
9420 035122 005077 145710 CLR @WCSAR ;POINT AWAY
9421 ;
9422 ; INIT CSPAD REGISTERS WITH ALL ZEROES
9423 035126 012702 035156 MOV #405,R2 ;PTR TO MED TABLE FOR THIS TEST
9424 035132 112200 115: MOVB (R2)+,R0 ;GET A MED CODE
9425 035134 100417 BMI 135 ;END OF TABLE ??
9426 035136 152700 000200 BISB #BIT7,R0 ;MAKE DATA & WRITE MED CODE
9427 035142 110037 035152 MOVB R0,125 ;LOAD WRITE-MED CODE
9428 035146 005000 CLR R0 ;WRITE ZEROES
9429 035150 076600 MED ;
9430 035152 000000 125: .WORD 0 ;
9431 035154 000766 BR 115 ;
9432 ;
9433 ; TABLE FOR INIT:
9434 035156 100 101 102 405: .BYTE 100,101,102,103,104,105,106,107
9435 035161 103 104 105
9436 035164 106 107
9437 035166 110 111 112 .BYTE 110,111,112,113 ;EXCLUDE: CSP 14:17 BASE CNST & MD

```



```

9438 035171 113
9439 035172 377
9440 035174
9441
9442 035174 012702 001503 135: MOV #ADDR. 1, R2 ; R2 IS LS ADDR PTR
9443
9444 035200 012737 035422 000114 MOV #39%, @#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
9445 035206 012777 001000 145620 MOV #MAINT, @WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
9446 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL FROM WCS
9447
9448 035214 076700 XFCUDIS ; TRANSFER CONTROL FROM WCS
9449 035216 15: ; SHOULD RETURN FROM WCS TO THIS POINT
9450
9451 035216 042777 001000 145610 BIC #MAINT, @WCSST ; SHUT OFF THE TIMER
9452 035224 012737 003344 000114 MOV #BADPAR, @#114 ; RESET UNEXPECTED PARITY HANDLER
9453
9454 ; CHECK RESULTS LOADED INTO CSP REGISTERS
9455 035232 012705 035322 MOV #41%, R5 ; PTR TO RESULT TABLE
9456 035236 012704 000020 MOV #16, R4 ; THIS MANY REGISTERS
9457 035242 012737 001503 001164 MOV #ADDR. 1, %REG1 ; WCS LOC USED
9458
9459 035250 012501 205: MOV (R5)+, R1 ; GET MED/READ CODE
9460 035252 110137 035266 MOV R1, 21% ; STORE FOR MED
9461 035256 012502 MOV (R5)+, R2 ; GET DATA FROM TABLE (EXPECTED)
9462 035260 005701 TST R1 ; IGNORE THIS ENTRY ??
9463 035262 100413 BMI 22% ; YES
9464 035264 076600 MED ; GET CSP DATA
9465 035266 000000 215: .WORD 0
9466
9467 035270 020002 CMP R0, R2 ; E-DATA = R-DATA ??
9468 035272 001407 BEQ 22% ; BR IF A-OK
9469 035274 010037 001170 MOV R0, %REG3 ; R-DATA
9470 035300 010237 001166 MOV R2, %REG2 ; E-DATA
9471 035304 010137 001162 MOV R1, %REG0 ; MED
9472 035310 104140 ERROR 140 ; DATA ERROR !!
9473
9474 035312 005237 001164 225: INC %REG1 ; BUMP WCS LOC CNTR
9475 035316 077424 SOB R4, 20% ; LOOP
9476
9477 035320 000447 BR TST106 ; EXIT
9478
9479
9480 ; ---RESULT TABLE---
9481 035322 000117 000001 415: .WORD 117, 000001 ; CSP 17 = # 1
9482 035326 000116 000000 .WORD 116, 000000 ; CSP 16 = # 0
9483 035332 177777 000000 .WORD -1, 000000 ; CSP 15 = MD **IGNORE**
9484 035336 000114 000002 .WORD 114, 000002 ; CSP 14 = # 2
9485 035342 000113 134113 .WORD 113, PAT. C+113 ; CSP 13 = PATTERN
9486 035346 000112 134112 .WORD 112, PAT. C+112 ; CSP 12 = PATTERN
9487 035352 000111 134111 .WORD 111, PAT. C+111 ; CSP 11 = PATTERN
9488 035356 000110 134110 .WORD 110, PAT. C+110 ; CSP 10 = PATTERN
9489 035362 000107 134107 .WORD 107, PAT. C+107 ; CSP 07 = PATTERN
9490 035366 000106 134106 .WORD 106, PAT. C+106 ; CSP 06 = PATTERN
9491 035372 000105 134105 .WORD 105, PAT. C+105 ; CSP 05 = PATTERN
9492 035376 000104 134104 .WORD 104, PAT. C+104 ; CSP 04 = PATTERN
9493 035402 000103 134103 .WORD 103, PAT. C+103 ; CSP 03 = PATTERN

```



9494 035406 000102 134102  
9495 035412 000101 134101  
9496 035416 000100 134100

.WORD 102, PAT. C+102 ;CSP 02 = PATTERN  
.WORD 101, PAT. C+101 ;CSP 01 = PATTERN  
.WORD 100, PAT. C+100 ;CSP 00 = PATTERN

9497  
9498  
9499

; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING  
; A BLOCK MOVE FUNCTION IN THE WCS.

9500  
9501 035422 012777 000020 145404 395:  
9502 035430 022626  
9503 035432 005077 145376  
9504 035436 104141

MOV #PARDIS, @WCSST ;  
CMP (SP)+, (SP)+ ;  
CLR @WCSST ;  
ERROR 141 ;!!! WATCH DOG TIMEOUT !!!

9505  
9506  
9507  
9508  
9509

; WCS WATCH DOG TIMED OUT  
; UPON TRANSFERRING CONTROL FROM  
; BASE MACHINE FROM WCS MICROCODE.  
; NORMALLY, AFTER UCODE EXECUTION CONTROL  
; SHOULD BE RETURNED TO BASE MACHINE.

9510  
9511  
9512  
9513

\*\*\*\*\*  
\*TEST 106 CHECK BLOCK STORE OF A&BSP TEMPS TO WCS

9514  
9515  
9516  
9517  
9518  
9519

THIS TEST CHECKS THE BLOCK STORE FUNCTION OF THE WCS:  
A&BSP TEMPS -> WCS.ARRAY

9520  
9521  
9522  
9523  
9524  
9525

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED FROM THE BASE MACHINE TO THE WCS.

9526  
9527  
9528  
9529  
9530

THE FOLLOWING REGISTERS ARE STORED IN THE GIVEN ORDER:

R SRC A, R SRC B, R DST A, R DST B,  
FDST 0, FDST 1, FDST 2, FDST 3

9531  
9532  
9533  
9534  
9535  
9536

THE REGISTERS ARE INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS IN THEM.

9537  
9538  
9539  
9540  
9541

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

9542  
9543  
9544  
9545  
9546  
9547

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL STORE ADDRESS (-1) STARTING TO WHICH THE REGISTERS  
SHOULD BE SAVED IN THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTJALLY DOES THE STORE OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN SAVED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

9548  
9549

THE WCS IS THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE REGISTERS WAS STORED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL



```

9550 ; IS NOT TRANSFERRED BACK TO THE BASE MACHINE,
9551 ; THE WATCH DOG WILL TIME OUT AND A TRAP TO
9552 ; 114 WILL OCCUR.
9553 ;
9554 ; FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:
9555 ;
9556 ; TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY
9557 ; BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION
9558 ; CONTROL PATH USED BY MICRO-CODE EXECUTING FROM
9559 ; THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,
9560 ; BUS-U-MUX-B PORT, BUS-U-LINES).
9561 ;
9562 ; *****
9563 035440 000004 TST106: SCOPE
9564 035442 012737 036004 003056 MOV #TST107,NXTST ; STARTING ADDRESS OF NEXT TEST
9565 ;
9566 001444 ADDR. 1= 1444 ; LS ADDR FOR CSPAD STORE
9567 123000 PAT. T= 123000 ; BASE PATTERN FOR DATA
9568 ;
9569 035450 004537 037472 JSR R5,LDWCS ; LOAD WCS WITH MICRO CODE
9570 035454 006760 6760 ; CONTROL BLOCK LOADED HERE
9571 035456 060626 MTMPLS ; FROM THIS MEMORY LOCATION
9572 035460 006002 UDISP ; ENTRY POINT IN WCS
9573 ;
9574 035462 012737 035766 000114 MOV #395,@#114 ; SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
9575 035470 012777 001000 145336 MOV #MAINT,@WCSST ; SET UP MAINT BIT, TIMER WILL START WHEN
9576 ; WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL TO WCS
9577 ;
9578 ; INIT TEMPORARY REGISTERS WITH DATA, DATA = '123MED'
9579 035476 012700 123277 MOV #PAT.T+BIT7+77,R0 ; DATA FOR FDST 0
9580 035502 076600 000277 MED ,277 ; WRITE IT
9581 035506 012700 123237 MOV #PAT.T+BIT7+37,R0 ; DATA FOR FDST 1
9582 035512 076600 000237 MED ,237 ; WRITE IT
9583 035516 012700 123257 MOV #PAT.T+BIT7+57,R0 ; DATA FOR FDST 2
9584 035522 076600 000257 MED ,257 ; WRITE IT
9585 035526 012700 123217 MOV #PAT.T+BIT7+17,R0 ; DATA FOR FDST 3
9586 035532 076600 000217 MED ,217 ; WRITE IT
9587 ;
9588 ; **** THE FOLLOWING CODE SEQUENCE CAN NEVER BE ALTERED ****
9589 ; GET DATA PATTERNS FOR BELOW
9590 035536 012700 123224 MOV #PAT.T+BIT7+24,R0 ; R SRC A
9591 035542 012701 123225 MOV #PAT.T+BIT7+25,R1 ; R DST A
9592 035546 012702 001444 MOV #ADDR.1,R2 ; R2 IS LS ADDR PTR
9593 035552 012703 123264 MOV #PAT.T+BIT7+64,R3 ; R SRC B
9594 035556 012704 123265 MOV #PAT.T+BIT7+65,R4 ; R DST B
9595 ; NOTE: MUST USE MACRO INSTR. THAT DON'T REFERENCE R SRC OR R DST
9596 035562 076600 000224 MED ,224 ; R SRC A
9597 035566 010100 MOV R1,R0 ;
9598 035570 076600 000225 MED ,225 ; R DST A
9599 035574 010300 MOV R3,R0 ;
9600 035576 076600 000264 MED ,264 ; R SRC B
9601 035602 010400 MOV R4,R0 ;
9602 035604 076600 000265 MED ,265 ; R DST B
9603 035610 005000 CLR R0 ;
9604 035612 005001 CLR R1 ;
9605 035614 005003 CLR R3 ;

```



```
9606 035616 005004 CLR R4 ;
9607 035620 005005 CLR R5 ;
9608 ;
9609 035622 076700 XFCUDIS ; TRANSFER CONTROL TO WCS
9610 ;**** END OF UNALTERABLE SEQUENCE **** ;
9611 035624 15: ; SHOULD RETURN FROM WCS TO THIS POINT
9612 ;
9613 035624 042777 001000 145202 BIC #MAINT,@WCSST ; SHUT OFF THE TIMER
9614 035632 012737 003344 000114 MOV #BADPAR,@#114 ; RESET UNEXPECTED PARITY HANDLER
9615 ;
9616 ; CHECK RESULTS STORED INTO WCS ARRAY
9617 035640 012705 035726 MOV #415,R5 ; PTR TO RESULT TABLE
9618 035644 012704 000010 MOV #8,R4 ; 8. REGISTERS SHOULD HAVE BEEN STORED
9619 035650 012777 001444 145160 MOV #ADDR.1,@WCSAR ; POINT TO WHERE DATA SHOULD BE
9620 ;
9621 035656 017737 145156 001170 205: MOV @WCSDR,$REG3 ; GET DATA FROM LS (RECEIVED)
9622 035664 012537 001162 MOV (R5)+,$REG0 ; GET MED CODE FROM TABLE
9623 035670 012537 001166 MOV (R5)+,$REG2 ; GET DATA FROM TABLE (EXPECTED)
9624 ;
9625 035674 023737 001166 001170 CMP $REG2,$REG3 ; E-DATA = R-DATA ??
9626 035702 001404 BEQ 215 ; BR IF A-OK
9627 035704 017737 145126 001164 MOV @WCSAR,$REG1 ; GET WCS LOC
9628 035712 104144 ERROR 144 ; DATA ERROR !!
9629 ;
9630 035714 005277 145116 215: INC @WCSAR ; ADDRESS NEXT LS LOC
9631 035720 005304 DEC R4 ; LOOP
9632 035722 003355 BGT 205 ;
9633 ;
9634 035724 000427 BR TST107 ; EXIT
9635 ;
9636 ;
9637 ; ---RESULT TABLE---
9638 035726 000024 123224 415: WORD 024, PAT. T+BIT7+24 ; R SRC A
9639 035732 000025 123225 WORD 025, PAT. T+BIT7+25 ; R DST A
9640 035736 000064 123264 WORD 064, PAT. T+BIT7+64 ; R SRC B
9641 035742 000065 123265 WORD 065, PAT. T+BIT7+65 ; R DST B
9642 035746 000077 123277 WORD 077, PAT. T+BIT7+77 ; FDST 0
9643 035752 000037 123237 WORD 037, PAT. T+BIT7+37 ; FDST 1
9644 035756 000057 123257 WORD 057, PAT. T+BIT7+57 ; FDST 2
9645 035762 000017 123217 WORD 017, PAT. T+BIT7+17 ; FDST 3
9646 ;
9647 ;
9648 ; ENTER HERE IF THE WCS WATCH DOG TIMER TIMED OUT WHILE EXECUTING
9649 ; A BLOCK MOVE FUNCTION IN THE WCS.
9650 035766 012777 000020 145040 195: MOV #PARDIS,@WCSST
9651 035774 022626 CMP (SP)+,(SP)+
9652 035776 005077 145032 CLR @WCSST
9653 036002 104145 ERROR 145 ; !!! WATCH DOG TIMEOUT !!!
9654 ; WCS WATCH DOG TIMED OUT
9655 ; UPON TRANSFERRING CONTROL FROM
9656 ; BASE MACHINE TO WCS MICROCODE.
9657 ; NORMALLY, AFTER UCODE EXECUTION CONTROL
9658 ; SHOULD BE RETURNED TO BASE MACHINE.
9659
9660
9661
```



9662  
9663  
9664  
9665  
9666  
9667  
9668  
9669  
9670  
9671  
9672  
9673  
9674  
9675  
9676  
9677  
9678  
9679  
9680  
9681  
9682  
9683  
9684  
9685  
9686  
9687  
9688  
9689  
9690  
9691  
9692  
9693  
9694  
9695  
9696  
9697  
9698  
9699  
9700  
9701  
9702  
9703  
9704  
9705  
9706  
9707  
9708  
9709  
9710  
9711  
9712  
9713  
9714  
9715  
9716  
9717

\*\*\*\*\*

\*TEST 107 CHECK BLOCK LOAD OF A&BSP TEMPS FROM WCS

THIS TEST CHECKS THE BLOCK LOAD FUNCTION OF THE WCS:

WCS. ARRAY -> A&BSP TEMPS

WHEN THIS FUNCTION IS INITIATED A BLOCK OF REGISTERS  
GET TRANSFERRED TO THE BASE MACHINE FROM THE WCS.

THE FOLLOWING REGISTERS ARE LOADED IN THE GIVEN ORDER:

R SRC A, R SRC B, R DST A, R DST B,  
FDST 0 , FDST 1 , FDST 2 , FDST 3

THE WCS ARRAY IS INITIALIZED TO A KNOWN STATE, BY WRITING  
UNIQUE DATA PATTERNS TO IT.

THE MICRO-CODE IS LOADED INTO THE WCS. USING  
AN "XFC" DISPATCH INSTRUCTION CONTROL IS TRANSFERRED  
TO THE MICRO-CODE IN THE WCS.

FROM GPR "R2", THE MICRO-CODE SETS UP IN "D" (BASE MACHINE),  
THE LOCAL LOAD ADDRESS (-1) STARTING FROM WHICH THE REGISTERS  
SHOULD BE RESTORED FROM THE WCS. THEN THIS MICRO-CODE SELECTS  
THE WCS HOT BOX AND TRANSFERS CONTROL TO THE TMS-ROM  
WHICH ACTUALLY DOES THE LOAD OF THE REGISTERS. AFTER THE  
REGISTERS HAVE BEEN RESTORED, THE MICRO-CODE RETURNS  
CONTROL TO THE BASE MACHINE.

THE REGISTERS ARE THEN CHECKED TO MAKE SURE THAT THE  
DATA FROM THE WCS WAS LOADED CORRECTLY. IF NOT,  
AN ERROR IS REPORTED.

THIS TEST USES THE WCS WATCH-DOG-TIMER. IF AS  
A RESULT OF SOME FAULT CONDITION, THE CONTROL  
IS NOT TRANSFERRED BACK TO THE BASE MACHINE,  
THE WATCH DOG WILL TIME OUT AND A TRAP TO  
114 WILL OCCUR.

FAILURE OF THIS TEST COULD INDICATE A FAULT IN THE:

TMS-CROM SECTION OF WCS INCLUDING ENTRY-POINT ARRAY  
BUS-U-MUX-PORT-A, SUPPLYING TMS CONTROL INFORMATION  
CONTROL PATH USED BY MICRO-CODE EXECUTING FROM  
THE WCS- (NUA<9:0>, ARRAY ADDRESS MUX-A-PORT,  
BUS-U-MUX-B PORT, BUS-U-LINES).

\*\*\*\*\*

TST107. SCOPE  
MOV #TST110.NXTST ;STARTING ADDRESS OF NEXT TEST

036004 000004  
036006 012737 036404 003056

```

9718          001770          ADDR. 1= 1770          ;LS ADDR FOR CSPAD LOAD
9719          076000          PAT. T= 076000          ;BASE PATTERN FOR DATA
9720
9721 036014 004537 037472    JSR      R5,LDWCS          ;LOAD WCS WITH MICRO CODE
9722 036020 006750          6750          ;CONTROL BLOCK LOADED HERE
9723 036022 060566          MLSTMP          ;FROM THIS MEMORY LOCATION
9724 036024 006002          UDISP          ;ENTRY POINT IN WCS
9725
9726          ;INIT WCS LOC'S WITH DATA TO BE LOADED
9727 036026 012777 001770 145002  MOV      #ADDR. 1, @WCSAR ;POINT TO AREA
9728 036034 012705 036132    MOV      #40$, R5          ;TABLE PTR TO MED CODES
9729
9730 036040 112500          10$: MOVB   (R5)+, R0          ;GET A MED CODE
9731 036042 100407          BMI     29$              ;DONE IF -1
9732 036044 052700 076000    BIS     #PAT. T+000, R0   ;FORM DATA PATTERN = 076MED
9733 036050 010077 144764    MOV     R0, @WCSDR        ;DATA PATTERN
9734 036054 005277 144756    INC     @WCSAR           ;NEXT LOC
9735 036060 000767          BR      10$              ;LOOP
9736
9737 036062 005077 144750    29$: CLR     @WCSAR        ;POINT AWAY
9738
9739 036066 012737 036366 000114  MOV     #39$, @#114      ;SETUP SERVICE VECTOR FOR WCS WATCH DOG TIMER
9740 036074 012777 001000 144732  MOV     #MAINT, @WCSST   ;SET UP MAINT BIT, TIMER WILL START WHEN
9741          ;WCS HOT BOX IS SELECTED, UPON TRANSFERRING CONTROL FROM WCS
9742
9743          ;INIT TEMPORARY REGISTERS WITH ALL ZEROES
9744 036102 012702 036132    MOV     #40$, R2          ;PTR TO MED TABLE FOR THIS TEST
9745 036106 112200          11$: MOVB   (R2)+, R0          ;GET A MED CODE
9746 036110 100415          BMI     13$              ;END OF TABLE ??
9747 036112 152700 000200    BISB   #BIT7, R0         ;MAKE DATA & WRITE MED CODE
9748 036116 110037 036126    MOVB   R0, 12$          ;LOAD WRITE-MED CODE
9749 036122 005000          CLR     R0               ;WRITE ZEROES
9750 036124 076600          MED
9751 036126 000000          12$: .WORD 0
9752 036130 000766          BR      11$
9753
9754          ;TABLE FOR INIT:
9755 036132          024 025 064 40$: .BYTE 024, 025, 064, 065 ;R SRC A, R DST A, R SRC B, R DST B
9756 036135          065
9757 036136          077 037 057 .BYTE 077, 037, 057, 017 ;FDST 0, 1, 2, 3
9758 036141          017
9759 036142          377 .BYTE -1 ;END
9760          036144 .EVEN
9761
9762 036144 012702 001770    13$: MOV     #ADDR. 1, R2    ;R2 IS LS ADDR PTR
9763 036150 005000          CLR     R0
9764 036152 005001          CLR     R1
9765 036154 005003          CLR     R3
9766 036156 005004          CLR     R4
9767
9768          ;**** THE FOLLOWING CODE SEQUENCE CAN NEVER BE ALTERED ****
9769 036160 076700          XFCUDIS ;TRANSFER CONTROL FROM WCS
9770 036162          1$: ;SHOULD RETURN FROM WCS TO THIS POINT
9771 036162 076600 000024    MED     , 024 ;GET R SRC A
9772 036166 010001          MOV     R0, R1
9773 036170 076600 000025    MED     , 025 ;GET R DST A

```





9830  
9831  
9832  
9833  
9834  
9835  
9836  
9837  
9838  
9839  
9840  
9841  
9842  
9843  
9844  
9845  
9846  
9847  
9848

036404 000004  
036406 005037 001222  
036412 105037 001103  
036416 012700 015507  
036422 076600 000352

```
;;*****  
;*TEST 110 ... END...  
;;*****  
TST110: SCOPE  
CLR STIMES ;NO ITER  
CLRB SERFLG ;OR ERRORS  
MOV #015507,R0 ;CNST FOR 'INIT' ROUTINE  
MED ,352 ;GO INIT THE MACHINE ...  
; END OF TMS TESTS
```



```
9849      .SBTTL  END OF PASS ROUTINE
9850
9851      ;*****
9852      ;*INCREMENT THE PASS NUMBER ($PASS)
9853      ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
9854      ;*IF THERES A MONITOR GO TO IT
9855      ;*IF THERE ISN'T JUMP TO NEWPAS
9856
9857      SEOP:
9858      036426 000004          SCOPE
9859      036430 005037 001102  CLR      $TSTNM      ;; ZERO THE TEST NUMBER
9860      036434 005037 001222  CLR      $TIMES      ;; ZERO THE NUMBER OF ITERATIONS
9861      036440 005237 001244  INC      $PASS       ;; INCREMENT THE PASS NUMBER
9862      036444 042737 100000 001244  BIC      #100000,$PASS ;; DON'T ALLOW A NEG. NUMBER
9863      036452 005327          DEC      (PC)+      ;; LOOP?
9864      036454 000001          SEOPCT: .WORD 1
9865      036456 003022          BGT      $DOAGN      ;; YES
9866      036460 012737          MOV      (PC)+,@(PC)+ ;; RESTORE COUNTER
9867      036462 000001          SENDCT: .WORD 1
9868      036464 036454          SEOPCT
9869      036466 104401 036533  TYPE      , $SENDMG  ;; TYPE "END PASS #"
9870      036472 013746 001244  MOV      $PASS,-(SP) ;; SAVE $PASS FOR TYPEOUT
9871      036476 104405          TYPDS
9872      036500 104401 036530  TYPE      , $ENULL   ;; TYPE A NULL CHARACTER
9873      036504 013700 000042  SGET42: MOV      @#42,R0 ;; GET MONITOR ADDRESS
9874      036510 001405          BEQ      $DOAGN      ;; BRANCH IF NO MONITOR
9875      036512 000005          RESET
9876      036514 004710          SENDAD: JSR     PC,(R0) ;; GO TO MONITOR
9877      036516 000240          NOP
9878      036520 000240          NOP
9879      036522 000240          NOP
9880      036524          $DOAGN:
9881      036524 000137          JMP      @(PC)+      ;; RETURN
9882      036526 004054          SRTNAD: .WORD  NEWPAS
9883      036530      377      377      000  $ENULL: .BYTE  -1,-1,0 ;; NULL CHARACTER STRING
9884      036533      015      042412 042116  $ENDMG: .ASCIIZ <15><12>/END PASS #/
9885      036540 050040 051501 020123
9886      036546 000043
```



9887  
9888  
9889  
9890  
9891  
9892  
9893  
9894  
9895  
9896  
9897  
9898  
9899  
9900  
9901  
9902  
9903  
9904  
9905  
9906  
9907  
9908  
9909  
9910  
9911  
9912  
9913  
9914  
9915  
9916  
9917  
9918  
9919  
9920  
9921  
9922  
9923  
9924  
9925  
9926  
9927  
9928  
9929  
9930  
9931  
9932  
9933  
9934  
9935  
9936  
9937  
9938  
9939  
9940  
9941  
9942

036550 012737 003276 000004  
036556 012737 003344 000114  
036564 012737 000012 000010  
036572 000207  
  
036574 032777 020000 142336  
036602 001005  
036604 017637 000000 036614  
036612 104401  
036614 000000  
036616 062716 000002  
036622 000002  
  
036624 010046  
036626 012777 000020 144200  
036634 005000  
036636 010077 144174  
036642 005077 144172  
036646 005200  
036650 020027 006000  
036654 001370  
036656 005077 144152  
036662 012600  
036664 000207  
  
036666 010346  
036670 012703 000001  
036674 000402

; VECTRS

VECTRS: MOV #BADTMO, @#4  
MOV #BADPAR, @#114  
MOV #12, @#10  
RTS PC

; TYPMSG

; THIS ROUTINE IS USED FOR TYPING OUT ASCII MESSAGES. BEFORE THE  
; MESSAGE IS TYPED OUT SW13 IS CHECKED & IF SET THE TYPEOUT IS  
; INHIBITED, AND AN EXIT IS MADE.

; CALL: TYPMSG  
; MSGPOINTER

TY. MSG:

BIT #SW13, @SWR ; INHIBIT TYPEOUT?  
BNE 25 ; YES, EXIT  
MOV @ (SP), 15 ; GET POINTER TO ASCII MESSAGE  
TYPE ; TYPE THE ASCII STRING  
0 ;  
25: ADD #2, (SP) ; ADJUST RETURN ADDRESS, SKIP  
RTI ; OVER POINTER ON RETURN

; CLRWCS

CLRWCS: MOV R0, -(SP)  
MOV #PARDIS, @WCSST  
CLP R0  
15: MOV R0, @WCSAP  
CLR @WCSDR  
INC R0  
CMP R0, #6000  
BNE 15  
CLR @WCSST  
MOV (SP)+, R0  
RTS PC

; GENPAR

; THIS ROUTINE GENERATES EVEN PARITY FOR GIVEN 16 BIT PATTERN.

; GENWPAR

; THIS ROUTINE GENERATES ODD PARITY FOR GIVEN 16 BIT PATTERN.

; ON ENTRY, R4 CONTAINS THE PATTERN FOR WHICH THE PARITY IS TO BE  
; GENERATED. ON EXIT, R2 CONTAINS THE PARITY BIT GENERATED, IT IS  
; PLACED IN BIT POSITION 7 (SAME AS 'PARGEN' OF WCSST).

; CALL: JSR PC, PARGEN ; R4 CONTAINS THE PATTERN  
; R2 CONTAINS PARITY BIT. ON RETURN

; THIS ROUTINE USES R2.

GENWPAR:

MOV R3, -(SP)  
MOV #1, R3  
BR GENP1



```

9943
9944 036676 010346      GENPAR: MOV    R3, -(SP)      ; SAVE R3, R4
9945 036700 005003      CLR    R3
9946 036702      GENP1:
9947 036702 010446      MOV    R4, -(SP)
9948 036704 012702 000020  MOV    #16, R2
9949 036710 006104      1$:   ROL    R4
9950 036712 005503      ADC    R3      ; KEEP COUNT OF 1'S
9951 036714 077203      SOB    R2, 1$  ; DONE?
9952 036716 006003      ROR    R3      ; ODD OR EVEN?
9953 036720 103002      BCC    3$,    ; EVEN, PARITY BIT=0
9954 036722 012702 000200  MOV    #200, R2 ; ODD, PARITY BIT=1
9955 036726 012604      3$:   MOV    (SP)+, R4
9956 036730 012603      MOV    (SP)+, R3
9957 036732 000207      RTS    PC      ; EXIT
9958
9959
9960      ; XFRPGN
9961      ; THIS ROUTINE TRANSFORMS THE PATTERN (CONTAINED IN R4) TO A
9962      ; PATTERN COMPATIBLE WITH THE PARITY GENERATOR CHIPS.
9963      ; THE INPUTS TO THE TWO PARITY GENERATOR CHIPS ARE:
9964      ;      <----- HI BYTE----->      <--- LO BYTE--->
9965      ; DB      15 14 13 12 11 10 1 0      9 8 7 6 5 4 3 2
9966
9967      ; THE PATTERN GENERATED AT THE MACRO LEVEL HAS A DIFFERENT BIT CONFIGURATION:
9968      ;
9969      ;      15 14 13 12 11 10 9 8      7 6 5 4 3 2 1 0
9970
9971      ; THUS THE NORMAL MACRO LEVEL COUNT PATTERN (USED FOR EACH OF THE TWO PARITY
9972      ; CHIPS) HAS TO BE TRANSFORMED, SO THAT AT MACHINE LEVEL THE
9973      ; PARITY CHIPS SEE THE RIGHT COUNT PATTERN.
9974
9975      ; CALL:  MOV R0, R4      ; R4 CONTAINS PATTERN TO BE TRANSFORMED
9976      ;      JSR PC, XFRPGN  ; ON EXIT, R4 CONTAINS THE TRANSFORMED PATTERN
9977
9978 036734 010546      XFRPGN: MOV    R5, -(SP)      ; SAVE R5, R3
9979 036736 010346      MOV    R3, -(SP)
9980 036740 010405      MOV    R4, R5      ; COPY R4 INTO R5 AND
9981 036742 000305      SWAB   R5      ; MAP BITS 9, 8 TO 1, 0
9982 036744 042705 177774  BIC    #177774, R5
9983 036750 010403      MOV    R4, R3      ; COPY R4 INTO R3 AND
9984 036752 006303      ASL    R3      ; MAP BITS 7, 6, 5, 4, 3, 2, 1, 0 TO
9985 036754 006303      ASL    R3      ; 9, 8, 7, 6, 5, 4, 3, 2 RESPECTIVELY
9986 036756 042703 176000  BIC    #176000, R3
9987 036762 042704 001777  BIC    #1777, R4      ; BITS 15-10 MAP INTO SAME 15-10
9988 036766 050504      BIS    R5, R4      ; CONSTRUCT THE TRANSFORMED PATTERN
9989 036770 050304      BIS    R3, R4
9990 036772 012603      MOV    (SP)+, R3      ; RESTORE R3, R5
9991 036774 012605      MOV    (SP)+, R5
9992 036776 000207      RTS    PC      ; EXIT
9993      ; TRPMSG
9994      ; ROUTINE TO TYPE OUT THE PC AT WHICH PARITY ERROR TRAP TOOK PLACE.
9995      ; ON ENTRY, 'TRAPC' CONTAINS THE PC ON THE STACK AFTER THE TRAP
9996      ; TOOK PLACE.
9997
9998 037000 032777 020000 142132 TRPMSG: BIT    #SW13, @SWR      ; INHIBIT TYPING?

```



```

9999 037006 001011          BNE      1$          ;YES, SKIP
10000
10001 037010 104401 042520  TYPE,    MSG12      ;REPORT UNEXPECTED PARITY ERROR
10002 037014 104401 043233  TYPE,    MSG31      ;TRAP
10003 037020 013746 037034  MOV      TRAPC, -(SP)
10004 037024 162716 000002  SUB      #2, (SP)
10005 037030 104402          TYP0C          ;PC AT WHICH TRAP OCCURRED
10006 037032 000207          1$: RTS      PC
10007
10008 037034 000000          TRAPC: .WORD    0          ;CONTAINS PC AT WHICH TRAP OCCURRED
10009
10010          ;BITERR
10011          ;THIS ROUTINE IS ENTERED UPON AN ERROR IN THE CONTENTS OF STATUS REGISTER.
10012          ;IT PRINTS OUT INFORMATION ABOUT THE BIT/S THAT WAS/WERE
10013          ;EITHER NOT SET OR NOT CLEAR (AS THE CASE MAY BE). ON ENTRY
10014          ;'CRSTAT' CONTAINS THE STATUS REGISTER CONTENTS (DONT CARE BITS (CMASK)
10015          ;HAVE BEEN MASKED). 'CESTAT' CONTAINS THE EXPECTED CONTENTS OF THE
10016          ;STATUS REGISTER.
10017
10018 037036 010046          BITERR: MOV      R0, -(SP)      ;SAVE R0,R1,R2,R3 ON STACK
10019 037040 010146          MOV      R1, -(SP)
10020 037042 010246          MOV      R2, -(SP)
10021 037044 010346          MOV      R3, -(SP)
10022
10023 037046 013701 037254          MOV      CRSTAT, R1      ;GET RECEIVED STATUS (BAD)
10024 037052 013702 037252          MOV      CESTAT, R2      ;GET EXPECTED STATUS (GOOD)
10025
10026 037056 074201          XOR      R2, R1          ;FIND WHICH BITS ARE IN ERROR?
10027 037060 012703 177777          MOV      #-1, R3
10028 037064 005000          CLR      R0
10029 037066 000261          SEC
10030 037070 006100          1$: ROL      R0          ;FIND OUT THE BIT POSITION IN ERROR
10031 037072 001422          BEQ      5$,           ;EXIT IF DONE
10032 037074 005203          INC      R3
10033 037076 030001          BIT      R0, R1          ;THIS BIT?
10034 037100 001773          BEQ      1$,           ;NO
10035 037102 006303          ASL      R3          ;YES, FORM POINTER TO MESSAGE
10036 037104 016337 003066 037116          MOV      PMSG2(R3), 3$  ;TYPE OUT THE BIT DESCRIPTION
10037 037112 006203          ASR      R3
10038 037114 104406          TYPMSG
10039 037116 000000          3$: .WORD    0
10040 037120 030002          BIT      R0, R2          ;WAS THAT BIT DROPPED OR PICKED?
10041 037122 001003          BNE      2$,           ;DROPPED
10042
10043 037124 104406 042470          TYPMSG  ,MSG10          ;BIT WAS PICKED UP. HENCE,
10044 037130 000757          BR       1$          ;TYPE OUY 'NOT CLEAR'
10045
10046 037132 104406 042505          2$: TYPMSG  ,MSG11          ;THE BIT WAS DROPPED. HENCE,
10047
10048 037136 000754          BR       1$          ;TYPE OUT 'NOT SET'
10049
10050 037140 012603          5$: MOV      (SP)+, R3      ;CHECK REST OF THE BITS
10051 037142 012602          MOV      (SP)+, R2      ;RESTORE R0,R1,R2,R3
10052 037144 012601          MOV      (SP)+, R1
10053 037146 012600          MOV      (SP)+, R0
10054 037150 000207          RTS      PC          ;RETURN

```



```

10055
10056 ;CHKSTAT
10057 ;THIS ROUTINE CHECKS THAT THE CONTENTS OF THE WCS STATUS REGISTER
10058 ;ARE CORRECT (AS EXPECTED). UPON ENTRY, 'CESTAT' CONTAINS THE
10059 ;EXPECTED STATUS REGISTER CONTENTS; 'CMASK' CONTAINS THE BITS THAT
10060 ;HAVE TO BE MASKED OUT (NOT TO BE COMPARED).
10061 ;ON COMPARISON OF THE STATUS REGISTER, IF AN ERROR IS FOUND THE
10062 ;ROUTINE 'BITERR' IS ENTERED. THIS ROUTINE TYPES OUT WHICH BIT
10063 ;OF THE STATUS REGISTER WAS IN ERROR AND WHETHER IT WAS NOT SET
10064 ;OR NOT CLEAR, AS THE CASE MAY BE.
10065 ;UPON RETURN, '$REGO' CONTAINS THE EXPECTED STATUS REGISTER CONTENTS
10066 ;AND '$REG1' CONTAINS THE RECEIVED CONTENTS.
10067
10068 037152 010146      CHKSTAT: MOV    R1, -(SP)      ;SAVE R1,R5 ON STACK
10069 037154 010546      MOV    R5, -(SP)
10070
10071 037156 017701 143652  MOV    @WCSST,R1      ;READ STATUS REGISTER
10072 037162 010137 001164  MOV    R1,$REG1      ;SAVE A COPY FOR ERROR REPORTING
10073 037166 043701 037256  BIC    CMASK,R1      ;MASK OUT 'DONT CARE' BITS
10074 037172 010137 037254  MOV    R1,CRSTAT     ;SAVE THE RESULT (RECORD STATUS)
10075 037176 023701 037252  CMP    CESTAT,R1     ;CONTENTS OF STATUS REGISTER OK?
10076 037202 001415      BEQ    1$           ;YES
10077                               ;NO, TYPE OUT WHICH BIT WAS
10078 037204 004737 037036  JSR    PC,BITERR    ;NOT SET OR NOT CLEAR?
10079
10080 037210 013701 037256  MOV    CMASK,R1      ;GET THE MASK BITS
10081 037214 005101      COM    R1
10082 037216 013705 001164  MOV    $REG1,R5      ;FORM THE EXPECTED STATUS REGISTER
10083 037222 040105      BIC    R1,R5         ;CONTENTS, CORRESPONDING TO THE
10084 037224 053705 037252  BIS    CESTAT,R5     ;MASK BITS
10085 037230 010537 001162  MOV    R5,$REGO      ;SAVE FOR ERROR REPORTING
10086 037234 000403      BR     2$
10087 037236 013766 037260 000004 1$: MOV    (CRETRN,4(SP)) ;RETURN TO SKIP OVER ERROR MESSAGE
10088 037244 012605      2$: MOV    (SP)+,R5   ;RESTORE R1,R5
10089 037246 012601      MOV    (SP)+,R1
10090 037250 000207      RTS    PC
10091
10092 037252 000000      CESTAT: .WORD 0
10093 037254 000000      CRSTAT: .WORD 0
10094 037256 000000      CMASK:  .WORD 0
10095 037260 000000      CRETRN: .WORD 0
10096 ;XFRCHK
10097 ;THIS ROUTINE REMAPS THE MACRO-LEVEL PATTERN USED FOR THE
10098 ;PARITY CHECKERS CORRESPONDING TO PAR1, WP1 (DATA BITS D<16:31>
10099 ;OF THE DATA ARRAY D<47:0>). THE INPUTS TO THE ABOVE TWO
10100 ;PARITY CHECKERS ARE AS FOLLOWING:
10101 ;
10102 ;
10103 ;          HI BYTE                LO BYTE
10104 ;DOUT    31 30 29 28 27 26 19 16    25 24 23 22 21 20 18 17
10105 ;WHEREAS THE MACRO-LEVEL BIT CONFIGURATION IS:
10106 ;
10107 ;          HI BYTE                LO BYTE
10108 ;          31 30 29 28 27 26 25 24    23 22 21 20 19 18 17 16
10109 ;THE MAPPING IS DONE SO THAT THE SAME (MACRO-LEVEL)
10110 ;PATTERN APPEARS AT THE INPUTS OF THE TWO PARITY CHIPS.

```



```

10111          ;CALL:  MOV      PATRN,R4      ;PATTERN TO BE MAPPED
10112          ;      JSR      PC,XFRCHK     ;ON EXIT THE MAPPED PATTERN
10113          ;      ;                      ;IS IN R4
10114
10115 037262 010346      XFRCHK: MOV      R3,-(SP)      ;SAVE R3,R5
10116 037264 010546      MOV      R5,-(SP)
10117 037266 010403      MOV      R4,R3
10118 037270 006303      ASL      R3          ;MAP BITS 17, 18
10119 037272 010305      MOV      R3,R5
10120 037274 042703 177771 BIC      #177771,R3
10121 037300 006305      ASL      R5
10122 037302 042705 176017 BIC      #176017,R5      ;MAP BITS <25-20>
10123 037306 050503      BIS      R5,R3
10124 037310 032704 000400 BIT      #BIT8,R4      ;MAP BIT 16
10125 037314 001402      BEQ      15
10126 037316 052703 000001 BIS      #BIT0,R3
10127 037322 032704 001000 15: BIT      #BIT9,R4      ;MAP BIT 19
10128 037326 001402      BEQ      25
10129 037330 052703 000010 BIS      #BIT3,R3
10130 037334 042704 001777 25: BIC      #1777,R4      ;MAP BITS <31:26>
10131 037340 050304      BIS      R3,R4
10132 037342 012605      MOV      (SP)+,R5      ;RESTORE R5,R3
10133 037344 012603      MOV      (SP)+,R3
10134 037346 000207      RTS      PC
10135          ;FAULT ANALYZER
10136          ;THIS FAULT ANALYZER ATTEMPTS TO ANALYZE THE ERRORS THAT
10137          ;HAVE OCCURED IN THE ADDRESSING TESTS AND INDICATE WHICH ADDRESS
10138          ;LINE/S WERE FAULTY. NOTE THAT THIS ANALYZER IS BASED ON
10139          ;SINGLE FAULT CONCEPT AND GIVES A HIGH PROBABILITY
10140          ;ANALYSIS (NOT A 100% TRUE ANALYSIS).
10141          ;'TMP0' CONTAINS LOGICAL 'OR' OF ALL THE DATA PATTERNS THAT FAILED.
10142          ;'TMP1' CONTAINS LOGICAL 'AND' OF ALL THE DATA PATTERNS THAT FAILED.
10143          ;LOGICAL 'XOR' OF 'TMP0' AND 'TMP1' IS DONE. THE RESULT
10144          ;IS THEN COMPLEMENTED.
10145          ;THE FINAL RESULT WILL BE A WORD
10146          ;IN WHICH ONE OR MORE BITS MAY BE SET. ADDRESS LINES
10147          ;CORRESPONDING TO THESE BIT POSITIONS ARE PROBABLY FAULTY.
10148
10149          ;THIS ROUTINE ALSO TYPES OUT TWO OCTAL NUMBERS, CONTENTS
10150          ;OF 'TMP0' (OR), 'TMP1' (AND). THIS MAY BE INTERPRETED
10151          ;AS FOLLOWING.
10152          ;CASE 1
10153          ;OR = ALL 1'S, AND IS NOT = 0
10154          ;ADDRESS LINES CORRESPONDING TO THE BIT POSITIONS THAT
10155          ;ARE SET IN 'AND' ARE EITHER OPEN OR STUCK HIGH.
10156          ;CASE 2
10157          ;OR IS NOT = 0, AND = 0
10158          ;ADDRESS LINES CORRESPONDING TO THE BIT POSITIONS THAT ARE
10159          ;CLEAR IN 'OR' ARE STUCK TO 0
10160          ;CASE 3
10161          ;'AND' IS GREATER THAN 0, 'OR' IS LESS THAN 177777
10162          ;NON-PREDICTABLE CASE. THERE COULD BE MULTIPLE FAULTS, NON-
10163          ;CONSISTENT SOFT FAILURES, MIXED DATA AND ADDRESS
10164          ;FAILURES.
10165          ;AT THE POINT OF ENTRY, 'TMP0' CONTAINS THE LOGICAL 'OR'.
10166          ;'TMP1' CONTAINS THE LOGICAL 'AND'. R0, R1 CONTAIN THE

```



```

10167 ; LOGICAL 'OR' AND 'AND' SHIFTED TO COVER THE ADDRESS BITS
10168 ; THAT ARE BEING CHECKED.
10169 ; CALL: MOV #MSG, -(SP) ; MESSAGE POINTER
10170 ; MOV #BITMASK, -(SP) ; MASK TO BE USED
10171 ; JSR PC, ANLYSO
10172
10173 037350 010004 ANLYSO: MOV R0, R4 ; XOR (OR, AND) ---> R4
10174 037352 074104 XOR R1, R4
10175 037354 005104 COM R4 ; COM (XOR) ---> R4
10176 037356 046604 000002 BIC 2(SP), R4 ; USE THE BIT MASK TO CLEAR OUT
10177 ; UNUSED BITS
10178 037362 010405 MOV R4, R5 ; BITS (POSITIONS) SET IN R4
10179 ; GIVE ADDRESS LINES THAT ARE
10180 ; FAULTY
10181 037364 016637 000004 037400 MOV 4(SP), 15 ; GET MESSAGE POINTER
10182 037372 104406 042701 TYPMSG, MSG21 ; TYPE - FAULTY ADDRESS LINES
10183 037376 104406 TYPMSG
10184 037400 000000 15: .WORD 0
10185 ; TYPE OUT THE ADDRESS LINES
10186 037402 012702 177777 MOV #-1, R2 ; CORRESPONDING TO BITS THAT
10187 037406 005202 25: INC R2 ; ARE SET
10188 037410 006005 ROR R5
10189 037412 103002 BCC 35
10190 037414 010246 MOV R2, -(SP)
10191 037416 104405 TYPDS
10192 037420 020227 000011 35: CMP R2, #9
10193 037424 101770 BLOS 25
10194
10195 037426 104406 042757 TYPMSG, MSG23 ; AND =
10196 037432 010146 MOV R1, -(SP) ; TYPE OUT CONTENTS OF TMP1
10197 037434 046616 000004 BIC 4(SP), (SP) ; FAKE OUT BITS CORRESPONDING TO THE MASK
10198 037440 104403 TYPOS
10199 037442 004 .BYTE 4
10200 037443 001 .BYTE 1
10201
10202 037444 104406 042766 TYPMSG, MSG24 ; OR =
10203 037450 010046 MOV R0, -(SP) ; TYPE OUT CONTENTS OF TMP0
10204 037452 056616 000004 BIS 4(SP), (SP) ; FAKE OUT BITS CORRESPONDING TO THE MASK
10205 037456 104403 TYPOS
10206 037460 004 .BYTE 4
10207 037461 001 .BYTE 1
10208
10209 037462 011666 000004 MOV (SP), 4(SP) ; ADJUST RETURN ADDRESS
10210 037466 022626 CMP (SP)+, (SP)+
10211
10212 037470 000207 PTS PC ; RETURN
10213
10214
10215
10216
10217
10218 ; WCS LOADER, LDWCS
10219
10220 ; CALL: JSR P5, LDWCS
10221 ; BGNCB ; STARTING ADDRESS OF CONTROL BLOCK
10222 ; IN WCS

```



```

10223 ; MEMLOC ; STARTING ADDRESS IN MEMORY FROM
10224 ; ; WHICH THE WCS IS TO BE LOADED
10225 ; ENTRY ; ENTRY POINT INTO THE WCS
10226
10227 ; FORMAT OF THE MICRO-CODE (CONTROL) BLOCK TO BE LOADED
10228 ; INTO THE WCS
10229
10230 ; ENTRYWRDD ; THIS IS THE MICRO-WORD TO BE LOADED
10231 ; ENTRYWRD1 ; INTO THE WCS AT THE ENTRY POINT
10232 ; ENTRWRD2 ; 'ENTRY' - 'WRDD' (16 BITS) GOES INTO
10233 ; ; SEGMENT 0, 'WRD1' IN SEGMENT 1,
10234 ; ; 'WRD2' IN SEGMENT 2 OF WCS.
10235 ;
10236 ; CBWRDD ;
10237 ; CBWRD1 ; 3 16-BIT WORDS MAKE UP THE MICRO-WORDS
10238 ; CBWRD2 ; WHICH ARE TO BE LOADED INTO WCS STARTING
10239 ; ; AT 'BGNCB'
10240 ; UWDEND ; END OF BLOCK, TERMINATOR
10241
10242 ; NOTE THE WCS OPERATES IN PAGES 6-7. THUS THE ADDRESSES
10243 ; ARE FROM 6000 - 7777. WCS ARRAY IS MADE UP OF 3 16-BIT
10244 ; SEGMENTS. THE FIRST SEGMENT CONTAINS BITS (15:0), SECOND ONE IS
10245 ; BITS (31:16), THIRD IS BITS (47:32)
10246
10247 ; THE ROUTINE INITIALLY LOADS UP THE ENTIRE WCS WITH NULL (BRANCH ON
10248 ; SELF) MICROWORDS. THIS IS DONE TO PROTECT THE BASE MACHINE, IN CASE OF
10249 ; A FAULT INSIDE THE WCS (IF THE CONTROL PASSES TO ANY MICRO-WORD INSIDE
10250 ; THE WCS, TO WHICH IT SHOULD NOT HAVE PASSED, THE "BRANCH ON SELF" MICRO
10251 ; WORD WILL BE EXECUTED AND THE MACHINE WILL LOOP ON IT.)
10252
10253 ; AFTER LOADING EACH MICROWORD INSIDE THE WCS THE LOADER CHECKS IF THE WORD
10254 ; WAS LOADED CORRECTLY. IF IT WAS INCORRECT AN ERROR IS REPORTED AND THE
10255 ; TEST IS ABORTED.
10256
10257 037472 010046 LDWCS: MOV R0, -(SP) ; SAVE R0, R1, R2, R3
10258 037474 010146 MOV R1, -(SP)
10259 037476 010246 MOV R2, -(SP)
10260 037500 010346 MOV R3, -(SP)
10261
10262 ; INITIALIZE THE WCS WITH NULL (BRANCH ON SELF) MICRO-WORDS
10263 037502 005001 CLR R1
10264 037504 010100 45: MOV R1, R0
10265 037506 010177 143324 MOV R1, @WCSAR
10266 037512 005100 COM R0 ; NUA<8:0> BITS ARE INVERTED
10267 037514 042700 147000 BIC #147000, R0 ; BIT12,13 ARE SET, NULL BUT
10268 037520 010077 143314 MOV R0, @WCSDR ; LOAD SEGMENT 0
10269 037524 005201 INC R1
10270 037526 020127 002000 CMP R1, #1024. ; DONE?
10271 037532 001364 BNE 45.
10272 037534 010177 143276 55: MOV R1, @WCSAR ; CLEAR OUT SEGMENT 1-2
10273 037540 005077 143274 CLR @WCSDR
10274 037544 005201 INC R1
10275 037546 020127 006000 CMP R1, #3072. ; DONE?
10276 037552 001370 BNE 55
10277
10278 ; NOW, LOAD THE MICRO-TEST

```



```

10279 037554 012501          MOV      (R5)+,R1      ;WCS ADDRESS WHERE MICRO-CODE
10280                                ;IS TO BE LOADED
10281 037556 012502          MOV      (R5)+,R2      ;MEMORY ADDRESS FROM WHICH
10282                                ;MICRO-CODE IS TO BE LOADED
10283 037560 012577 143252    MOV      (R5)+,@WCSAR  ;ENTRY POINT INTO WCS
10284 037564 012700 177777    MOV      #-1,R0
10285 037570 042777 176000 143240 25:  BIC      #176000,@WCSAR ;CONVERT MICRO-ADDRESS TO
10286                                ;ARRAY ADDRESS
10287
10288 037576 012737 000777 001164 15:  MOV      #777,$REG1    ;LOAD FIRST SEGMENT OF UWORD
10289 037604 012203          MOV      (R2)+,R3
10290 037606 074337 001164    XOR      R3,$REG1      ;COMPLEMENT BITS<8:0>, THAT IS
10291 037612 013777 001164 143220  MOV      $REG1,@WCSDR  ;NUA<8:0>
10292 037620 027737 143214 001164    CMP      @WCSDR,$REG1  ;LOADED CORRECTLY?
10293 037626 001040          BNE      6$           ;NO, ERROR
10294 037630 062777 002000 143200    ADD      #2000,@WCSAR  ;ADDRESS THE MIDDLE SEGMENT
10295 037636 011277 143176    MOV      (R2),@WCSDR  ;LOAD SECOND SEGMENT OF UWORD
10296 037642 022277 143172    CMP      (R2)+,@WCSDR ;LOADED CORRECTLY?
10297 037646 001030          BNE      6$           ;NO, ERROR
10298 037650 062777 002000 143160    ADD      #2000,@WCSAR  ;ADDRESS THE THIRD SEGMENT
10299 037656 011277 143156    MOV      (R2),@WCSDR  ;LOAD 3RD SEGMENT OF UWORD
10300 037662 022277 143152    CMP      (R2)+,@WCSDR ;LOADED CORRECTLY?
10301 037666 001020          BNE      6$           ;NO, ERROR
10302 037670 022712 125252    CMP      #UWDEND,(R2) ;END OF BLOCK?
10303 037674 001410          BEQ      3$           ;YES,EXIT
10304
10305 037676 162777 003777 143130    SUB      #3777,@WCSAR ;ADDRESS THE NEXT FIRST SEGMENT
10306 037704 005200          INC      R0           ;IF THE ENTRY POINT MICRO-WORD
10307 037706 001333          BNE      1$           ;HAS BEEN LOADED, CHANGE THE
10308 037710 010177 143120    MOV      R1,@WCSAR    ;WCSADR TO THE CONTROL BLOCK
10309 037714 000725          BR       2$           ;LOAD REST OF THE MICRO-CODE
10310
10311 037716 012603          35:  MOV      (SP)+,R3
10312 037720 012602          MOV      (SP)+,R2      ;RESTORE R0,R1,R2,R3
10313 037722 012601          MOV      (SP)+,R1
10314 037724 012600          MOV      (SP)+,R0
10315
10316
10317
10318
10319
10320 037726 000205          RTS      R5           ;EXIT
10321                                ;IF MICRO-CODE WAS LOADED INCORRECTLY ENTER HERE, REPORT
10322                                ;AN ERROR. THEN ABORT THE TEST.
10323
10324 037730 017737 143102 001162 65:  MOV      @WCSAR,$REG0  ;GET WCS ADDRESS
10325 037736 032777 006000 143072    BIT      #BIT10+BIT11,@WCSAR ;SEGMENT 0?
10326 037744 001403          BEQ      7$           ;YES
10327 037746 016237 177776 001164    MOV      -2(R2),$REG1  ;CORRECT WORD THAT SHOULD HAVE BEEN LOADED
10328 037754 017737 143060 001166 75:  MOV      @WCSDR,$REG2  ;INCORRECT WORD READ BACK
10329 037762 104070          ERROR   70           ;MICRO-WORD WAS LOADED IN THE WCS
10330                                ;INCORRECTLY. ERROR MESSAGE INDICATES
10331                                ;THE WCS ADDRESS WHERE THE MICRO-WORD SEGMENT
10332                                ;WAS LOADED, THE EXPCTD CORRECT WORD
10333                                ;AND THE WORD THAT WAS READ BACK.
10334

```



PDP-11/60 WCS DIAGNOSTIC  
DQM UAA P11 25-OCT-77 20:03

MACY11 30(1046) 25-OCT-77 20:12 PAGE 190  
END OF PASS ROUTINE

SEQ 0192  
SEQ 0203

10335	037764	104401	042652	TYPE,	MSG17	;ABORT TEST
10336	037770	013746	001102	MOV	\$TSTNM, -(SP)	;TEST NO.
10337	037774	104402		TYPOC		
10338						
10339	037776	013705	003056	MOV	NXTST, R5	;SET UP RETURN ADDRESS TO BEGINING
10340	040002	000745		BR	35	;OF NEXT TEST
10341						



```
10342 .SBTTL SCOPE HANDLER ROUTINE
10343
10344 ;*****
10345 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
10346 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG. (DISPLAY<7: 0>)
10347 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15: 08>
10348 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10349 ;*SW14=1 LOOP ON TEST
10350 ;*SW11=1 INHIBIT ITERATIONS
10351 ;*SW09=1 LOOP ON ERROR
10352 ;*SW08=1 LOOP ON TEST IN SWR<7: 0>
10353 ;*CALL
10354 ;* SCOPE ; SCOPE=10T
10355
10356 040004 $SCOPE:
10357 040004 032777 040000 141126 1$: BIT #BIT14, @SWR ; LOOP ON PRESENT TEST?
10358 040012 001114 BNE $OVER ; YES IF SW14=1
10359 ;####START OF CODE FOR THE XOR TESTER####
10360 040014 000416 $XTSTR: BR 6$ ; IF RUNNING ON THE "XOR" TESTER CHANGE
10361 ; THIS INSTRUCTION TO A "NOP" (NOP=240)
10362 040016 013746 000004 MOV @#ERRVEC, -(SP) ; SAVE THE CONTENTS OF THE ERROR VECTOR
10363 040022 012737 040042 000004 MOV #5$, @#ERRVEC ; SET FOR TIMEOUT
10364 040030 005737 177060 TST @#177060 ; TIME OUT ON XOR?
10365 040034 012637 000004 MOV (SP)+, @#ERRVEC ; RESTORE THE ERROR VECTOR
10366 040040 000463 BR $SVLAD ; GO TO THE NEXT TEST
10367 040042 022626 5$: CMP (SP)+, (SP)+ ; CLEAR THE STACK AFTER A TIME OUT
10368 040044 012637 000004 MOV (SP)+, @#ERRVEC ; RESTORE THE ERROR VECTOR
10369 040050 000423 BR 7$ ; LOOP ON THE PRESENT TEST
10370 040052 6$: ;####END OF CODE FOR THE XOR TESTER####
10371 040052 032777 000400 141060 BIT #BIT08, @SWR ; LOOP ON SPEC. TEST?
10372 040060 001404 BEQ 2$ ; BR IF NO
10373 040062 127737 141052 001102 CMPB @SWR, $TSTNM ; ON THE RIGHT TEST? SWR<7: 0>
10374 040070 001465 BEQ $OVER ; BR IF YES
10375 040072 105737 001103 2$: TSTB $ERFLG ; HAS AN ERROR OCCURRED?
10376 040076 001421 BEQ 3$ ; BR IF NO
10377 040100 123737 001115 001103 CMPB $ERMAX, $ERFLG ; MAX. ERRORS FOR THIS TEST OCCURRED?
10378 040106 101015 BHI 3$ ; BR IF NO
10379 040110 032777 001000 141022 BIT #BIT09, @SWR ; LOOP ON ERROR?
10380 040116 001404 BEQ 4$ ; BR IF NO
10381 040120 013737 001110 001106 7$: MOV $LPERR, $LPADR ; SET LOOP ADDRESS TO LAST SCOPE
10382 040126 000446 BR $OVER
10383 040130 105037 001103 4$: CLRB $ERFLG ; ZERO THE ERROR FLAG
10384 040134 005037 001222 CLR $TIMES ; CLEAR THE NUMBER OF ITERATIONS TO MAKE
10385 040140 000415 BR 1$ ; ESCAPE TO THE NEXT TEST
10386 040142 032777 004000 140770 3$: BIT #BIT11, @SWR ; INHIBIT ITERATIONS?
10387 040150 001011 BNE 1$ ; BR IF YES
10388 040152 005737 001244 TST $PASS ; IF FIRST PASS OF PROGRAM
10389 040156 001406 BEQ 1$ ; INHIBIT ITERATIONS
10390 040160 005237 001104 INC $ICNT ; INCREMENT ITERATION COUNT
10391 040164 023737 001222 001104 CMP $TIMES, $ICNT ; CHECK THE NUMBER OF ITERATIONS MADE
10392 040172 002024 BGE $OVER ; BR IF MORE ITERATION REQUIRED
10393 040174 012737 000001 001104 1$: MOV #1, $ICNT ; REINITIALIZE THE ITERATION COUNTER
10394 040202 013737 040260 001222 MOV $MXCNT, $TIMES ; SET NUMBER OF ITERATIONS TO DO
10395 040210 105237 001102 $SVLAD: INCB $TSTNM ; COUNT TEST NUMBERS
10396 040214 113737 001102 001242 MOVB $TSTNM, $TESTN ; SET TEST NUMBER IN APT MAILBOX
10397 040222 011637 001106 MOV (SP), $LPADR ; SAVE SCOPE LOOP ADDRESS
```



```

10398 040226 011637 001110      MOV      (SP), $LPERR      ;; SAVE ERROR LOOP ADDRESS
10399 040232 005037 001224      CLR      $ESCAPE          ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
10400 040236 112737 000001 001115  MOVVB   #1, $SERMAX       ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
10401 040244 013777 001102 140670 SOVER:  MOV      $TSTNM, @DISPLAY ;; DISPLAY TEST NUMBER
10402 040252 013716 001106      MOV      $LPADR, (SP)     ;; FUDGE RETURN ADDRESS
10403 040256 000002      RTI                      ;; FIXES PS
10404 040260 000010      SMXCNT: 10                ;; MAX. NUMBER OF ITERATIONS
10405
10406      .SBTTL  ERROR HANDLER ROUTINE
10407
10408      ;; *****
10409      ;; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
10410      ;; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
10411      ;; *AND GO TO $ERRTYP ON ERROR
10412      ;; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
10413      ;; *SW15=1      HALT ON ERROR
10414      ;; *SW13=1      INHIBIT ERROR TYPEOUTS
10415      ;; *SW10=1      BELL ON ERROR
10416      ;; *SW09=1      LOOP ON ERROR
10417      ;; *CALL
10418      ;; *      ERROR      N      ;; ERROR=EMT AND N=ERROR ITEM NUMBER
10419
10419 040262      $ERROR:
10420 040262 105237 001103 7$:      INCB     $ERFLG          ;; SET THE ERROR FLAG
10421 040266 001775      BEQ      7$              ;; DON'T LET THE FLAG GO TO ZERO
10422 040270 013777 001102 140644  MOV      $TSTNM, @DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
10423 040276 032777 002000 140634  BIT      #BIT10, @SWR     ;; BELL ON ERROR?
10424 040304 001402      BEQ      1$              ;; NO - SKIP
10425 040306 104401 001226      TYPE     , $BELL         ;; RING BELL
10426 040312 005237 001112 1$:      INC      $ERTTL         ;; COUNT THE NUMBER OF ERRORS
10427 040316 011637 001116      MOV      (SP), $ERRPC    ;; GET ADDRESS OF ERROR INSTRUCTION
10428 040322 162737 000002 001116  SUB      #2, $ERRPC
10429 040330 117737 140562 001114  MOVVB   @ $ERRPC, $ITEMB  ;; STRIP AND SAVE THE ERROR ITEM CODE
10430 040336 032777 020000 140574  BIT      #BIT13, @SWR     ;; SKIP TYPEOUT IF SET
10431 040344 001004      BNE      20$            ;; SKIP TYPEOUTS
10432 040346 004737 040456      JSR      PC, $ERRTYP     ;; GO TO USER ERROR ROUTINE
10433 040352 104401 001233      TYPE     , $CRLF
10434 040356      20$:
10435 040356 122737 000001 001256  CMPB    #APTENV, $ENV     ;; RUNNING IN APT MODE
10436 040364 001007      BNE      2$              ;; NO, SKIP APT ERROR REPORT
10437 040366 113737 001114 040400  MOVVB   $ITEMB, 21$      ;; SET ITEM NUMBER AS ERROR NUMBER
10438 040374 004737 041564      JSR      PC, $ATY4       ;; REPORT FATAL ERROR TO APT
10439 040400      21$:      .BYTE    0
10440 040401      .BYTE    0
10441 040402 000777      22$:      BR      22$              ;; APT ERROR LOOP
10442 040404 005777 140530 2$:      TST     @SWR             ;; HALT ON ERROR
10443 040410 100001      BPL      3$              ;; SKIP IF CONTINUE
10444 040412 000000      HALT
10445 040414 032777 001000 140516 3$:      BIT     #BIT09, @SWR     ;; LOOP ON ERROR SWITCH SET?
10446 040422 001402      BEQ      4$              ;; BR IF NO
10447 040424 013716 001110      MOV      $LPERR, (SP)    ;; FUDGE RETURN FOR LOOPING
10448 040430 005737 001224 4$:      TST     $ESCAPE         ;; CHECK FOR AN ESCAPE ADDRESS
10449 040434 001402      BEQ      5$              ;; BR IF NONE
10450 040436 013716 001224      MOV      $ESCAPE, (SP)  ;; FUDGE RETURN ADDRESS FOR ESCAPE
10451 040442      5$:
10452 040442 022737 036514 000042  CMP     # $ENDAD, @#42    ;; ACT-11 AUTO-ACCEPT?
10453 040450 001001      BNE      6$              ;; BRANCH IF NO

```



10454 040452 000000  
10455 040454  
10456 040454 000002  
10457  
10458  
10459  
10460  
10461  
10462  
10463  
10464 040456  
10465 040456 104401 001233  
10466 040462 010046  
10467 040464 005000  
10468 040466 153700 001114  
10469 040472 001004  
10470  
10471 040474 013746 001116  
10472  
10473 040500 104402  
10474 040502 000426  
10475 040504 005300  
10476 040506 006300  
10477 040510 006300  
10478 040512 006300  
10479 040514 062700 001362  
10480 040520 012037 040530  
10481 040524 001404  
10482 040526 104401  
10483 040530 000000  
10484 040532 104401 001233  
10485 040536 012037 040546  
10486 040542 001404  
10487 040544 104401  
10488 040546 000000  
10489 040550 104401 001233  
10490 040554 011000  
10491 040556 001004  
10492 040560 012600  
10493 040562 104401 001233  
10494 040566 000207  
10495 040570  
10496 040570 013046  
10497 040572 104402  
10498 040574 005710  
10499 040576 001770  
10500 040600 104401 040606  
10501 040604 000771  
10502 040606 020040 000  
10503 040612  
10504  
10505  
10506  
10507  
10508  
10509

```
HALT ;; YES
65: RTI ;; RETURN
.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

;*****
;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

SERRTYP:
      TYPE , $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"
      MOV RO, -(SP) ;; SAVE RO
      CLR RO ;; PICKUP THE ITEM INDEX
      BISB @#$ITEMB, RO
      BNE 15 ;; IF ITEM NUMBER IS ZERO, JUST
                ;; TYPE THE PC OF THE ERROR
      MOV $ERRPC, -(SP) ;; SAVE $ERRPC FOR TYPEOUT
                ;; ERROR ADDRESS
      TYP0C ;; GO TYPE--OCTAL ASCII (ALL DIGITS)
      BR 65 ;; GET OUT
15: DEC RO ;; ADJUST THE INDEX SO THAT IT WILL
      ASL RO ;; WORK FOR THE ERROR TABLE
      ASL RO
      ASL RO
      ADD #$SERRTB, RO ;; FORM TABLE POINTER
      MOV (RO)+, 25 ;; PICKUP "ERROR MESSAGE" POINTER
      BEQ 35 ;; SKIP TYPEOUT IF NO POINTER
      TYPE ;; TYPE THE "ERROR MESSAGE"
25: .WORD 0 ;; "ERROR MESSAGE" POINTER GOES HERE
      TYPE , $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"
35: MOV (RO)+, 45 ;; PICKUP "DATA HEADER" POINTER
      BEQ 55 ;; SKIP TYPEOUT IF 0
      TYPE ;; TYPE THE "DATA HEADER"
45: .WORD 0 ;; "DATA HEADER" POINTER GOES HERE
      TYPE , $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"
55: MOV (RO), RO ;; PICKUP "DATA TABLE" POINTER
      BNE 75 ;; GO TYPE THE DATA
65: MOV (SP)+, RO ;; RESTORE RO
      TYPE , $CRLF ;; "CARRIAGE RETURN" & "LINE FEED"
      RTS PC ;; RETURN
75: MOV @ (RO)+, -(SP) ;; SAVE @ (RO)+ FOR TYPEOUT
      TYP0C ;; GO TYPE--OCTAL ASCII (ALL DIGITS)
      TST (RO) ;; IS THERE ANOTHER NUMBER?
      BEQ 65 ;; BR IF NO
      TYPE , 35 ;; TYPE TWO(2) SPACES
      BR 75 ;; LOOP
85: .ASCIZ / / ;; TWO(2) SPACES
      .EVEN
.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

;*****
;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;*OCTAL (ASCII) NUMBER AND TYPE IT.
;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
```



```

10510 ;*CALL:
10511 ;*      MOV      NUM, -(SP)      ;;NUMBER TO BE TYPED
10512 ;*      TYPOS    ;;CALL FOR TYPEOUT
10513 ;*      .BYTE   N                ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
10514 ;*      .BYTE   M                ;;M=1 OR 0
10515 ;*                                     ;;1=TYPE LEADING ZEROS
10516 ;*                                     ;;0=SUPPRESS LEADING ZEROS
10517 ;*
10518 ;*STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
10519 ;*STYPOS OR STYPOC
10520 ;*CALL:
10521 ;*      MOV      NUM, -(SP)      ;;NUMBER TO BE TYPED
10522 ;*      TYPON    ;;CALL FOR TYPEOUT
10523 ;*
10524 ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
10525 ;*CALL:
10526 ;*      MOV      NUM, -(SP)      ;;NUMBER TO BE TYPED
10527 ;*      TYPOC    ;;CALL FOR TYPEOUT
10528
10529 040612 017646 000000          STYPOS: MOV      @ (SP), -(SP)      ;;PICKUP THE MODE
10530 040616 116637 000001 041035  MOVB     1 (SP), %OFILL      ;;LOAD ZERO FILL SWITCH
10531 040624 112637 041037          MOVB     (SP)+, %OMODE+1    ;;NUMBER OF DIGITS TO TYPE
10532 040630 062716 000002          ADD      #2, (SP)         ;;ADJUST RETURN ADDRESS
10533 040634 000406          BR      STYPON
10534 040636 112737 000001 041035  STYPOC: MOVB     #1, %OFILL      ;;SET THE ZERO FILL SWITCH
10535 040644 112737 000006 041037  MOVB     #6, %OMODE+1    ;;SET FOR SIX(6) DIGITS
10536 040652 112737 000005 041034  STYPON: MOVB     #5, %OCNT      ;;SET THE ITERATION COUNT
10537 040660 010346          MOV      R3, -(SP)        ;;SAVE R3
10538 040662 010446          MOV      R4, -(SP)        ;;SAVE R4
10539 040664 010546          MOV      R5, -(SP)        ;;SAVE R5
10540 040666 113704 041037          MOVB     %OMODE+1, R4    ;;GET THE NUMBER OF DIGITS TO TYPE
10541 040672 005404          NEG      R4
10542 040674 062704 000006          ADD      #6, R4          ;;SUBTRACT IT FOR MAX. ALLOWED
10543 040700 110437 041036          MOVB     R4, %OMODE      ;;SAVE IT FOR USE
10544 040704 113704 041035          MOVB     %OFILL, R4      ;;GET THE ZERO FILL SWITCH
10545 040710 016605 000012          MOV      12(SP), R5      ;;PICKUP THE INPUT NUMBER
10546 040714 005003          CLR      R3              ;;CLEAR THE OUTPUT WORD
10547 040716 006105          1%:    ROL      R5          ;;ROTATE MSB INTO "C"
10548 040720 000404          BR      3%              ;;GO DO MSB
10549 040722 006105          2%:    ROL      R5          ;;FORM THIS DIGIT
10550 040724 006105          ROL      R5
10551 040726 006105          ROL      R5
10552 040730 010503          MOV      R5, R3
10553 040732 006103          3%:    ROL      R3          ;;GET LSB OF THIS DIGIT
10554 040734 105337 041036          DECB     %OMODE          ;;TYPE THIS DIGIT?
10555 040740 100016          BPL      7%              ;;BR IF NO
10556 040742 042703 177770          BIC      #177770, R3     ;;GET RID OF JUNK
10557 040746 001002          BNE      4%              ;;TEST FOR 0
10558 040750 005704          TST      R4              ;;SUPPRESS THIS 0?
10559 040752 001403          BEQ      5%              ;;BR IF YES
10560 040754 005204          4%:    INC      R4          ;;DON'T SUPPRESS ANYMORE 0'S
10561 040756 052703 000060          BIS      #'0, R3        ;;MAKE THIS DIGIT ASCII
10562 040762 052703 000040          5%:    BIS      #' , R3    ;;MAKE ASCII IF NOT ALREADY
10563 040766 110337 041032          MOVB     R3, 8%         ;;SAVE FOR TYPING
10564 040772 104401 041032          TYPE     , 8%           ;;GO TYPE THIS DIGIT
10565 040776 105337 041034          7%:    DECB     %OCNT      ;;COUNT BY 1

```



```

10566 041002 003347      BGT      25      ;;BR IF MORE TO DO
10567 041004 002402      BLT      65      ;;BR IF DONE
10568 041006 005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
10569 041010 000744      BR       25      ;;GO DO THE LAST DIGIT
10570 041012 012605      65:    MOV      (SP)+,R5      ;;RESTORE R5
10571 041014 012604      MOV      (SP)+,R4      ;;RESTORE R4
10572 041016 012603      MOV      (SP)+,R3      ;;RESTORE R3
10573 041020 016666 000002 000004      MOV      2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
10574 041026 012616      MOV      (SP)+,(SP)
10575 041030 000002      RTI      ;;RETURN
10576 041032 000      85:    .BYTE    0      ;;STORAGE FOR ASCII DIGIT
10577 041033 000      .BYTE    0      ;;TERMINATOR FOR TYPE ROUTINE
10578 041034 000      SOCNT:  .BYTE    0      ;;OCTAL DIGIT COUNTER
10579 041035 000      $OFILL: .BYTE    0      ;;ZERO FILL SWITCH
10580 041036 000000      $OMODE: .WORD    0      ;;NUMBER OF DIGITS TO TYPE
10581      .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
10582
10583      ;;*****
10584      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
10585      ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
10586      ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
10587      ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
10588      ;*REPLACED WITH SPACES.
10589      ;*CALL:
10590      ;*      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
10591      ;*      TYPDS      ;;GO TO THE ROUTINE
10592
10593      $TYPDS:
10594 041040      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
10595 041042 010046      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
10596 041044 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
10597 041046 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
10598 041050 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
10599 041052 012746 020200      MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
10600 041056 016605 000020      MOV      20(SP),R5      ;;GET THE INPUT NUMBER
10601 041062 100004      BPL      15      ;;BR IF INPUT IS POS.
10602 041064 005405      NEG      R5      ;;MAKE THE BINARY NUMBER POS.
10603 041066 112766 000055 000001      MOV      #'-',1(SP)      ;;MAKE THE ASCII NUMBER NEG.
10604 041074 005000      15:    CLR      R0      ;;ZERO THE CONSTANTS INDEX
10605 041076 012703 041254      MOV      #$DBLK,R3      ;;SETUP THE OUTPUT POINTER
10606 041102 112723 000040      MOV      #'',(R3)+      ;;SET THE FIRST CHARACTER TO A BLANK
10607 041106 005002      25:    CLR      R2      ;;CLEAR THE BCD NUMBER
10608 041110 016001 041244      MOV      $DTBL(R0),R1      ;;GET THE CONSTANT
10609 041114 160105      35:    SUB      R1,R5      ;;FORM THIS BCD DIGIT
10610 041116 002402      BLT      45      ;;BR IF DONE
10611 041120 005202      INC      R2      ;;INCREASE THE BCD DIGIT BY 1
10612 041122 000774      BR       35
10613 041124 060105      45:    ADD      R1,R5      ;;ADD BACK THE CONSTANT
10614 041126 005702      TST      R2      ;;CHECK IF BCD DIGIT=0
10615 041130 001002      BNE      55      ;;FALL THROUGH IF 0
10616 041132 105716      TSTB     (SP)      ;;STILL DOING LEADING 0'S?
10617 041134 100407      BMI      75      ;;BR IF YES
10618 041136 106316      55:    ASLB     (SP)      ;;MSD?
10619 041140 103003      BCC      65      ;;BR IF NO
10620 041142 116663 000001 177777      MOV      1(SP),-1(R3)      ;;YES--SET THE SIGN
10621 041150 052702 000060      65:    BIS      #'0,R2      ;;MAKE THE BCD DIGIT ASCII

```

```

10622 041154 052702 000040      7$:   BIS      #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
10623 041160 110223              MOVB     R2,(R3)+     ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
10624 041162 005720              TST      (R0)+       ;;JUST INCREMENTING
10625 041164 020027 000010      CMP      RO,#10      ;;CHECK THE TABLE INDEX
10626 041170 002746              BLT      2$          ;;GO DO THE NEXT DIGIT
10627 041172 003002              BGT      8$          ;;GO TO EXIT
10628 041174 010502              MOV      R5,R2       ;;GET THE LSD
10629 041176 000764              BR       6$          ;;GO CHANGE TO ASCII
10630 041200 105726      8$:   TSTB     (SP)+     ;;WAS THE LSD THE FIRST NON-ZERO?
10631 041202 100003              BPL      9$          ;;BR IF NO
10632 041204 116663 177777 177776 MOVB     -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
10633 041212 105013      9$:   CLRB     (R3)       ;;SET THE TERMINATOR
10634 041214 012605              MOV      (SP)+,R5    ;;POP STACK INTO R5
10635 041216 012603              MOV      (SP)+,R3    ;;POP STACK INTO R3
10636 041220 012602              MOV      (SP)+,R2    ;;POP STACK INTO R2
10637 041222 012601              MOV      (SP)+,R1    ;;POP STACK INTO R1
10638 041224 012600              MOV      (SP)+,R0    ;;POP STACK INTO R0
10639 041226 104401 041254      TYPE    ,SDBLK       ;;NOW TYPE THE NUMBER
10640 041232 016666 000002 000004 MOV      2(SP),4(SP)  ;;ADJUST THE STACK
10641 041240 012616              MOV      (SP)+,(SP)
10642 041242 000002              RTI                          ;;RETURN TO USER
10643 041244 023420      SDBLK: 10000.
10644 041246 001750              1000.
10645 041250 000144              100.
10646 041252 000012              10.
10647 041254 000004      SDBLK: .BLKW 4
10648              .SBTTL TYPE ROUTINE
10649
10650      ;;*****
10651      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
10652      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
10653      ;*NOTE1:          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
10654      ;*NOTE2:          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
10655      ;*NOTE3:          $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
10656      ;*
10657      ;*CALL:
10658      ;*1) USING A TRAP INSTRUCTION
10659      ;*   TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
10660      ;*OR
10661      ;*   TYPE
10662      ;*   MESADR
10663      ;*
10664
10665 041264 105737 001157      $TYPE: TSTB     $TPFLG   ;; IS THERE A TERMINAL?
10666 041270 100002              BPL      1$          ;;BR IF YES
10667 041272 000000              HALT     ;;HALT HERE IF NO TERMINAL
10668 041274 000430              BR       3$          ;;LEAVE
10669 041276 010046      1$:   MOV      RO,-(SP)   ;;SAVE RO
10670 041300 017600 000002      MOV      @2(SP),RO   ;;GET ADDRESS OF ASCIZ STRING
10671 041304 122737 000001 001256 CMPB     #APTENV,$ENV ;;RUNNING IN APT MODE
10672 041312 001011              BNE      62$         ;;NO, GO CHECK FOR APT CONSOLE
10673 041314 132737 000100 001257 BITB     #APTPOOL,$ENV ;;SPOOL MESSAGE TO APT
10674 041322 001405              BEQ      62$         ;;NO, GO CHECK FOR CONSOLE
10675 041324 010037 041334      MOV      RO,61$     ;;SETUP MESSAGE ADDRESS FOR APT
10676 041330 004737 041554      JSR     PC,$ATY3    ;;SPOOL MESSAGE TO APT
10677 041334 000000      61$: .WORD    0      ;;MESSAGE ADDRESS

```



```

10678 041336 132737 000040 001257 62$: BITB #APTCSUP,$ENVM ;;APT CONSOLE SUPPRESSED
10679 041344 001003 BNE 60$ ;;YES,SKIP TYPE OUT
10680 041346 112046 2$: MOVB (RO)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
10681 041350 001005 BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
10682 041352 005726 TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
10683 041354 012600 60$: MOV (SP)+,RO ;;RESTORE RO
10684 041356 062716 000002 3$: ADD #2,(SP) ;;ADJUST RETURN PC
10685 041362 000002 RTI ;;RETURN
10686 041364 122716 000011 4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
10687 041370 001430 BEQ 8$
10688 041372 122716 000200 CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
10689 041376 001006 BNE 5$
10690 041400 005726 TST (SP)+ ;;POP <CR><LF> EQUIV
10691 041402 104401 TYPE ;;TYPE A CR AND LF
10692 041404 001233 $CRLF
10693 041406 105037 041542 CLRB $CHARCNT ;;CLEAR CHARACTER COUNT
10694 041412 000755 BR 2$ ;;GET NEXT CHARACTER
10695 041414 004737 041476 5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
10696 041420 123726 001156 6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS. ?
10697 041424 001350 BNE 2$ ;;IF NO GO GET NEXT CHAR.
10698 041426 013746 001154 MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
10699 AND THE NULL CHAR.
10700 041432 105366 000001 7$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?
10701 041436 002770 BLT 6$ ;;BR IF NO--GO POP THE NULL OFF OF STACK
10702 041440 004737 041476 JSR PC,$TYPEC ;;GO TYPE A NULL
10703 041444 105337 041542 DECB $CHARCNT ;;DO NOT COUNT AS A COUNT
10704 041450 000770 BR 7$ ;;LOOP
10705
10706 ; HORIZONTAL TAB PROCESSOR
10707
10708 041452 112716 000040 8$: MOVB #' ,(SP) ;;REPLACE TAB WITH SPACE
10709 041456 004737 041476 9$: JSR PC,$TYPEC ;;TYPE A SPACE
10710 041462 132737 000007 041542 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
10711 041470 001372 BNE 9$ ;;TAB STOP
10712 041472 005726 TST (SP)+ ;;POP SPACE OFF STACK
10713 041474 000724 BR 2$ ;;GET NEXT CHARACTER
10714 041476 105777 137446 $TYPEC: TSTB @STPS ;;WAIT UNTIL PRINTER IS READY
10715 041502 100375 BPL $TYPEC
10716 041504 116677 000002 137440 MOVB 2(SP),@STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
10717 041512 122766 000015 000002 CMPB #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
10718 041520 001003 BNE 1$ ;;BRANCH IF NO
10719 041522 105037 041542 CLRB $CHARCNT ;;YES--CLEAR CHARACTER COUNT
10720 041526 000406 BR $TYPEX ;;EXIT
10721 041530 122766 000012 000002 1$: CMPB #LF,2(SP) ;;IS CHARACTER A LINE FEED?
10722 041536 001402 BEQ $TYPEX ;;BRANCH IF YES
10723 041540 105227 INCB (PC)+ ;;COUNT THE CHARACTER
10724 041542 000000 $CHARCNT: WORD 0 ;;CHARACTER COUNT STORAGE
10725 041544 000207 $TYPEX: RTS PC
10726
10727 .SBTTL APT COMMUNICATIONS ROUTINE
10728
10729 ;*****
10730 041546 112737 000001 042012 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
10731 041554 112737 000001 042010 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
10732 041562 000403 BR $ATYC
10733 041564 112737 000001 042012 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR

```

```
10734 041572          SATYC:
10735 041572 010046      MOV      RO,-(SP)      ;;PUSH RO ON STACK
10736 041574 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
10737 041576 105737 042010  TSTB     $MFLG        ;;SHOULD TYPE A MESSAGE?
10738 041602 001450      BEQ      5$           ;;IF NOT: BR
10739 041604 122737 000001 001256  CMPB     #APTENV,$ENV  ;;OPERATING UNDER APT?
10740 041612 001031      BNE     3$           ;;IF NOT: BR
10741 041614 132737 000100 001257  BITB     #APTPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
10742 041622 001425      BEQ      3$           ;;IF NOT: BR
10743 041624 017600 000004      MOV     @4(SP),RO     ;;GET MESSAGE ADDR.
10744 041630 062766 000002 000004  ADD     #2,4(SP)      ;;BUMP RETURN ADDR.
10745 041636 005737 001236      15:    TST     $MSGTYPE   ;;SEE IF DONE W/ LAST XMISSION?
10746 041642 001375      BNE     1$           ;;IF NOT: WAIT
10747 041644 010037 001252      MOV     RO,$MSGAD     ;;PUT ADDR IN MAILBOX
10748 041650 105720      25:    TSTB     (RO)+    ;;FIND END OF MESSAGE
10749 041652 001376      BNE     2$
10750 041654 163700 001252      SUB     $MSGAD,RO     ;;SUB START OF MESSAGE
10751 041660 006200      ASR     RO           ;;GET MESSAGE LGTH IN WORDS
10752 041662 010037 001254      MOV     RO,$MSGGLT   ;;PUT LENGTH IN MAILBOX
10753 041666 012737 000004 001236  MOV     #4,$MSGTYPE  ;;TELL APT TO TAKE MSG.
10754 041674 000413      BR      5$
10755 041676 017637 000004 041722 35:    MOV     @4(SP),4$    ;;PUT MSG ADDR IN JSR LINKAGE
10756 041704 062766 000002 000004  ADD     #2,4(SP)      ;;BUMP RETURN ADDRESS
10757 041712 013746 177776      MOV     177776,-(SP) ;;PUSH 177776 ON STACK
10758 041716 004737 041264      JSR     PC,$TYPE     ;;CALL TYPE MACRO
10759 041722 000000      45:    .WORD  0
10760 041724          55:
10761 041724 105737 042012      105:   TSTB     $FFLG        ;;SHOULD REPORT FATAL ERROR?
10762 041730 001416      BEQ     12$         ;;IF NOT: BR
10763 041732 005737 001256      TST     $ENV         ;;RUNNING UNDER APT?
10764 041736 001413      BEQ     12$         ;;IF NOT: BR
10765 041740 005737 001236      115:   TST     $MSGTYPE   ;;FINISHED LAST MESSAGE?
10766 041744 001375      BNE     11$        ;;IF NOT: WAIT
10767 041746 017637 000004 001240  MOV     @4(SP),$FATAL ;;GET ERROR #
10768 041754 062766 000002 000004  ADD     #2,4(SP)      ;;BUMP RETURN ADDR.
10769 041762 005237 001236      INC     $MSGTYPE     ;;TELL APT TO TAKE ERROR
10770 041766 105037 042012      125:   CLRB    $FFLG        ;;CLEAR FATAL FLAG
10771 041772 105037 042011      CLRB    $LFLG        ;;CLEAR LOG FLAG
10772 041776 105037 042010      CLRB    $MFLG        ;;CLEAR MESSAGE FLAG
10773 042002 012601      MOV     (SP)+,R1     ;;POP STACK INTO R1
10774 042004 012600      MOV     (SP)+,RO     ;;POP STACK INTO RO
10775 042006 000207      RTS     PC           ;;RETURN
10776 042010 000          $MFLG: .BYTE 0      ;;MESSG. FLAG
10777 042011 000          $LFLG: .BYTE 0      ;;LOG FLAG
10778 042012 000          $FFLG: .BYTE 0      ;;FATAL FLAG
10779          042014          .EVEN
10780          000200      APTSIZE=200
10781          000001      APTENV=001
10782          000100      APTPOOL=100
10783          000040      APTCSUP=040
10784          .SBTTL TRAP DECODER
10785
10786          ;;*****
10787          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10788          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10789          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
```



```
10790 ;*GO TO THAT ROUTINE.
10791
10792 042014 010046 STRAP: MOV RO, -(SP) ;;SAVE RO
10793 042016 016600 000002 MOV 2(SP),RO ;;GET TRAP ADDRESS
10794 042022 005740 TST -(RO) ;;BACKUP BY 2
10795 042024 111000 MOVB (RO),RO ;;GET RIGHT BYTE OF TRAP
10796 042026 006300 ASL RO ;;POSITION FOR INDEXING
10797 042030 016000 042050 MOV STRPAD(RO),RO ;;INDEX TO TABLE
10798 042034 000200 RTS RO ;;GO TO ROUTINE
10799
10800
10801 ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
10802
10803 042036 011646 STRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
10804 042040 016666 000004 000002 MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
10805 042046 000002 RTI ;;RESTORE THE PSW
10806
10807 .SBTTL TRAP TABLE
10808
10809 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10810 ;*BY THE "TRAP" INSTRUCTION.
10811
10812 ; ROUTINE
10813 ; -----
10814 042050 042036 STRPAD: .WORD STRAP2
10815 042052 041264 $TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
10816 042054 040636 $TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
10817 042056 040612 $TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
10818 042060 040652 $TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
10819 042062 041040 $TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
10820
10821
10822 042064 036574 TY.MSG ;;CALL=TYPMSG TRAP+6(104406) TYPE ROUTINE WITH SW13 FOR INHIBIT
10823 .SBTTL POWER DOWN AND UP ROUTINES
10824
10825 ;*****
10826 ;POWER DOWN ROUTINE
10827 042066 012737 042246 000024 $PWRDN: MOV $ILLUP,@#PWRVEC ;;SET FOR FAST UP
10828 042074 012737 000340 000026 MOV #340,@#PWRVEC+2 ;;PRIO:7
10829 042102 010046 MOV RO, -(SP) ;;PUSH RO ON STACK
10830 042104 010146 MOV R1, -(SP) ;;PUSH R1 ON STACK
10831 042106 010246 MOV R2, -(SP) ;;PUSH R2 ON STACK
10832 042110 010346 MOV R3, -(SP) ;;PUSH R3 ON STACK
10833 042112 010446 MOV R4, -(SP) ;;PUSH R4 ON STACK
10834 042114 010546 MOV R5, -(SP) ;;PUSH R5 ON STACK
10835 042116 017746 137016 MOV @SWR, -(SP) ;;PUSH @SWR ON STACK
10836 042122 010637 042252 MOV SP, $SAVR6 ;;SAVE SP
10837 042126 012737 042140 000024 MOV $PWRUP,@#PWRVEC ;;SET UP VECTOR
10838 042134 000000 HALT
10839 042136 000776 BR -2 ;;HANG UP
10840
10841 ;*****
10842 ;POWER UP ROUTINE
10843 042140 012737 042246 000024 $PWRUP: MOV $ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
10844 042146 013706 042252 MOV $SAVR6, SP ;;GET SP
10845 042152 005037 042252 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
```

```

10846 042156 005237 042252      15:   INC   $$AVR6      ;;WAIT FOR THE INC
10847 042162 001375              BNE   15          ;;OF WORD
10848 042164 011600              MOV   (SP),R0    ;GET SAVED SWR OFF STACK
10849 042166 076600 000226      MED   ,226       ;AND RESTORE AS 'CNLS. SW'
10850 042172 012737 003344 000114 MOV   #BADPAR,@#114 ;POINT PARITY-ERR-HNDLER AT GENERAL ROUTINE
10851
10852 042200 012677 136734      MOV   (SP)+,@SWR  ;;POP STACK INTO @SWR
10853 042204 012605              MOV   (SP)+,R5   ;;POP STACK INTO R5
10854 042206 012604              MOV   (SP)+,R4   ;;POP STACK INTO R4
10855 042210 012603              MOV   (SP)+,R3   ;;POP STACK INTO R3
10856 042212 012602              MOV   (SP)+,R2   ;;POP STACK INTO R2
10857 042214 012601              MOV   (SP)+,R1   ;;POP STACK INTO R1
10858 042216 012600              MOV   (SP)+,R0   ;;POP STACK INTO R0
10859 042220 012737 042066 000024 MOV   #SPWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
10860 042226 012737 000340 000026 MOV   #340,@#PWRVEC+2 ;;PRIO: 7
10861 042234 104401              TYPE  ;REPORT THE POWER FAILURE
10862 042236 042254      SPWRMG: .WORD  SPOWER  ;;POWER FAIL MESSAGE POINTER
10863 042240 012716              MOV   (PC)+,(SP) ;;RESTART AT RESTART
10864 042242 004006      SPWRAD: .WORD  RESTART  ;;RESTART ADDRESS
10865 042244 000002              RTI
10866 042246 000000      $ILLUP: HALT      ;;THE POWER UP SEQUENCE WAS STARTED
10867 042250 000776              BR    -2         ;; BEFORE THE POWER DOWN WAS COMPLETE
10868 042252 000000      $$AVR6: 0        ;;PUT THE SP HERE
10869 042254 005015 047520 042527 SPOWER: .ASCIZ  <15><12>"POWER"
10870 042262 000122
10871
10872

```



Line	Address	Offset	Value	Label	Text
10873				.SBTTL	MESSAGES
10874					
10875	042264	040520	030122	000	MSG2: .ASCIZ "PAR0"
10876	042271	120	051101	000061	MSG3: .ASCIZ "PAR1"
10877	042276	040520	031122	000	MSG4: .ASCIZ "PAR2"
10878	042303	120	051101	052111	MSG5: .ASCIZ "PARITY DISABLE"
10879	042310	020131	044504	040523	
10880	042316	046102	000105		
10881	042322	051127	052111	020105	MSG6: .ASCIZ "WRITE WRONG PARITY"
10882	042330	051127	047117	020107	
10883	042336	040520	044522	054524	
10884	042344	000			
10885	042345	127	044522	042524	MSG7: .ASCIZ "WRITE ENABLE"
10886	042352	042440	040516	046102	
10887	042360	000105			
10888	042362	040520	044522	054524	MSG8: .ASCIZ "PARITY GENERATED"
10889	042370	043440	047105	051105	
10890	042376	052101	042105	000	
10891	042403	102	052111	000070	MSG80: .ASCIZ "BIT8"
10892	042410	044502	034524	000	MSG81: .ASCIZ "BIT9"
10893	042415	102	052111	030061	MSG82: .ASCIZ "BIT10"
10894	042422	000			
10895	042423	102	052111	030461	MSG83: .ASCIZ "BIT11"
10896	042430	000			
10897	042431	102	052111	031061	MSG84: .ASCIZ "BIT12"
10898	042436	000			
10899	042437	102	052111	031461	MSG85: .ASCIZ "BIT13"
10900	042444	000			
10901	042445	102	052111	032061	MSG86: .ASCIZ "BIT14"
10902	042452	000			
10903	042453	120	051101	052111	MSG9: .ASCIZ "PARITY ERROR"
10904	042460	020131	051105	047522	
10905	042466	000122			
10906	042470	047040	052117	041440	MSG10: .ASCIZ " NOT CLEAR"<CR><LF>
10907	042476	042514	051101	005015	
10908	042504	000			
10909	042505	040	047516	020124	MSG11: .ASCIZ " NOT SET"<CR><LF>
10910	042512	042523	006524	000012	
10911	042520	005015	047125	054105	MSG12: .ASCIZ <CR><LF>"UNEXPECTED PARITY ERROR"
10912	042526	042520	052103	042105	
10913	042534	050040	051101	052111	
10914	042542	020131	051105	047522	
10915	042550	000122			
10916	042552	005015	047125	054105	MSG13: .ASCIZ <CR><LF>"UNEXPECTED CPU ERROR TRAP, CPUERR="
10917	042560	042520	052103	042105	
10918	042566	041440	052520	042440	
10919	042574	051122	051117	052040	
10920	042602	040522	026120	041440	
10921	042610	052520	051105	036522	
10922	042616	000			
10923	042617	015	044412	046114	MSG14: .ASCIZ <CR><LF>"ILLEGAL INSTRUCTION TRAP"
10924	042624	043505	046101	044440	
10925	042632	051516	051124	041525	
10926	042640	044524	047117	052040	
10927	042646	040522	000120		
10928	042652	052040	051505	020124	MSG17: .ASCIZ " TEST ABORTED"

10929	042660	041101	051117	042524			
10930	042666	000104					
10931	042670	051440	043505	042515	MSG20:	. ASCII	" SEGMENT "
10932	042676	052116	040				
10933	042701	106	052501	052114	MSG21:	. ASCII	"FAULTY ADDRESS LINE: "
10934	042706	020131	042101	051104			
10935	042714	051505	020123	044514			
10936	042722	042516	072				
10937	042725	040	047506	020122	MSG22:	. ASCIIZ	" FOR DATA ARRAY <11: 00>"<CR><LF>
10938	042732	040504	040524	040440			
10939	042740	051122	054501	036040			
10940	042746	030461	030072	037060			
10941	042754	005015	000				
10942	042757	015	040412	042116	MSG23:	. ASCIIZ	<CR><LF>"AND="
10943	042764	000075					
10944	042766	005015	051117	000075	MSG24:	. ASCIIZ	<CR><LF>"OR="
10945	042774	043040	051117	042040	MSG25:	. ASCIIZ	" FOR DATA ARRAY <23: 12>"<CR><LF>
10946	043002	052101	020101	051101			
10947	043010	040522	020131	031074			
10948	043016	035063	031061	006476			
10949	043024	000012					
10950	043026	043040	051117	042040	MSG26:	. ASCIIZ	" FOR DATA ARRAY <35: 24>"<CR><LF>
10951	043034	052101	020101	051101			
10952	043042	040522	020131	031474			
10953	043050	035065	032062	006476			
10954	043056	000012					
10955	043060	043040	051117	042040	MSG27:	. ASCIIZ	" FOR DATA ARRAY <47: 36>"<CR><LF>
10956	043066	052101	020101	051101			
10957	043074	040522	020131	032074			
10958	043102	035067	033063	006476			
10959	043110	000012					
10960	043112	005015	051120	043517	MSG28:	. ASCIIZ	<CR><LF>"PROGRAM ABORTED"
10961	043120	040522	020115	041101			
10962	043126	051117	042524	000104			
10963	043134	005015	047125	054105	MSG29:	. ASCIIZ	<CR><LF>"UNEXPECTED WCS PARITY ERROR"
10964	043142	042520	052103	042105			
10965	043150	053440	051503	050040			
10966	043156	051101	052111	020131			
10967	043164	051105	047522	000122			
10968	043172	005015	047125	054105	MSG30:	. ASCIIZ	<CR><LF>"UNEXPECTED MEMORY PARITY ERROR"
10969	043200	042520	052103	042105			
10970	043206	046440	046505	051117			
10971	043214	020131	040520	044522			
10972	043222	054524	042440	051122			
10973	043230	051117	000				
10974	043233	040	041520	000075	MSG31:	. ASCIIZ	" PC="
10975							
10976							
10977						. SBTTL	ERROR MESSAGES
10978							
10979	043240	044524	042515	052517	EM1:	. ASCIIZ	"TIMEOUT ON REFERENCING WCS REGISTER"
10980	043246	020124	047117	051040			
10981	043254	043105	051105	047105			
10982	043262	044503	043516	053440			
10983	043270	051503	051040	043505			
10984	043276	051511	042524	000122			



Address	Byte 1	Byte 2	Byte 3	Byte 4	Code	Message
10985	043304	053447	051503	050040	EM2:	.ASCIZ "'WCS PRESENT' BIT NOT SET IN WHAMI"
10986	043312	042522	042523	052116		
10987	043320	020047	044502	020124		
10988	043326	047516	020124	042523		
10989	043334	020124	047111	053440		
10990	043342	040510	044515	000		
10991	043347	047	051127	052111	EM3:	.ASCIZ "'WRITE ENABLE' BIT NOT SET"
10992	043354	020105	047105	041101		
10993	043362	042514	020047	044502		
10994	043370	020124	047516	020124		
10995	043376	042523	000124			
10996	043402	050047	051101	052111	EM4:	.ASCIZ "'PARITY DISABLE' BIT NOT CLEAR"
10997	043410	020131	044504	040523		
10998	043416	046102	023505	041040		
10999	043424	052111	047040	052117		
11000	043432	041440	042514	051101		
11001	043440	000				
11002	043441	047	051127	052111	EM5:	.ASCIZ "'WRITE WRONG PARITY' BIT NOT CLEAR"
11003	043446	020105	051127	047117		
11004	043454	020107	040520	044522		
11005	043462	054524	020047	044502		
11006	043470	020124	047516	020124		
11007	043476	046103	040505	000122		
11008	043504	050047	051101	052111	EM6:	.ASCIZ "'PARITY DISABLE' BIT NOT SET"
11009	043512	020131	044504	040523		
11010	043520	046102	023505	041040		
11011	043526	052111	047040	052117		
11012	043534	051440	052105	000		
11013	043541	047	040520	044522	EM7:	.ASCIZ "'PARITY ERROR' NOT CLEAR"
11014	043546	054524	042440	051122		
11015	043554	051117	020047	047516		
11016	043562	020124	046103	040505		
11017	043570	000122				
11018	043572	053447	044522	042524	EM10:	.ASCIZ "'WRITE WRONG PARITY' BIT NOT SET"
11019	043600	053440	047522	043516		
11020	043606	050040	051101	052111		
11021	043614	023531	041040	052111		
11022	043622	047040	052117	051440		
11023	043630	052105	000			
11024	043633	123	040524	052524	EM11:	.ASCIZ "STATUS ERROR ON WRITE TO WCS LOCATION"
11025	043640	020123	051105	047522		
11026	043646	020122	047117	053440		
11027	043654	044522	042524	052040		
11028	043662	020117	041527	020123		
11029	043670	047514	040503	044524		
11030	043676	047117	000			
11031	043701	104	052101	020101	EM12:	.ASCIZ "DATA ERROR ON READ FROM WCS LOCATION"
11032	043706	051105	047522	020122		
11033	043714	047117	051040	040505		
11034	043722	020104	051106	046517		
11035	043730	053440	051503	046040		
11036	043736	041517	052101	047511		
11037	043744	000116				
11038	043746	052123	052101	051525	EM13:	.ASCIZ "STATUS ERROR ON READ FROM WCS LOCATION"
11039	043754	042440	051122	051117		
11040	043762	047440	020116	042522		

11041	043770	042101	043040	047522	
11042	043776	020115	041527	020123	
11043	044004	047514	040503	044524	
11044	044012	047117	000		
11045	044015	124	040522	020120	EM14: .ASCIZ "TRAP AT LOCATION"
11046	044022	052101	046040	041517	
11047	044030	052101	047511	000116	
11048	044036	047111	047503	051122	EM15: .ASCIZ "INCORRECT PARITY BIT GENERATED FOR PATTERN"
11049	044044	041505	020124	040520	
11050	044052	044522	054524	041040	
11051	044060	052111	043440	047105	
11052	044066	051105	052101	042105	
11053	044074	043040	051117	050040	
11054	044102	052101	042524	047122	
11055	044110	000			
11056	044111	111	041516	051117	EM16: .ASCIZ "INCORRECT PARITY BIT GENERATED-BIT7-(WITH WWP) FOR PATTERN"
11057	044116	042522	052103	050040	
11058	044124	051101	052111	020131	
11059	044132	044502	020124	042507	
11060	044140	042516	040522	042524	
11061	044146	026504	044502	033524	
11062	044154	024055	044527	044124	
11063	044162	053440	050127	020051	
11064	044170	047506	020122	040520	
11065	044176	052124	051105	000116	
11066	044204	047125	054105	042520	EM17: .ASCIZ "UNEXPECTED PARITY TRAP ON READING BAD PARITY THOUGH 'PARDIS' WAS SET"
11067	044212	052103	042105	050040	
11068	044220	051101	052111	020131	
11069	044226	051124	050101	047440	
11070	044234	020116	042522	042101	
11071	044242	047111	020107	040502	
11072	044250	020104	040520	044522	
11073	044256	054524	052040	047510	
11074	044264	043525	020110	050047	
11075	044272	051101	044504	023523	
11076	044300	053440	051501	051440	
11077	044306	052105	000		
11078	044311	047	040520	044522	EM20: .ASCIZ "'PARITY ERROR' DID NOT SET ON PARITY ERROR TRAP"
11079	044316	054524	042440	051122	
11080	044324	051117	020047	044504	
11081	044332	020104	047516	020124	
11082	044340	042523	020124	047117	
11083	044346	050040	051101	052111	
11084	044354	020131	051105	047522	
11085	044362	020122	051124	050101	
11086	044370	000			
11087	044371	111	041516	051117	EM21: .ASCIZ "INCORRECT PARITY BIT STORED"
11088	044376	042522	052103	050040	
11089	044404	051101	052111	020131	
11090	044412	044502	020124	052123	
11091	044420	051117	042105	000	
11092	044425	116	020117	040520	EM22: .ASCIZ "NO PARITY ERROR TRAP ON READING WRONG PARITY"
11093	044432	044522	054524	042440	
11094	044440	051122	051117	052040	
11095	044446	040522	020120	047117	
11096	044454	051040	040505	044504	



11097	044462	043516	053440	047522	
11098	044470	043516	050040	051101	
11099	044476	052111	000131		
11100	044502	050047	051101	052111	EM23: .ASCIZ "'PARITY ERROR' BIT NOT SET ON READING BAD PARITY"
11101	044510	020131	051105	047522	
11102	044516	023522	041040	052111	
11103	044524	047040	052117	051440	
11104	044532	052105	047440	020116	
11105	044540	042522	042101	047111	
11106	044546	020107	040502	020104	
11107	044554	040520	044522	054524	
11108	044562	000			
11109	044563	123	052105	044524	EM24: .ASCIZ "SETTING 'PARDIS' DID NOT CLEAR 'PARERR'"
11110	044570	043516	023440	040520	
11111	044576	042122	051511	020047	
11112	044604	044504	020104	047516	
11113	044612	020124	046103	040505	
11114	044620	020122	050047	051101	
11115	044626	051105	023522	000	
11116	044633	102	042101	050040	EM25: .ASCIZ "BAD PARITY NOT DETECTED BY CHECKER"
11117	044640	051101	052111	020131	
11118	044646	047516	020124	042504	
11119	044654	042524	052103	042105	
11120	044662	041040	020131	044103	
11121	044670	041505	042513	000122	
11122	044676	040520	044522	054524	EM26: .ASCIZ "PARITY ERROR ON READING WCS"
11123	044704	042440	051122	051117	
11124	044712	047440	020116	042522	
11125	044720	042101	047111	020107	
11126	044726	041527	000123		
11127	044732	051124	050101	040440	EM27: .ASCIZ "TRAP AT LOCATION, ON DOING READ"
11128	044740	020124	047514	040503	
11129	044746	044524	047117	020054	
11130	044754	047117	042040	044517	
11131	044762	043516	051040	040505	
11132	044770	000104			
11133	044772	047503	051122	041505	EM30: .ASCIZ "CORRECT XFC-CODE WAS NOT SAVED AFTER 'OTHER USER' DISPATCH OF XFC"
11134	045000	020124	043130	026503	
11135	045006	047503	042504	053440	
11136	045014	051501	047040	052117	
11137	045022	051440	053101	042105	
11138	045030	040440	052106	051105	
11139	045036	023440	052117	042510	
11140	045044	020122	051525	051105	
11141	045052	020047	044504	050123	
11142	045060	052101	044103	047440	
11143	045066	020106	043130	000103	
11144	045074	040527	041524	026510	EM31: .ASCIZ "WATCH-DOG TIMED OUT AFTER 'OTHER USER' DISPATCH OF XFC"
11145	045102	047504	020107	044524	
11146	045110	042515	020104	052517	
11147	045116	020124	043101	042524	
11148	045124	020122	047447	044124	
11149	045132	051105	052440	042523	
11150	045140	023522	042040	051511	
11151	045146	040520	041524	020110	
11152	045154	043117	054040	041506	

11153	045162	000			
11154	045163	103	051117	042522	EM32: .ASCIZ "CORRECT XFC-CODE WAS NOT SAVED AFTER 'USER' DISPATCH OF XFC"
11155	045170	052103	054040	041506	
11156	045176	041455	042117	020105	
11157	045204	040527	020123	047516	
11158	045212	020124	040523	042526	
11159	045220	020104	043101	042524	
11160	045226	020122	052447	042523	
11161	045234	023522	042040	051511	
11162	045242	040520	041524	020110	
11163	045250	043117	054040	041506	
11164	045256	000			
11165	045257	127	052101	044103	EM33: .ASCIZ "WATCH-DOG TIMED OUT AFTER 'USER' DISPATCH OF XFC"
11166	045264	042055	043517	052040	
11167	045272	046511	042105	047440	
11168	045300	052125	040440	052106	
11169	045306	051105	023440	051525	
11170	045314	051105	020047	044504	
11171	045322	050123	052101	044103	
11172	045330	047440	020106	043130	
11173	045336	000103			
11174	045340	046106	043501	047040	EM34: .ASCIZ "FLAG NOT =1 AFTER ENTERING WCS VIA 'ODD PC' DISPATCH"
11175	045346	052117	036440	020061	
11176	045354	043101	042524	020122	
11177	045362	047105	042524	044522	
11178	045370	043516	053440	051503	
11179	045376	053040	040511	023440	
11180	045404	042117	020104	041520	
11181	045412	020047	044504	050123	
11182	045420	052101	044103	000	
11183	045425	127	052101	044103	EM35: .ASCIZ "WATCH-DOG TIMED OUT AFTER 'ODD PC' DISPATCH TO WCS"
11184	045432	042055	043517	052040	
11185	045440	046511	042105	047440	
11186	045446	052125	040440	052106	
11187	045454	051105	023440	042117	
11188	045462	020104	041520	020047	
11189	045470	044504	050123	052101	
11190	045476	044103	052040	020117	
11191	045504	041527	000123		
11192	045510	046106	043501	047040	EM36: .ASCIZ "FLAG NOT =1 AFTER ENTERING WCS VIA 'MICRO-BREAK' DISPATCH"
11193	045516	052117	036440	020061	
11194	045524	043101	042524	020122	
11195	045532	047105	042524	044522	
11196	045540	043516	053440	051503	
11197	045546	053040	040511	023440	
11198	045554	044515	051103	026517	
11199	045562	051102	040505	023513	
11200	045570	042040	051511	040520	
11201	045576	041524	000110		
11202	045602	040527	041524	026510	EM37: .ASCIZ "WATCH-DOG TIMED OUT AFTER 'MICRO-BREAK' DISPATCH TO WCS"
11203	045610	047504	020107	044524	
11204	045616	042515	020104	052517	
11205	045624	020124	043101	042524	
11206	045632	020122	046447	041511	
11207	045640	047522	041055	042522	
11208	045646	045501	020047	044504	



Line No	Module 1	Module 2	Module 3	Module 4	Module 5	Message
11209	045654	050123	052101	044103		
11210	045662	052040	020117	041527		
11211	045670	000123				
11212	045672	047503	051122	041505	EM40:	. ASCIZ "CORRECT DATA NOT STORED ON BLOCK MOVE OF GR'S TO WCS"
11213	045700	020124	040504	040524		
11214	045706	047040	052117	051440		
11215	045714	047524	042522	020104		
11216	045722	047117	041040	047514		
11217	045730	045503	046440	053117		
11218	045736	020105	043117	043440		
11219	045744	023522	020123	047524		
11220	045752	053440	051503	000		
11221	045757	127	052101	044103	EM41:	. ASCIZ "WATCH-DOG TIMED OUT ON STORING GR'S IN WCS"
11222	045764	042055	043517	052040		
11223	045772	046511	042105	047440		
11224	046000	052125	047440	020116		
11225	046006	052123	051117	047111		
11226	046014	020107	051107	051447		
11227	046022	044440	020116	041527		
11228	046030	000123				
11229	046032	047503	051122	041505	EM42:	. ASCIZ "CORRECT DATA NOT STORED ON LOAD OF GR'S FROM WCS"
11230	046040	020124	040504	040524		
11231	046046	047040	052117	051440		
11232	046054	047524	042522	020104		
11233	046062	047117	046040	040517		
11234	046070	020104	043117	043440		
11235	046076	023522	020123	051106		
11236	046104	046517	053440	051503		
11237	046112	000				
11238	046113	127	052101	044103	EM43:	. ASCIZ "WATCH-DOG TIMED OUT ON LOADING OF GR'S FROM WCS"
11239	046120	042055	043517	052040		
11240	046126	046511	042105	047440		
11241	046134	052125	047440	020116		
11242	046142	047514	042101	047111		
11243	046150	020107	043117	043440		
11244	046156	023522	020123	051106		
11245	046164	046517	053440	051503		
11246	046172	000				
11247	046173	103	051117	042522	EM44:	. ASCIZ "CORRECT RESERVED INST. CODE NOT SAVED AFTER 'RESVD INSTR' DISPATCH TO WCS"
11248	046200	052103	051040	051505		
11249	046206	051105	042526	020104		
11250	046214	047111	052123	020056		
11251	046222	047503	042504	047040		
11252	046230	052117	051440	053101		
11253	046236	042105	040440	052106		
11254	046244	051105	023440	042522		
11255	046252	053123	020104	047111		
11256	046260	052123	023522	042040		
11257	046266	051511	040520	041524		
11258	046274	020110	047524	053440		
11259	046302	051503	000			
11260	046305	127	052101	044103	EM45:	. ASCIZ "WATCH-DOG TIMED OUT AFTER 'RESVD INST' DISPATCH TO WCS"
11261	046312	042055	043517	052040		
11262	046320	046511	042105	047440		
11263	046326	052125	040440	052106		

11264	046334	051105	023440	042522	
11265	046342	053123	020104	047111	
11266	046350	052123	020047	044504	
11267	046356	050123	052101	044103	
11268	046364	052040	020117	041527	
11269	046372	000123			
11270	046374	047503	051122	041505	EM46: . ASCIZ "CORRECT DATA WAS NOT STORED ON BLOCK MOVE OF WFP REGISTERS TO WCS"
11271	046402	020124	040504	040524	
11272	046410	053440	051501	047040	
11273	046416	052117	051440	047524	
11274	046424	042522	020104	047117	
11275	046432	041040	047514	045503	
11276	046440	046440	053117	020105	
11277	046446	043117	053440	050106	
11278	046454	051040	043505	051511	
11279	046462	042524	051522	052040	
11280	046470	020117	041527	000123	
11281	046476	040527	041524	026510	EM47: . ASCIZ "WATCH-DOG TIMED OUT ON BLOCK MOVE OF WFP REGISTERS TO WCS"
11282	046504	047504	020107	044524	
11283	046512	042515	020104	052517	
11284	046520	020124	047117	041040	
11285	046526	047514	045503	046440	
11286	046534	053117	020105	043117	
11287	046542	053440	050106	051040	
11288	046550	043505	051511	042524	
11289	046556	051522	052040	020117	
11290	046564	041527	000123		
11291	046570	047503	051122	041505	EM50: . ASCIZ "CORRECT DATA WAS NOT STORED ON BLOCK MOVE TO WFP REGISTERS FROM WCS"
11292	046576	020124	040504	040524	
11293	046604	053440	051501	047040	
11294	046612	052117	051440	047524	
11295	046620	042522	020104	047117	
11296	046626	041040	047514	045503	
11297	046634	046440	053117	020105	
11298	046642	047524	053440	050106	
11299	046650	051040	043505	051511	
11300	046656	042524	051522	043040	
11301	046664	047522	020115	041527	
11302	046672	000123			
11303	046674	040527	041524	026510	EM51: . ASCIZ "WATCH-DOG TIMED OUT ON BLOCK MOVE FROM WCS TO CSP"
11304	046702	047504	020107	044524	
11305	046710	042515	020104	052517	
11306	046716	020124	047117	041040	
11307	046724	047514	045503	046440	
11308	046732	053117	020105	051106	
11309	046740	046517	053440	051503	
11310	046746	052040	020117	051503	
11311	046754	000120			
11312	046756	047503	051122	041505	EM52: . ASCIZ "CORRECT DATA WAS NOT STORED ON BLOCK MOVE OF CSP REGISTERS TO WCS"
11313	046764	020124	040504	040524	
11314	046772	053440	051501	047040	
11315	047000	052117	051440	047524	
11316	047006	042522	020104	047117	
11317	047014	041040	047514	045503	
11318	047022	046440	053117	020105	
11319	047030	043117	041440	050123	



11320	047036	051040	043505	051511	
11321	047044	042524	051522	052040	
11322	047052	020117	041527	000123	
11323	047060	040527	041524	026510	EM53: . ASCIZ "WATCH-DOG TIMED OUT ON BLOCK MOVE OF CSP REGISTERS TO WCS"
11324	047066	047504	020107	044524	
11325	047074	042515	020104	052517	
11326	047102	020124	047117	041040	
11327	047110	047514	045503	046440	
11328	047116	053117	020105	043117	
11329	047124	041440	050123	051040	
11330	047132	043505	051511	042524	
11331	047140	051522	052040	020117	
11332	047146	041527	000123		
11333	047152	047503	051122	041505	EM54: . ASCIZ "CORRECT DATA WAS NOT STORED ON BLOCK MOVE FROM WCS TO CSP"
11334	047160	020124	040504	040524	
11335	047166	053440	051501	047040	
11336	047174	052117	051440	047524	
11337	047202	042522	020104	047117	
11338	047210	041040	047514	045503	
11339	047216	046440	053117	020105	
11340	047224	051106	046517	053440	
11341	047232	051503	052040	020117	
11342	047240	051503	000120		
11343	047244	040527	041524	026510	EM55: . ASCIZ "WATCH-DOG TIMED OUT ON BLOCK MOVE OF CSP REGISTERS TO WCS"
11344	047252	047504	020107	044524	
11345	047260	042515	020104	052517	
11346	047266	020124	047117	041040	
11347	047274	047514	045503	046440	
11348	047302	053117	020105	043117	
11349	047310	041440	050123	051040	
11350	047316	043505	051511	042524	
11351	047324	051522	052040	020117	
11352	047332	041527	000123		
11353	047336	047503	051122	041505	EM56: . ASCIZ "CORRECT DATA WAS NOT STORED ON BLOCK MOVE OF WCS-USER-REG'S TO WCS"
11354	047344	020124	040504	040524	
11355	047352	053440	051501	047040	
11356	047360	052117	051440	047524	
11357	047366	042522	020104	047117	
11358	047374	041040	047514	045503	
11359	047402	046440	053117	020105	
11360	047410	043117	053440	051503	
11361	047416	052455	042523	026522	
11362	047424	042522	023507	020123	
11363	047432	047524	053440	051503	
11364	047440	000			
11365	047441	127	052101	044103	EM57: . ASCIZ "WATCH DOG TIMED OUT ON BLOCK MOVE OF WCS-USER-REG'S TO WCS"
11366	047446	042040	043517	052040	
11367	047454	046511	042105	047440	
11368	047462	052125	047440	020116	
11369	047470	046102	041517	020113	
11370	047476	047515	042526	047440	
11371	047504	020106	041527	026523	
11372	047512	051525	051105	051055	
11373	047520	043505	051447	052040	
11374	047526	020117	041527	000123	
11375	047534	047503	051122	041505	EM60: . ASCIZ "CORRECT DATA WAS NOT LOADED ON BLOCK MOVE FROM WCS TO WCS-USER-REG'S"

11376	047542	020124	040504	040524	
11377	047550	053440	051501	047040	
11378	047556	052117	046040	040517	
11379	047564	042504	020104	047117	
11380	047572	041040	047514	045503	
11381	047600	046440	053117	020105	
11382	047606	051106	046517	053440	
11383	047614	051503	052040	020117	
11384	047622	041527	026523	051525	
11385	047630	051105	051055	043505	
11386	047636	051447	000		
11387	047641	127	052101	044103	EM61: .ASCIZ "WATCH DOG TIMED OUT ON BLOCK MOVE FROM WCS TO WCS-USER-REG'S"
11388	047646	042040	043517	052040	
11389	047654	046511	042105	047440	
11390	047662	052125	047440	020116	
11391	047670	046102	041517	020113	
11392	047676	047515	042526	043040	
11393	047704	047522	020115	041527	
11394	047712	020123	047524	053440	
11395	047720	051503	052455	042523	
11396	047726	026522	042522	023507	
11397	047734	000123			
11398	047736	042117	026504	041520	EM62: .ASCIZ "ODD-PC TRAP INSTEAD OF ODD-PC DISPATCH TO WCS"
11399	047744	052040	040522	020120	
11400	047752	047111	052123	040505	
11401	047760	020104	043117	047440	
11402	047766	042104	050055	020103	
11403	047774	044504	050123	052101	
11404	050002	044103	052040	020117	
11405	050010	041527	000123		
11406	050014	042522	051123	042126	EM63: .ASCIZ "RESRVD. INSTR. TRAP INSTEAD OF RSRVD. INST. DISPATCH TO WCS"
11407	050022	020056	047111	052123	
11408	050030	027122	052040	040522	
11409	050036	020120	047111	052123	
11410	050044	040505	020104	043117	
11411	050052	051040	051123	042126	
11412	050060	020056	047111	052123	
11413	050066	020056	044504	050123	
11414	050074	052101	044103	052040	
11415	050102	020117	041527	000123	
11416	050110	041527	020123	040527	EM64: .ASCIZ "WCS WATCH DOG TIMER DID NOT TIME OUT"
11417	050116	041524	020110	047504	
11418	050124	020107	044524	042515	
11419	050132	020122	044504	020104	
11420	050140	047516	020124	044524	
11421	050146	042515	047440	052125	
11422	050154	000			
11423	050155	127	051503	053440	EM65: .ASCIZ "WCS WATCH DOG TIMER NOT WITHIN ACCEPTED RANGE"
11424	050162	052101	044103	042040	
11425	050170	043517	052040	046511	
11426	050176	051105	047040	052117	
11427	050204	053440	052111	044510	
11428	050212	020116	041501	042503	
11429	050220	052120	042105	051040	
11430	050226	047101	042507	000	
11431	050233	130	041506	047440	EM66: .ASCIZ "XFC OPCODE DID NOT TRAP AS RESRVD. INSTR. WHEN WCS WAS DISABLED"



11432	050240	041520	042117	020105	
11433	050246	044504	020104	047516	
11434	050254	020124	051124	050101	
11435	050262	040440	020123	042522	
11436	050270	053123	027104	044440	
11437	050276	051516	051124	020056	
11438	050304	044127	047105	053440	
11439	050312	051503	053440	051501	
11440	050320	042040	051511	041101	
11441	050326	042514	000104		
11442	050332	041527	026523	047111	EM67: .ASCIZ "WCS-INIT DID NOT CLEAR BAD PARITY IN WCS LOC."
11443	050340	052111	042040	042111	
11444	050346	047040	052117	041440	
11445	050354	042514	051101	041040	
11446	050362	042101	050040	051101	
11447	050370	052111	020131	047111	
11448	050376	053440	051503	046040	
11449	050404	041517	000056		
11450	050410	044515	051103	026517	EM70: .ASCIZ "MICRO-WORD LOADED WRONGLY IN WCS AT LOC"
11451	050416	047527	042122	046040	
11452	050424	040517	042504	020104	
11453	050432	051127	047117	046107	
11454	050440	020131	047111	053440	
11455	050446	051503	040440	020124	
11456	050454	047514	000103		
11457	050460	051127	047117	020107	EM71: .ASCIZ "WRONG PARITY BIT-PAR0-(BIT 1 IN STATUS) READ FROM WCS LOC"
11458	050466	040520	044522	054524	
11459	050474	041040	052111	050055	
11460	050502	051101	026460	041050	
11461	050510	052111	030440	044440	
11462	050516	020116	052123	052101	
11463	050524	051525	020051	042522	
11464	050532	042101	043040	047522	
11465	050540	020115	041527	020123	
11466	050546	047514	000103		
11467	050552	051127	047117	020107	EM72: .ASCIZ "WRONG PARITY BIT-PAR1-(BIT 2 IN STATUS) READ FROM WCS LOC"
11468	050560	040520	044522	054524	
11469	050566	041040	052111	050055	
11470	050574	051101	026461	041050	
11471	050602	052111	031040	044440	
11472	050610	020116	052123	052101	
11473	050616	051525	020051	042522	
11474	050624	042101	043040	047522	
11475	050632	020115	041527	020123	
11476	050640	047514	000103		
11477	050644	051127	047117	020107	EM73: .ASCIZ "WRONG PARITY BIT-PAR2-(BIT 3 IN STATUS) READ FROM WCS LOC"
11478	050652	040520	044522	054524	
11479	050660	041040	052111	050055	
11480	050666	051101	026462	041050	
11481	050674	052111	031440	044440	
11482	050702	020116	052123	052101	
11483	050710	051525	020051	042522	
11484	050716	042101	043040	047522	
11485	050724	020115	041527	020123	
11486	050732	047514	000103		
11487	050736	046106	043501	047040	EM74: .ASCIZ "FLAG NOT =2 AFTER ENTERING WCS VIA 'FLAG<7>-SVC' DISPATCH"

11488	050744	052117	036440	020062	
11489	050752	043101	042524	020122	
11490	050760	047105	042524	044522	
11491	050766	043516	053440	051503	
11492	050774	053040	040511	023440	
11493	051002	046106	043501	033474	
11494	051010	026476	053123	023503	
11495	051016	042040	051511	040520	
11496	051024	041524	000110		
11497	051030	040527	041524	026510	EM75: .ASCIZ "WATCH-DOG TIMED OUT AFTER 'FLAG<7>-SVC' DISPATCH TO WCS"
11498	051036	047504	020107	044524	
11499	051044	042515	020104	052517	
11500	051052	020124	043101	042524	
11501	051060	020122	043047	040514	
11502	051066	036107	037067	051455	
11503	051074	041526	020047	044504	
11504	051102	050123	052101	044103	
11505	051110	052040	020117	041527	
11506	051116	000123			
11507	051120	047111	047503	051122	EM76: .ASCIZ "INCORRECT DATA STORED DURING 'LOAD WRITE 2' TMS FUNCTION"
11508	051126	041505	020124	040504	
11509	051134	040524	051440	047524	
11510	051142	042522	020104	052504	
11511	051150	044522	043516	023440	
11512	051156	047514	042101	053440	
11513	051164	044522	042524	031040	
11514	051172	020047	046524	020123	
11515	051200	052506	041516	044524	
11516	051206	047117	000		
11517	051211	127	052101	044103	EM77: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD WRITE 2' TMS FUNCTION"
11518	051216	042055	043517	052040	
11519	051224	046511	047505	052125	
11520	051232	042040	051125	047111	
11521	051240	020107	046047	040517	
11522	051246	020104	051127	052111	
11523	051254	020105	023462	052040	
11524	051262	051515	043040	047125	
11525	051270	052103	047511	000116	
11526	051276	047111	047503	051122	EM100: .ASCIZ "INCORRECT DATA STORED DURING 'LOAD READ 2' TMS FUNCTION"
11527	051304	041505	020124	040504	
11528	051312	040524	051440	047524	
11529	051320	042522	020104	052504	
11530	051326	044522	043516	023440	
11531	051334	047514	042101	051040	
11532	051342	040505	020104	023462	
11533	051350	052040	051515	043040	
11534	051356	047125	052103	047511	
11535	051364	000116			
11536	051366	040527	041524	026510	EM101: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD READ 2' TMS FUNCTION"
11537	051374	047504	020107	044524	
11538	051402	042515	052517	020124	
11539	051410	052504	044522	043516	
11540	051416	023440	047514	042101	
11541	051424	051040	040505	020104	
11542	051432	023462	052040	051515	
11543	051440	043040	047125	052103	



11544	051446	047511	000116		
11545	051452	047111	047503	051122	EM102: .ASCIZ "INCORRECT DATA STORED DURING 'SET STORE / WRITE 2' TMS FUNCTION"
11546	051460	041505	020124	040504	
11547	051466	040524	051440	047524	
11548	051474	042522	020104	052504	
11549	051502	044522	043516	023440	
11550	051510	042523	020124	052123	
11551	051516	051117	020105	020057	
11552	051524	051127	052111	020105	
11553	051532	023462	052040	051515	
11554	051540	043040	047125	052103	
11555	051546	047511	000116		
11556	051552	040527	041524	026510	EM103: .ASCIZ "WATCH-DOG TIMEOUT DURING 'SET STORE / WRITE 2' TMS FUNCTION"
11557	051560	047504	020107	044524	
11558	051566	042515	052517	020124	
11559	051574	052504	044522	043516	
11560	051602	023440	042523	020124	
11561	051610	052123	051117	020105	
11562	051616	020057	051127	052111	
11563	051624	020105	023462	052040	
11564	051632	051515	043040	047125	
11565	051640	052103	047511	000116	
11566	051646	047111	047503	051122	EM104: .ASCIZ "INCORRECT DATA STORED DURING 'SET LOAD / INC READ 2' TMS FUNCTION"
11567	051654	041505	020124	040504	
11568	051662	040524	051440	047524	
11569	051670	042522	020104	052504	
11570	051676	044522	043516	023440	
11571	051704	042523	020124	047514	
11572	051712	042101	027440	044440	
11573	051720	041516	051040	040505	
11574	051726	020104	023462	052040	
11575	051734	051515	043040	047125	
11576	051742	052103	047511	000116	
11577	051750	040527	041524	026510	EM105: .ASCIZ "WATCH-DOG TIMEOUT DURING 'SET LOAD / INC READ 2' TMS FUNCTION"
11578	051756	047504	020107	044524	
11579	051764	042515	052517	020124	
11580	051772	052504	044522	043516	
11581	052000	023440	042523	020124	
11582	052006	047514	042101	027440	
11583	052014	044440	041516	051040	
11584	052022	040505	020104	023462	
11585	052030	052040	051515	043040	
11586	052036	047125	052103	047511	
11587	052044	000116			
11588	052046	047111	047503	051122	EM106: .ASCIZ "INCORRECT DATA STORED DURING 'WRITE INDIRECT' TMS FUNCTION"
11589	052054	041505	020124	040504	
11590	052062	040524	051440	047524	
11591	052070	042522	020104	052504	
11592	052076	044522	043516	023440	
11593	052104	051127	052111	020105	
11594	052112	047111	044504	042522	
11595	052120	052103	020047	046524	
11596	052126	020123	052506	041516	
11597	052134	044524	047117	000	
11598	052141	127	052101	044103	EM107: .ASCIZ "WATCH-DOG TIMEOUT DURING 'WRITE-INDIRECT' TMS FUNCTION"
11599	052146	042055	043517	052040	

11600	052154	046511	047505	052125	
11601	052162	042040	051125	047111	
11602	052170	020107	053447	044522	
11603	052176	042524	044455	042116	
11604	052204	051111	041505	023524	
11605	052212	052040	051515	043040	
11606	052220	047125	052103	047511	
11607	052226	000116			
11608	052230	047111	047503	051122	EM110: .ASCIZ "INCORRECT DATA STORED DURING 'READ-INDIRECT' TMS FUNCTION"
11609	052236	041505	020124	040504	
11610	052244	040524	051440	047524	
11611	052252	042522	020104	052504	
11612	052260	044522	043516	023440	
11613	052266	042522	042101	044455	
11614	052274	042116	051111	041505	
11615	052302	023524	052040	051515	
11616	052310	043040	047125	052103	
11617	052316	047511	000116		
11618	052322	040527	041524	026510	EM111: .ASCIZ "WATCH-DOG TIMEOUT DURING 'READ-INDIRECT' TMS FUNCTION"
11619	052330	047504	020107	044524	
11620	052336	042515	052517	020124	
11621	052344	052504	044522	043516	
11622	052352	023440	042522	042101	
11623	052360	044455	042116	051111	
11624	052366	041505	023524	052040	
11625	052374	051515	043040	047125	
11626	052402	052103	047511	000116	
11627	052410	047111	047503	051122	EM112: .ASCIZ "INCORRECT DATA STORED DURING 'STORE WFP<0:1>' TMS FUNCTION"
11628	052416	041505	020124	040504	
11629	052424	040524	051440	047524	
11630	052432	042522	020104	052504	
11631	052440	044522	043516	023440	
11632	052446	052123	051117	020105	
11633	052454	043127	036120	035060	
11634	052462	037061	020047	046524	
11635	052470	020123	052506	041516	
11636	052476	044524	047117	000	
11637	052503	127	052101	044103	EM113: .ASCIZ "WATCH-DOG TIMEOUT DURING 'STORE WFP<0:1>' TMS FUNCTION"
11638	052510	042055	043517	052040	
11639	052516	046511	047505	052125	
11640	052524	042040	051125	047111	
11641	052532	020107	051447	047524	
11642	052540	042522	053440	050106	
11643	052546	030074	030472	023476	
11644	052554	052040	051515	043040	
11645	052562	047125	052103	047511	
11646	052570	000116			
11647	052572	047111	047503	051122	EM114: .ASCIZ "INCORRECT DATA STORED DURING 'STORE WFP<2:3>' TMS FUNCTION"
11648	052600	041505	020124	040504	
11649	052606	040524	051440	047524	
11650	052614	042522	020104	052504	
11651	052622	044522	043516	023440	
11652	052630	052123	051117	020105	
11653	052636	043127	036120	035062	
11654	052644	037063	020047	046524	
11655	052652	020123	052506	041516	



11656	052660	044524	047117	000	
11657	052665	127	052101	044103	EM115: .ASCIZ "WATCH-DOG TIMEOUT DURING 'STORE WFP<2: 3>' TMS FUNCTION"
11658	052672	042055	043517	052040	
11659	052700	046511	047505	052125	
11660	052706	042040	051125	047111	
11661	052714	020107	051447	047524	
11662	052722	042522	053440	050106	
11663	052730	031074	031472	023476	
11664	052736	052040	051515	043040	
11665	052744	047125	052103	047511	
11666	052752	000116			
11667	052754	047111	047503	051122	EM116: .ASCIZ "INCORRECT DATA STORED DURING 'STORE WFP<4: 5>' TMS FUNCTION"
11668	052762	041505	020124	040504	
11669	052770	040524	051440	047524	
11670	052776	042522	020104	052504	
11671	053004	044522	043516	023440	
11672	053012	052123	051117	020105	
11673	053020	043127	036120	035064	
11674	053026	037065	020047	046524	
11675	053034	020123	052506	041516	
11676	053042	044524	047117	000	
11677	053047	127	052101	044103	EM117: .ASCIZ "WATCH-DOG TIMEOUT DURING 'STORE WFP<4: 5>' TMS FUNCTION"
11678	053054	042055	043517	052040	
11679	053062	046511	047505	052125	
11680	053070	042040	051125	047111	
11681	053076	020107	051447	047524	
11682	053104	042522	053440	050106	
11683	053112	032074	032472	023476	
11684	053120	052040	051515	043040	
11685	053126	047125	052103	047511	
11686	053134	000116			
11687	053136	047111	047503	051122	EM120: .ASCIZ "INCORRECT DATA STORED DURING 'LOAD WFP<0: 1>' TMS FUNCTION"
11688	053144	041505	020124	040504	
11689	053152	040524	051440	047524	
11690	053160	042522	020104	052504	
11691	053166	044522	043516	023440	
11692	053174	047514	042101	053440	
11693	053202	050106	030074	030472	
11694	053210	023476	052040	051515	
11695	053216	043040	047125	052103	
11696	053224	047511	000116		
11697	053230	040527	041524	026510	EM121: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD WFP<0: 1>' TMS FUNCTION"
11698	053236	047504	020107	044524	
11699	053244	042515	052517	020124	
11700	053252	052504	044522	043516	
11701	053260	023440	047514	042101	
11702	053266	053440	050106	030074	
11703	053274	030472	023476	052040	
11704	053302	051515	043040	047125	
11705	053310	052103	047511	000116	
11706	053316	047111	047503	051122	EM122: .ASCIZ "INCORRECT DATA STORED DURING 'LOAD WFP<2: 3>' TMS FUNCTION"
11707	053324	041505	020124	040504	
11708	053332	040524	051440	047524	
11709	053340	042522	020104	052504	
11710	053346	044522	043516	023440	
11711	053354	047514	042101	053440	

11712	053362	050106	031074	031472	
11713	053370	023476	052040	051515	
11714	053376	043040	047125	052103	
11715	053404	047511	000116		
11716	053410	040527	041524	026510	EM123: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD WFP<2: 3>' TMS FUNCTION"
11717	053416	047504	020107	044524	
11718	053424	042515	052517	020124	
11719	053432	052504	044522	043516	
11720	053440	023440	047514	042101	
11721	053446	053440	050106	031074	
11722	053454	031472	023476	052040	
11723	053462	051515	043040	047125	
11724	053470	052103	047511	000116	
11725	053476	047111	047503	051122	EM124: .ASCIZ "INCORRECT DATA STORED DURING 'LOAD WFP<4: 5>' TMS FUNCTION"
11726	053504	041505	020124	040504	
11727	053512	040524	051440	047524	
11728	053520	042522	020104	052504	
11729	053526	044522	043516	023440	
11730	053534	047514	042101	053440	
11731	053542	050106	032074	032472	
11732	053550	023476	052040	051515	
11733	053556	043040	047125	052103	
11734	053564	047511	000116		
11735	053570	040527	041524	026510	EM125: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD WFP<4: 5>' TMS FUNCTION"
11736	053576	047504	020107	044524	
11737	053604	042515	052517	020124	
11738	053612	052504	044522	043516	
11739	053620	023440	047514	042101	
11740	053626	053440	050106	032074	
11741	053634	032472	023476	052040	
11742	053642	051515	043040	047125	
11743	053650	052103	047511	000116	
11744	053656	047111	047503	051122	EM126: .ASCIZ "INCORRECT DATA STORED DURING 'STORE ASPAD 00: 37 ' TMS FUNCTION"
11745	053664	041505	020124	040504	
11746	053672	040524	051440	047524	
11747	053700	042522	020104	052504	
11748	053706	044522	043516	023440	
11749	053714	052123	051117	020105	
11750	053722	051501	040520	055504	
11751	053730	030060	031472	056467	
11752	053736	020047	046524	020123	
11753	053744	052506	041516	044524	
11754	053752	047117	000		
11755	053755	127	052101	044103	EM127: .ASCIZ "WATCH-DOG TIMEOUT DURING 'STORE ASPAD 00: 37 ' TMS FUNCTION"
11756	053762	042055	043517	052040	
11757	053770	046511	047505	052125	
11758	053776	042040	051125	047111	
11759	054004	020107	051447	047524	
11760	054012	042522	040440	050123	
11761	054020	042101	030133	035060	
11762	054026	033463	023535	052040	
11763	054034	051515	043040	047125	
11764	054042	052103	047511	000116	
11765	054050	047111	047503	051122	EM130: .ASCIZ "INCORRECT DATA STORED DURING 'STORE BSPAD 00: 37 ' TMS FUNCTION"
11766	054056	041505	020124	040504	
11767	054064	040524	051440	047524	



11768	054072	042522	020104	052504	
11769	054100	044522	043516	023440	
11770	054106	052123	051117	020105	
11771	054114	051502	040520	055504	
11772	054122	030060	031472	056467	
11773	054130	020047	046524	020123	
11774	054136	052506	041516	044524	
11775	054144	047117	000		
11776	054147	127	052101	044103	EM131: .ASCIZ "WATCH-DOG TIMEOUT DURING 'STORE BSPAD 00:37' TMS FUNCTION"
11777	054154	042055	043517	052040	
11778	054162	046511	047505	052125	
11779	054170	042040	051125	047111	
11780	054176	020107	051447	047524	
11781	054204	042522	041040	050123	
11782	054212	042101	030133	035060	
11783	054220	033463	023535	052040	
11784	054226	051515	043040	047125	
11785	054234	052103	047511	000116	
11786	054242	047111	047503	051122	EM132: .ASCIZ "INCORRECT DATA STORED DURING 'STORE CSPAD 17:00' TMS FUNCTION"
11787	054250	041505	020124	040504	
11788	054256	040524	051440	047524	
11789	054264	042522	020104	052504	
11790	054272	044522	043516	023440	
11791	054300	052123	051117	020105	
11792	054306	051503	040520	055504	
11793	054314	033461	030072	056460	
11794	054322	020047	046524	020123	
11795	054330	052506	041516	044524	
11796	054336	047117	000		
11797	054341	127	052101	044103	EM133: .ASCIZ "WATCH-DOG TIMEOUT DURING 'STORE CSPAD 17:00' TMS FUNCTION"
11798	054346	042055	043517	052040	
11799	054354	046511	047505	052125	
11800	054362	042040	051125	047111	
11801	054370	020107	051447	047524	
11802	054376	042522	041440	050123	
11803	054404	042101	030533	035067	
11804	054412	030060	023535	052040	
11805	054420	051515	043040	047125	
11806	054426	052103	047511	000116	
11807	054434	047111	047503	051122	EM134: .ASCIZ "INCORRECT DATA LOADED DURING 'LOAD ASPAD 00:37' TMS FUNCTION"
11808	054442	041505	020124	040504	
11809	054450	040524	046040	040517	
11810	054456	042504	020104	052504	
11811	054464	044522	043516	023440	
11812	054472	047514	042101	040440	
11813	054500	050123	042101	030133	
11814	054506	035060	033463	023535	
11815	054514	052040	051515	043040	
11816	054522	047125	052103	047511	
11817	054530	000116			
11818	054532	040527	041524	026510	EM135: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD ASPAD 00:37' TMS FUNCTION"
11819	054540	047504	020107	044524	
11820	054546	042515	052517	020124	
11821	054554	052504	044522	043516	
11822	054562	023440	047514	042101	
11823	054570	040440	050123	042101	

11824	054576	030133	035060	033463	
11825	054604	023535	052040	051515	
11826	054612	043040	047125	052103	
11827	054620	047511	000116		
11828	054624	047111	047503	051122	EM136: .ASCIZ "INCORRECT DATA LOADED DURING 'LOAD BSPAD 00: 37 ' TMS FUNCTION"
11829	054632	041505	020124	040504	
11830	054640	040524	046040	040517	
11831	054646	042504	020104	052504	
11832	054654	044522	043516	023440	
11833	054662	047514	042101	041040	
11834	054670	050123	042101	030133	
11835	054676	035060	033463	023535	
11836	054704	052040	051515	043040	
11837	054712	047125	052103	047511	
11838	054720	000116			
11839	054722	040527	041524	026510	EM137: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD BSPAD 00: 37 ' TMS FUNCTION"
11840	054730	047504	020107	044524	
11841	054736	042515	052517	020124	
11842	054744	052504	044522	043516	
11843	054752	023440	047514	042101	
11844	054760	041040	050123	042101	
11845	054766	030133	035060	033463	
11846	054774	023535	052040	051515	
11847	055002	043040	047125	052103	
11848	055010	047511	000116		
11849	055014	047111	047503	051122	EM140: .ASCIZ "INCORRECT DATA LOADED DURING 'LOAD CSPAD 17: 00 ' TMS FUNCTION"
11850	055022	041505	020124	040504	
11851	055030	040524	046040	040517	
11852	055036	042504	020104	052504	
11853	055044	044522	043516	023440	
11854	055052	047514	042101	041440	
11855	055060	050123	042101	030533	
11856	055066	035067	030060	023535	
11857	055074	052040	051515	043040	
11858	055102	047125	052103	047511	
11859	055110	000116			
11860	055112	040527	041524	026510	EM141: .ASCIZ "WATCH-DOG TIMEOUT DURING 'LOAD CSPAD 17: 00 ' TMS FUNCTION"
11861	055120	047504	020107	044524	
11862	055126	042515	052517	020124	
11863	055134	052504	044522	043516	
11864	055142	023440	047514	042101	
11865	055150	041440	050123	042101	
11866	055156	030533	035067	030060	
11867	055164	023535	052040	051515	
11868	055172	043040	047125	052103	
11869	055200	047511	000116		
11870	055204	047111	047503	051122	EM142: .ASCIZ "INCORRECT DATA LOADED DURING 'LOAD TEMP-S' TMS FUNCTION"
11871	055212	041505	020124	040504	
11872	055220	040524	046040	040517	
11873	055226	042504	020104	052504	
11874	055234	044522	043516	023440	
11875	055242	047514	042101	052040	
11876	055250	046505	026520	023523	
11877	055256	052040	051515	043040	
11878	055264	047125	052103	047511	
11879	055272	000116			



11880	055274	040527	041524	026510	EM143:	.ASCIZ	"WATCH-DOG TIMEOUT DURING 'LOAD TEMP-S' TMS FUNCTION"
11881	055302	047504	020107	044524			
11882	055310	042515	052517	020124			
11883	055316	052504	044522	043516			
11884	055324	023440	047514	042101			
11885	055332	052040	046505	026520			
11886	055340	023523	052040	051515			
11887	055346	043040	047125	052103			
11888	055354	047511	000116				
11889	055360	047111	047503	051122	EM144:	.ASCIZ	"INCORRECT DATA STORED DURING 'STORE TEMP-S' TMS FUNCTION"
11890	055366	041505	020124	040504			
11891	055374	040524	051440	047524			
11892	055402	042522	020104	052504			
11893	055410	044522	043516	023440			
11894	055416	052123	051117	020105			
11895	055424	042524	050115	051455			
11896	055432	020047	046524	020123			
11897	055440	052506	041516	044524			
11898	055446	047117	000				
11899	055451	127	052101	044103	EM145:	.ASCIZ	"WATCH-DOG TIMEOUT DURING 'STORE TEMP-S' TMS FUNCTION"
11900	055456	042055	043517	052040			
11901	055464	046511	047505	052125			
11902	055472	042040	051125	047111			
11903	055500	020107	051447	047524			
11904	055506	042522	052040	046505			
11905	055514	026520	023523	052040			
11906	055522	051515	043040	047125			
11907	055530	052103	047511	000116			
11908							
11909							
11910						.SBTTL	ERROR DATA HEADERS
11911							
11912	055536	042444	051122	041520	DH1:	.ASCIZ	"\$ERRPC REG-ADRS"
11913	055544	020040	042522	026507			
11914	055552	042101	051522	000			
11915	055557	044	051105	050122	DH2:	.ASCIZ	"\$ERRPC WHAMI"
11916	055564	020103	053440	040510			
11917	055572	044515	000				
11918	055575	044	051105	050122	DH3:	.ASCIZ	"\$ERRPC STATUS"
11919	055602	020103	051440	040524			
11920	055610	052524	000123				
11921	055614	042444	051122	041520	DH11:	.ASCIZ	"\$ERRPC E-DATA R-DATA WCSLOC PATTRN"
11922	055622	020040	026505	040504			
11923	055630	040524	020040	026522			
11924	055636	040504	040524	020040			
11925	055644	041527	046123	041517			
11926	055652	020040	040520	052124			
11927	055660	047122	000				
11928	055663	044	051105	050122	DH12:	.ASCIZ	"\$ERRPC WCSLOC E-DATA R-DATA"
11929	055670	020103	053440	051503			
11930	055676	047514	020103	042440			
11931	055704	042055	052101	020101			
11932	055712	051040	042055	052101			
11933	055720	000101					
11934	055722	042444	051122	041520	DH14:	.ASCIZ	"\$ERRPC TRPLOC"
11935	055730	020040	051124	046120			

Line	Address	Offset	Value	Label	Description
11936	055736	041517	000		
11937	055741	044	051105	DH15:	. ASCIZ "\$ERRPC PATTRN E-DATA . PAR. ."
11938	055746	020103	050040		
11939	055754	051124	020116		
11940	055762	042055	052101		
11941	055770	027040	040520		
11942	055776	000056			
11943	056000	042444	051122	DH17:	. ASCIZ "\$ERRPC STATUS PATTRN"
11944	056006	020040	052123		
11945	056014	051525	020040		
11946	056022	052124	047122		
11947	056027	044	051105	DH21:	. ASCIZ "\$ERRPC WCSLOC PATTRN STATUS"
11948	056034	020103	053440		
11949	056042	047514	020103		
11950	056050	052101	051124		
11951	056056	051440	040524		
11952	056064	000123			
11953	056066	042444	051122	DH23:	. ASCIZ "\$ERRPC"
11954	056074	000			
11955	056075	044	051105	DH26:	. ASCIZ "\$ERRPC WCSLOC E-DATA STATUS"
11956	056102	020103	053440		
11957	056110	047514	020103		
11958	056116	042055	052101		
11959	056124	051440	040524		
11960	056132	000123			
11961	056134	042444	051122	DH30:	. ASCIZ "\$ERRPC E-DATA R-DATA (XFC-CODES)"
11962	056142	020040	026505		
11963	056150	040524	020040		
11964	056156	040504	040524		
11965	056164	054050	041506		
11966	056172	042117	051505		
11967	056200	042444	051122	DH31:	. ASCIZ "\$ERRPC XFC-CODE"
11968	056206	020040	043130		
11969	056214	047503	042504		
11970	056221	044	051105	DH34:	. ASCIZ "\$ERRPC R3-FLAG"
11971	056226	020103	051040		
11972	056234	046106	043501		
11973	056241	044	051105	DH40:	. ASCIZ "\$ERRPC WCSLOC E-DATA R-DATA GPR. ##"
11974	056246	020103	053440		
11975	056254	047514	020103		
11976	056262	042055	052101		
11977	056270	051040	042055		
11978	056276	020101	043440		
11979	056304	021456	000043		
11980	056310	042444	051122	DH42:	. ASCIZ "\$ERRPC GPR. ## E-DATA R-DATA WCSLOC"
11981	056316	020040	050107		
11982	056324	021443	020040		
11983	056332	040504	040524		
11984	056340	026522	040504		
11985	056346	020040	041527		
11986	056354	041517	000		
11987	056357	044	051105	DH44:	. ASCIZ "\$ERRPC E-DATA R-DATA"
11988	056364	020103	042440		
11989	056372	052101	020101		
11990	056400	042055	052101		
11991	056406	042444	051122	DH46:	. ASCIZ "\$ERRPC WCSLOC E-DATA R-DATA WFP###"



11992	056414	020040	041527	046123						
11993	056422	041517	020040	026505						
11994	056430	040504	040524	020040						
11995	056436	026522	040504	040524						
11996	056444	020040	043127	021520						
11997	056452	021443	000							
11998	056455	044	051105	050122	DH50:	. ASC IZ	"\$ERRPC	WFP###	E-DATA	R-DATA WCSLOC"
11999	056462	020103	053440	050106						
12000	056470	021443	020043	042440						
12001	056476	042055	052101	020101						
12002	056504	051040	042055	052101						
12003	056512	020101	053440	051503						
12004	056520	047514	000103							
12005	056524	042444	051122	041520	DH52:	. ASC IZ	"\$ERRPC	WCSLOC	E-DATA	R-DATA CSP###"
12006	056532	020040	041527	046123						
12007	056540	041517	020040	026505						
12008	056546	040504	040524	020040						
12009	056554	026522	040504	040524						
12010	056562	020040	051503	021520						
12011	056570	021443	000							
12012	056573	044	051105	050122	DH54:	. ASC IZ	"\$ERRPC	CSP###	E-DATA	R-DATA WCSLOC"
12013	056600	020103	041440	050123						
12014	056606	021443	020043	042440						
12015	056614	042055	052101	020101						
12016	056622	051040	042055	052101						
12017	056630	020101	053440	051503						
12018	056636	047514	000103							
12019	056642	042444	051122	041520	DH56:	. ASC IZ	"\$ERRPC	WCSLOC	E-DATA	R-DATA MED"
12020	056650	020040	041527	046123						
12021	056656	041517	020040	026505						
12022	056664	040504	040524	020040						
12023	056672	026522	040504	040524						
12024	056700	020040	046440	042105						
12025	056706	000								
12026	056707	044	051105	050122	DH62:	. ASC IZ	"\$ERRPC	ODD-PC-AT"		
12027	056714	020103	047440	042104						
12028	056722	050055	026503	052101						
12029	056730	000								
12030	056731	044	051105	050122	DH65:	. ASC IZ	"\$ERRPC	TIME"		
12031	056736	020103	052040	046511						
12032	056744	000105								
12033	056746	042444	051122	041520	DH67:	. ASC IZ	"\$ERRPC	STATUS	WCSLOC"	
12034	056754	020040	052123	052101						
12035	056762	051525	020040	041527						
12036	056770	046123	041517	000						
12037	056775	044	051105	050122	DH71:	. ASC IZ	"\$ERRPC	WCSLOC	STATUS"	
12038	057002	020103	053440	051503						
12039	057010	047514	020103	051440						
12040	057016	040524	052524	000123						
12041	057024	042444	051122	041520	DH76:	. ASC IZ	"\$ERRPC	GPR.##	E-DATA	R-DATA WCSLOC"
12042	057032	020040	050107	027122						
12043	057040	021443	020040	026505						
12044	057046	040504	040524	020040						
12045	057054	026522	040504	040524						
12046	057062	020040	041527	046123						
12047	057070	041517	000							

```
12048 057073 044 051105 050122 DH100: .ASCIZ "$ERRPC GPR.## E-DATA R-DATA"
12049 057100 020103 043440 051120
12050 057106 021456 020043 042440
12051 057114 042055 052101 020101
12052 057122 051040 042055 052101
12053 057130 000101
12054 057132 042444 051122 041520 DH106: .ASCIZ "$ERRPC ADDR. 1 DATA. 1 ADDR. 2 E-DATA R-DATA"
12055 057140 020040 042101 051104
12056 057146 030456 020040 040504
12057 057154 040524 030456 020040
12058 057162 042101 051104 031056
12059 057170 020040 026505 040504
12060 057176 040524 020040 026522
12061 057204 040504 040524 000
12062 057211 044 051105 050122 DH120: .ASCIZ "$ERRPC MED E-DATA R-DATA"
12063 057216 020103 020040 042515
12064 057224 020104 020040 042440
12065 057232 042055 052101 020101
12066 057240 051040 042055 052101
12067 057246 000101
12068 057250
12069 057250 042444 051122 041520 DH112: DH126: DH130:
DH132: .ASCIZ "$ERRPC MED WCSLOC E-DATA R-DATA"
12070 057256 020040 046440 042105
12071 057264 020040 020040 041527
12072 057272 046123 041517 020040
12073 057300 026505 040504 040524
12074 057306 020040 026522 040504
12075 057314 040524 000
12076
12077 .SBTTL ERROR DATA VECTORS
12078
12079 .EVEN
12080 057320 001116 001162 000000 DT1: .WORD $ERRPC, $REG0, 0
12081 057326 DT76: DT40: DT112:
12082 057326 001116 001162 001164 DT11: .WORD $ERRPC, $REG0, $REG1, $REG2, $REG3, 0
12083 057334 001166 001170 000000
12084 057342 DT100: DT120:
12085 057342 001116 001162 001164 DT12: .WORD $ERRPC, $REG0, $REG1, $REG2, 0
12086 057350 001166 000000
12087 057354 001116 001162 001164 DT15: .WORD $ERRPC, $REG0, $REG1, 0
12088 057362 000000
12089 057364 001116 000000 DT23: .WORD $ERRPC, 0
12090 057370 001116 001162 001164 DT106: .WORD $ERRPC, $REG0, $REG1, $REG2, $REG3, $REG4, 0
12091 057376 001166 001170 001172
12092 057404 000000
12093 ;END OF MACRO CODE
```



PDP-11/60 WCS DIAGNOSTIC  
DQKURR P11 25-OCT-77 20:03

MACY11 30(1046) 25-OCT-77 20:12<sup>D 3</sup> PAGE 223  
MICRO-CODE

SEQ 0225  
SEQ 0236

12094  
12095

.SBTTL MICRO-CODE

```
12096 ;  
12097 ; ; START OF MICROCODE  
12098 ;  
12099 ; ; *** NOTE ***  
12100 ;  
12101 ; ; ALL MICROWORDS FOR A PARTICULAR TEST MUST (!!!) BE IN CONSECUTIVE  
12102 ; ; MICROADDRESS LOCATIONS (IE, 6026,6027,6030,...).  
12103 ; ;  
12104 ; ; MICROADDRESSES:  
12105 ; ; (0XXX) MAP TO ADDRESSES (6XXX)  
12106 ; ; (1XXX) " " " (7XXX)  
12107 ; ;  
12108 ; ; (3XXX) MAP TO ENTRY POINT LOCATION  
12109 ; ;  
12110 ; ; ***  
12111 ; ;  
12112 ; ;  
12113 ; ;  
12114 ; ; *****  
12115 ; ; LOAD; READ-2 FROM LOCAL STORE TMS FUNCTION  
12116 ; ; *****  
12117 ; ;  
12118 057406 MLDRD2:  
12119 ; ; 6002:  
12120 ; ; UDISP:  
12121 ; ; D_R2, ; LOCAL STORE ADDRESS  
12122 ; ; J/570  
12123 ; ;  
12124 057406 030570 010000 170212 . WORD UWRD1, UWRD2, UWRD3  
12125 ; ;  
12126 ; ;  
12127 ; ; 6570:  
12128 ; ; ENTER. TMS, TMSPTR_LOAD. READ. 2(552). ; ENTER INTO TMS ROUTINE  
12129 ; ; J/571  
12130 ; ;  
12131 057414 030571 100660 022414 . WORD UWRD1, UWRD2, UWRD3  
12132 ; ;  
12133 ; ;  
12134 ; ; 6571: ; NULL WORD #1  
12135 ; ; RETURN_BRA05, ; WILL EXIT WCS THRU HERE  
12136 ; ; GOTO-PAGE(6),  
12137 ; ; J/572  
12138 ; ;  
12139 057422 034572 100000 000007 . WORD UWRD1, UWRD2, UWRD3  
12140 ; ;  
12141 ; ;  
12142 ; ; 6572: ; NULL WORD #2  
12143 ; ; MD_DATA, ; MUST DO TO SUPPORT TMS  
12144 ; ; J/573  
12145 ; ;  
12146 057430 030573 000010 007000 . WORD UWRD1, UWRD2, UWRD3  
12147 ; ;  
12148 ; ;  
12149 ; ; 6573:  
12150 ; ; R3_D, ; GET FIRST DATA ITEM INTO R3  
12151 ; ; J/574
```



```
12152
12153 057436 130574 000001 000032 . WORD UWRD1,UWRD2,UWRD3
12154
12155 ;
12156 ; 6574:
12157 ; D_MD, ;. READ 2ND DATA ITEM
12158 ; R4_D, ;. INTO R4
12159 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12160
12161 057444 137000 010001 127014 . WORD UWRD1,UWRD2,UWRD3
12162
12163 ;
12164 057452 125252 UWDEND
12165 ;
12166 ;
12167 ;
12168 ;
12169 ; *****
12170 ; LOAD;WRITE-2 TO LOCAL STORE TMS FUNCTION
12171 ; *****
12172 ;
12173 057454 MLDWR2:
12174 ; 6002:
12175 ; UDISP:
12176 ; D_R2, ;. LOCAL STORE ADDRESS
12177 ; J/560
12178
12179 057454 030560 010000 170212 . WORD UWRD1,UWRD2,UWRD3
12180
12181 ;
12182 ; 6560:
12183 ; ENTER.TMS, TMSPTR_LOAD.WRITE.2(561), ;. ENTER INTO TMS ROUTINE
12184 ; J/561
12185
12186 057462 030561 100660 120214 . WORD UWRD1,UWRD2,UWRD3
12187
12188 ;
12189 ; 6561: ;. NULL WORD #1
12190 ; D_R3, ;. MUST DO TO SUPPORT TMS
12191 ; J/562
12192
12193 057470 030562 010000 170232 . WORD UWRD1,UWRD2,UWRD3
12194
12195 ;
12196 ; 6562: ;. NULL WORD #2
12197 ; D_R4, ;. MUST DO TO SUPPORT TMS
12198 ; J/563
12199
12200 057476 030563 010000 170214 . WORD UWRD1,UWRD2,UWRD3
12201
12202 ;
12203 ; 6563:
12204 ; GOTO-PAGE(0), ;. EXIT TO BUT(SERVICE) WORD
12205 ; J/BRA05
12206
12207 057504 034003 000000 000000 . WORD UWRD1,UWRD2,UWRD3
```

```
12208
12209
12210 057512 125252 ; UWDEND
12211 ;
12212 ;
12213 ;
12214 ;
12215 ; *****
12216 ; READ INDIRECT TMS ROM FUNCTION
12217 ; *****
12218 ;
12219 057514 ; MRDIND:
12220 ; 6002:
12221 ; UDISP:
12222 ; D_R2, ; LOCAL STORE ADDRESS
12223 ; J/550
12224 ;
12225 057514 030550 010000 170212 ; WORD UWRD1,UWRD2,UWRD3
12226 ;
12227 ;
12228 ; 6550:
12229 ; ENTER.TMS, TMSPTR_READ.IND(571), ; ENTER INTO TMS ROUTINE
12230 ; J/551
12231 ;
12232 057522 030551 100660 122214 ; WORD UWRD1,UWRD2,UWRD3
12233 ;
12234 ;
12235 ; 6551: ; NULL WORD #1
12236 ; RETURN_BRA05, ; WILL EXIT WCS THRU HERE
12237 ;
12238 ; GOTO-PAGE(6),
12239 ; J/552
12240 ;
12241 057530 034552 100000 000007 ; WORD UWRD1,UWRD2,UWRD3
12242 ;
12243 ;
12244 ; 6552: ; NULL WORD #2
12245 ; MD_DATA, ; MUST DO TO SUPPORT TMS ROUTINE
12246 ; J/553
12247 ;
12248 057536 030553 000010 007000 ; WORD UWRD1,UWRD2,UWRD3
12249 ;
12250 ;
12251 ; 6553:
12252 ; D_MD, ; GET RETURNED DATA
12253 ; R3_D, ; INTO R3
12254 ; BJT(RETURN) ; EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12255 ;
12256 057544 137000 010001 127032 ; WORD UWRD1,UWRD2,UWRD3
12257 ;
12258 ;
12259 057552 125252 ; UWDEND
12260 ;
12261 ;
12262 ;
12263 ;
```



```
12264 ; ;*****  
12265 ; ;WRITE INDIRECT TMS ROM FUNCTION  
12266 ; ;*****  
12267 ;  
12268 057554 MWR IND:  
12269 ; 6002:  
12270 ; UDISP:  
12271 ; D_R2, ; LOCAL STORE ADDRESS  
12272 ; J/540  
12273 ;  
12274 057554 030540 010000 170212 . WORD UWRD1,UWRD2,UWRD3  
12275 ;  
12276 ;  
12277 ; 6540:  
12278 ; ENTER.TMS, TMSPTR_WRITE.IND(577), ; ENTER INTO TMS ROUTINE  
12279 ; J/541  
12280 ;  
12281 057562 030541 100660 123614 . WORD UWRD1,UWRD2,UWRD3  
12282 ;  
12283 ;  
12284 ; 6541: ; NULL WORD #1  
12285 ; RETURN_BRA05, ; WILL EXIT WCS THRU HERE  
12286 ; GOTO-PAGE(6),  
12287 ; J/542  
12288 ;  
12289 057570 034542 100000 000007 . WORD UWRD1,UWRD2,UWRD3  
12290 ;  
12291 ;  
12292 ; 6542: ; NULL WORD #2  
12293 ; MD_DATA, ; MUST DO TO SUPPORT TMS ROUTINE  
12294 ; J/543  
12295 ;  
12296 057576 030543 000010 007000 . WORD UWRD1,UWRD2,UWRD3  
12297 ;  
12298 ;  
12299 ; 6543:  
12300 ; ; NOTE: DATA WRITTEN COMES FROM WCSA 0 , MED 020/220  
12301 ; BUT(RETURN) ; EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"  
12302 ;  
12303 057604 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3  
12304 ;  
12305 ;  
12306 057612 125252 UWDEND  
12307 ;  
12308 ;  
12309 ; *****  
12310 ; ;LOAD: INCR-READ 2 TMS ROM FUNCTIONS  
12311 ; ;*****  
12312 ;  
12313 057614 MICRD2:  
12314 ; 6002:  
12315 ; UDISP:  
12316 ; D_R2 MINUS 2, ; LOCAL STORE ADDRESS  
12317 ; J/410  
12318 ;  
12319 057614 030410 010000 157612 . WORD UWRD1,UWRD2,UWRD3
```

12320							
12321							
12322					6410:		
12323						ENTER.TMS, TMSPTR_SET.LOAD(251),	; . ENTER INTO TMS ROUTINE
12324						J/411	
12325					. WORD	UWRD1, UWRD2, UWRD3	
12326	057622	030411	040660	022204			
12327							
12328							
12329					6411:	; . NULL WORD #1	
12330						RETURN_BRA05,	; . WILL EXIT WCS THRU HERE
12331						GOTO-PAGE(6),	
12332						J/412	
12333					. WORD	UWRD1, UWRD2, UWRD3	
12334	057630	034412	100000	000007			
12335							
12336							
12337					6412:	; . NULL WORD #2	
12338						D_0,	; . ZAP ADDR TO MARGIN WCS
12339						J/413	
12340					. WORD	UWRD1, UWRD2, UWRD3	
12341	057636	030413	010000	030000			
12342							
12343							
12344					6413:		
12345						ENTER.TMS, TMSPTR_INC.READ.2(556),	; . ENTER INTO TMS ROUTINE
12346						J/414	
12347					. WORD	UWRD1, UWRD2, UWRD3	
12348	057644	030414	100660	023414			
12349							
12350							
12351					6414:	; . NULL WORD #1	
12352						J/415	
12353					. WORD	UWRD1, UWRD2, UWRD3	
12354	057652	030415	000000	000000			
12355							
12356							
12357					6415:	; . NULL WORD #2	
12358						MD_DATA,	; . MUST DO TO SUPPORT TMS
12359						J/416	
12360					. WORD	UWRD1, UWRD2, UWRD3	
12361	057660	030416	000010	007000			
12362							
12363							
12364					6416:		
12365						R3_D,	GET FIRST DATA ITEM INTO R3
12366						J/417	
12367					. WORD	UWRD1, UWRD2, UWRD3	
12368	057666	130417	000001	000032			
12369							
12370							
12371					6417:		
12372						D_MD,	; . READ 2ND DATA ITEM
12373						R4_D,	; . INTO R4
12374						BUT(RETURN)	; . EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12375							



```
12376 057674 137000 010001 127014 . WORD UWRD1,UWRD2,UWRD3
12377
12378
12379
12380 057702 125252 UWDEND
12381
12382
12383
12384
12385 ;*****
12386 ;LOAD;WRITE-2 TMS ROM FUNCTIONS
12387 ;*****
12388
12389 057704 MWR2:
12390 ; 6002:
12391 ; UDISP:
12392 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12393 ; J/430
12394
12395 057704 030430 010000 156212 . WORD UWRD1,UWRD2,UWRD3
12396
12397
12398 ; 6430:
12399 ; ENTER.TMS, TMSPTR_SET.STORE(254), ;. ENTER INTO TMS ROUTINE
12400 ; J/431
12401
12402 057712 030431 040660 023004 . WORD UWRD1,UWRD2,UWRD3
12403
12404
12405 ; 6431: ;. NULL WORD #1
12406 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12407 ; GOTO-PAGE(6),
12408 ; J/432
12409
12410 057720 034432 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12411
12412
12413 ; 6432: ;. NULL WORD #2
12414 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12415 ; J/433
12416
12417 057726 030433 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12418
12419
12420 ; 6433:
12421 ; ENTER.TMS, TMSPTR_WRITE.2(565), ;. ENTER INTO TMS ROUTINE
12422 ; J/434
12423
12424 057734 030434 100660 121214 . WORD UWRD1,UWRD2,UWRD3
12425
12426
12427 ; 6434: ;. NULL WORD #1
12428 ; D_R3, ;. MUST DO TO SUPPORT TMS
12429 ; J/435
12430
12431 057742 030435 010000 170232 . WORD UWRD1,UWRD2,UWRD3
```





```
12488 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12489 ;
12490 060016 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12491 ;
12492 ;
12493 060024 125252 ; UWDEND
12494 ;
12495 ;
12496 ;
12497 ; ;*****
12498 ; STORE FP.AC<0,1> TO LOCAL STORE
12499 ; ;*****
12500 ;
12501 060026 ; MFO1LS:
12502 ; 6002:
12503 ; UDISP:
12504 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12505 ;
12506 ; J/610
12507 ;
12508 060026 030610 010000 156212 . WORD UWRD1,UWRD2,UWRD3
12509 ;
12510 ;
12511 ; 6610:
12512 ; ENTER.TMS, TMSPTR_STORE.FP.01(654), ;. ENTER INTO TMS ROUTINE
12513 ; J/611
12514 ;
12515 060034 030611 140660 023004 . WORD UWRD1,UWRD2,UWRD3
12516 ;
12517 ;
12518 ; 6611: ;. NULL WORD #1
12519 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12520 ; GOTO-PAGE(6),
12521 ; J/612
12522 ;
12523 060042 034612 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12524 ;
12525 ;
12526 ; 6612: ;. NULL WORD #2
12527 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12528 ; J/613
12529 ;
12530 060050 030613 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12531 ;
12532 ;
12533 ; 6613:
12534 ; BJT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12535 ;
12536 060056 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12537 ;
12538 ;
12539 060064 125252 ; UWDEND
12540 ;
12541 ;
12542 ;
12543 ; ;*****
```

```
12544 ; ;LOAD FP.AC<2,3> FROM LOCAL STORE
12545 ; ;*****
12546 ;
12547 060066 ;MLSF23.
12548 ; 6002:
12549 ; UDISP:
12550 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12551 ; J/620
12552 ;
12553 060066 030620 010000 156212 ;. WORD UWRD1,UWRD2,UWRD3
12554 ;
12555 ;
12556 ; 6620:
12557 ; ENTER.TMS, TMSPTR_LOAD.FP.23(622), ;. ENTER INTO TMS ROUTINE
12558 ; J/621
12559 ;
12560 060074 030621 140660 120400 ;. WORD UWRD1,UWRD2,UWRD3
12561 ;
12562 ;
12563 ; 6621: ;. NULL WORD #1
12564 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12565 ; GOTO-PAGE(6),
12566 ; J/622
12567 ;
12568 060102 034622 100000 000007 ;. WORD UWRD1,UWRD2,UWRD3
12569 ;
12570 ;
12571 ; 6622: ;. NULL WORD #2
12572 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12573 ; J/623
12574 ;
12575 060110 030623 010000 030000 ;. WORD UWRD1,UWRD2,UWRD3
12576 ;
12577 ;
12578 ; 6623:
12579 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12580 ;
12581 060116 037000 000000 000000 ;. WORD UWRD1,UWRD2,UWRD3
12582 ;
12583 ;
12584 060124 125252 ;UWDEND
12585 ;
12586 ;
12587 ;
12588 ; ;*****
12589 ; ;STORE FP.AC<2,3> TO LOCAL STORE
12590 ; ;*****
12591 ;
12592 060126 ;MF23LS:
12593 ; 6002:
12594 ; UDISP:
12595 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12596 ; J/630
12597 ;
12598 060126 030630 010000 156212 ;. WORD UWRD1,UWRD2,UWRD3
12599 ;
```



```
12600 ;
12601 ;
12602 ; 6630: ENTER.TMS, TMSPTR_STORE.FP.23(671), ;. ENTER INTO TMS ROUTINE
12603 ; J/631
12604 ;
12605 060134 030631 140660 122204 . WORD UWRD1,UWRD2,UWRD3
12606 ;
12607 ;
12608 ; 6631: ;. NULL WORD #1
12609 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12610 ; GOTO-PAGE(6),
12611 ; J/632
12612 ;
12613 060142 034632 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12614 ;
12615 ;
12616 ; 6632: ;. NULL WORD #2
12617 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12618 ; J/633
12619 ;
12620 060150 030633 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12621 ;
12622 ;
12623 ; 6633:
12624 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12625 ;
12626 060156 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12627 ;
12628 ;
12629 ;
12630 060164 125252 ; UWDEND
12631 ;
12632 ;
12633 ;
12634 ; *****
12635 ; LOAD FP.AC<4,5> FROM LOCAL STORE
12636 ; *****
12637 ;
12638 060166 ; MLSF45:
12639 ; 6002:
12640 ; UDISP:
12641 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12642 ; J/640
12643 ;
12644 060166 030640 010000 156212 . WORD UWRD1,UWRD2,UWRD3
12645 ;
12646 ;
12647 ; 6640:
12648 ; ENTER.TMS, TMSPTR_LOAD.FP.45(637), ;. ENTER INTO TMS ROUTINE
12649 ; J/641
12650 ;
12651 060174 030641 140660 123600 . WORD UWRD1,UWRD2,UWRD3
12652 ;
12653 ;
12654 ; 6641: ;. NULL WORD #1
12655 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
```

```
12656 ; GOTO-PAGE(6),
12657 ; J/642
12658 ;
12659 060202 034642 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12660 ;
12661 ;
12662 ; 6642: ;. NULL WORD #2
12663 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12664 ; J/643
12665 ;
12666 060210 030643 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12667 ;
12668 ;
12669 ; 6643:
12670 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12671 ;
12672 060216 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12673 ;
12674 ;
12675 060224 125252 ; UWDEND
12676 ;
12677 ;
12678 ;
12679 ; *****
12680 ; STORE FP.AC<4,5> TO LOCAL STORE
12681 ; *****
12682 ;
12683 060226 ; MF45LS:
12684 ; 6002:
12685 ; UDISP:
12686 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12687 ; J/650
12688 ;
12689 060226 030650 010000 156212 . WORD UWRD1,UWRD2,UWRD3
12690 ;
12691 ;
12692 ; 6650:
12693 ; ENTER.TMS, TMSPTR_STORE.FP.45(706), ;. ENTER INTO TMS ROUTINE
12694 ; J/651
12695 ;
12696 060234 030651 140660 021410 . WORD UWRD1,UWRD2,UWRD3
12697 ;
12698 ;
12699 ; 6651: ;. NULL WORD #1
12700 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12701 ; GOTO-PAGE(6),
12702 ; J/652
12703 ;
12704 060242 034652 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12705 ;
12706 ;
12707 ; 6652: ;. NULL WORD #2
12708 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12709 ; J/653
12710 ;
12711 060250 030653 010000 030000 . WORD UWRD1,UWRD2,UWRD3
```



```
12712  
12713 ;  
12714 ; 6653:  
12715 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"  
12716 ;  
12717 060256 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
12718 ;  
12719 ;  
12720 060264 125252 ; UWDEND  
12721 ;  
12722 ;  
12723 ;  
12724 ; ;*****  
12725 ; ;LOAD ASP<00: 37> FROM LOCAL STORE  
12726 ; ;*****  
12727 ;  
12728 060266 ; MLSASP:  
12729 ; 6002:  
12730 ; UDISP:  
12731 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS  
12732 ; J/660  
12733 ;  
12734 060266 030660 010000 156212 . WORD UWRD1, UWRD2, UWRD3  
12735 ;  
12736 ;  
12737 ; 6660:  
12738 ;  
12739 ; ENTER. TMS. TMSPTR_LOAD. ASPAD(256), ;. ENTER INTO TMS ROUTINE  
12740 ; J/661  
12741 ;  
12742 060274 030661 040660 023404 . WORD UWRD1, UWRD2, UWRD3  
12743 ;  
12744 ;  
12745 ; 6661: ;. NULL WORD #1  
12746 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE  
12747 ; GOTO-PAGE(6),  
12748 ; J/662  
12749 ;  
12750 060302 034662 100000 000007 . WORD UWRD1, UWRD2, UWRD3  
12751 ;  
12752 ;  
12753 ; 6662: ;. NULL WORD #2  
12754 ; D_0, ;. ZAP ADDR TO MARGIN WCS  
12755 ; J/663  
12756 ;  
12757 060310 030663 010000 030000 . WORD UWRD1, UWRD2, UWRD3  
12758 ;  
12759 ;  
12760 ; 6663:  
12761 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"  
12762 ;  
12763 060316 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
12764 ;  
12765 ;  
12766 060324 125252 ; UWDEND  
12767 ;
```

```
12768 ;
12769 ;
12770 ; *****
12771 ; STORE ASP<00: 37> TO LOCAL STORE
12772 ; *****
12773 ;
12774 060326 MASPLS:
12775 ; 6002:
12776 ; UDISP:
12777 ; D_R2 MINUS 1, ; LOCAL STORE ADDRESS
12778 ; J/670
12779 ;
12780 060326 030670 010000 156212 . WORD UWRD1, UWRD2, UWRD3
12781 ;
12782 ;
12783 ; 6670:
12784 ; ENTER TMS, TMSPTR_STORE, ASPAD(323), ; ENTER INTO TMS ROUTINE
12785 ; J/671
12786 ;
12787 060324 030671 040660 120610 . WORD UWRD1, UWRD2, UWRD3
12788 ;
12789 ;
12790 ; 6671: ; NULL WORD #1
12791 ; RETURN_BRA05, ; WILL EXIT WCS THRU HERE
12792 ; GOTO-PAGE(6),
12793 ; J/672
12794 ;
12795 060342 034672 100000 000007 . WORD UWRD1, UWRD2, UWRD3
12796 ;
12797 ;
12798 ; 6672: ; NULL WORD #2
12799 ; D_0, ; ZAP ADDR TO MARGIN WCS
12800 ; J/673
12801 ;
12802 060350 030673 010000 030000 . WORD UWRD1, UWRD2, UWRD3
12803 ;
12804 ;
12805 ; 6673:
12806 ; BUT(RETURN) ; EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12807 ;
12808 060356 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3
12809 ;
12810 ;
12811 060364 125252 ; UWDEND
12812 ;
12813 ;
12814 ; *****
12815 ; LOAD BSP<00: 37> FROM LOCAL STORE
12816 ; *****
12817 ;
12818 ;
12819 060366 MLBSBP:
12820 ; 6002:
12821 ; UDISP:
12822 ; D_R2 MINUS 1, ; LOCAL STORE ADDRESS
12823 ; J/700
```



```
12824
12825 060366 030700 010000 156212 . WORD UWRD1, UWRD2, UWRD3
12826
12827 ;
12828 ; 6700:
12829 ; ENTER. TMS, TMSPTR_LOAD. BSPAD(367), ;. ENTER INTO TMS ROUTINE
12830 ; J/701
12831
12832 060374 030701 040660 121614 . WORD UWRD1, UWRD2, UWRD3
12833
12834 ;
12835 ; 6701: ;. NULL WORD #1
12836 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12837 ; GOTO-PAGE(6),
12838 ; J/702
12839
12840 060402 034702 100000 000007 . WORD UWRD1, UWRD2, UWRD3
12841
12842 ;
12843 ;
12844 ; 6702: ;. NULL WORD #2
12845 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12846 ; J/703
12847
12848 060410 030703 010000 030000 . WORD UWRD1, UWRD2, UWRD3
12849
12850 ;
12851 ;
12852 ; 6703:
12853 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12854 060416 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3
12855
12856 ;
12857 060424 125252 ; UWDEND
12858 ;
12859 ;
12860 ;
12861 ; *****
12862 ; STORE BSP<00: 37> TO LOCAL STORE
12863 ; *****
12864 ;
12865 060426 ; MBSPLS:
12866 ; 6002:
12867 ; UDISP:
12868 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12869 ; J/710
12870
12871 060426 030710 010000 156212 . WORD UWRD1, UWRD2, UWRD3
12872
12873 ;
12874 ; 6710:
12875 ; ENTER. TMS, TMSPTR_STORE. BSPAD(434), ;. ENTER INTO TMS ROUTINE
12876 ; J/711
12877
12878 060434 030711 100660 123000 . WORD UWRD1, UWRD2, UWRD3
12879
```

```

12880 ;
12881 ; 6711: ;. NULL WORD #1
12882 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12883 ; GOTO-PAGE(6),
12884 ; J/712
12885 ;
12886 060442 034712 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12887 ;
12888 ;
12889 ; 6712: ;. NULL WORD #2
12890 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12891 ; J/713
12892 ;
12893 060450 030713 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12894 ;
12895 ;
12896 ; 6713:
12897 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12898 ;
12899 060456 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12900 ;
12901 ;
12902 060464 125252 ; UWDEND
12903 ;
12904 ;
12905 ;
12906 ; *****
12907 ; LOAD CSP<17:00> FROM LOCAL STORE
12908 ; *****
12909 ;
12910 060466 ; MLSCSP:
12911 ; 6002:
12912 ; UDISP:
12913 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12914 ; J/730
12915 ;
12916 060466 030730 010000 156212 . WORD UWRD1,UWRD2,UWRD3
12917 ;
12918 ;
12919 ; 6730:
12920 ; ENTER.TMS, TMSPTR_LOAD, ALL. CSP(501), ;. ENTER INTO TMS ROUTINE
12921 ; J/731
12922 ;
12923 060474 030731 100660 020210 . WORD UWRD1,UWRD2,UWRD3
12924 ;
12925 ;
12926 ; 6731: ;. NULL WORD #1
12927 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12928 ; GOTO-PAGE(6),
12929 ; J/732
12930 ;
12931 060502 034732 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12932 ;
12933 ;
12934 ; 6732: ;. NULL WORD #2
12935 ; D_0, ;. ZAP ADDR TO MARGIN WCS

```



```
12936 ; J/733
12937 ;
12938 060510 030733 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12939 ;
12940 ;
12941 ;
12942 ; 6733:
12943 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12944 ;
12945 060516 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12946 ;
12947 ;
12948 060524 125252 ; UWDEND
12949 ;
12950 ;
12951 ;
12952 ; *****
12953 ; STORE CSP<17:00> TO LOCAL STORE
12954 ; *****
12955 ;
12956 060526 ; MCSPLS:
12957 ; 6002:
12958 ; UDISP:
12959 ; D_R2 MINUS 1, ;. LOCAL STORE ADDRESS
12960 ; J/740
12961 ;
12962 060526 030740 010000 156212 . WORD UWRD1,UWRD2,UWRD3
12963 ;
12964 ;
12965 ; 6740:
12966 ; ENTER.TMS, TMSPTR_STORE.ALL(CSP(525), ;. ENTER INTO TMS ROUTINE
12967 ; J/741
12968 ;
12969 060534 030741 100660 121210 . WORD UWRD1,UWRD2,UWRD3
12970 ;
12971 ;
12972 ; 6741: ;. NULL WORD #1
12973 ; RETURN_BRA05, ;. WILL EXIT WCS THRU HERE
12974 ; GOTO-PAGE(6),
12975 ; J/742
12976 ;
12977 060542 034742 100000 000007 . WORD UWRD1,UWRD2,UWRD3
12978 ;
12979 ;
12980 ; 6742: ;. NULL WORD #2
12981 ; D_0, ;. ZAP ADDR TO MARGIN WCS
12982 ; J/743
12983 ;
12984 060550 030743 010000 030000 . WORD UWRD1,UWRD2,UWRD3
12985 ;
12986 ;
12987 ; 6743:
12988 ; BUT(RETURN) ;. EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
12989 ;
12990 060556 037000 000000 000000 . WORD UWRD1,UWRD2,UWRD3
12991 ;
```

```
12992 ;
12993 060564 125252 ; UWDEND
12994 ;
12995 ;
12996 ;
12997 ; *****
12998 ; LOAD T<1: 2>A, T<1: 2>B, FDST<0: 3> FROM LOCAL STORE
12999 ; *****
13000 ;
13001 060566 ; MLSTMP:
13002 ; 6002:
13003 ; UDISP:
13004 ; D_R2 MINUS 1, ; LOCAL STORE ADDRESS
13005 ; J/750
13006 ;
13007 060566 030750 010000 156212 ; WORD UWRD1, UWRD2, UWRD3
13008 ;
13009 ;
13010 ; 6750:
13011 ; ENTER. TMS, TMSPTR_LOAD. TEMP(723), ; ENTER INTO TMS ROUTINE
13012 ; J/751
13013 ;
13014 060574 030751 140660 120610 ; WORD UWRD1, UWRD2, UWRD3
13015 ;
13016 ;
13017 ; 6751: ; NULL WORD #1
13018 ; RETURN_BRA05, ; WILL EXIT WCS THRU HERE
13019 ; GOTO-PAGE(6),
13020 ; J/752
13021 ;
13022 060602 034752 100000 000007 ; WORD UWRD1, UWRD2, UWRD3
13023 ;
13024 ;
13025 ; 6752: ; NULL WORD #2
13026 ; D_0, ; ZAP ADDR TO MARGIN WCS
13027 ;
13028 ; J/753
13029 ;
13030 060610 030753 010000 030000 ; WORD UWRD1, UWRD2, UWRD3
13031 ;
13032 ;
13033 ; 6753:
13034 ; BUT(RETURN) ; EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
13035 ;
13036 060616 037000 000000 000000 ; WORD UWRD1, UWRD2, UWRD3
13037 ;
13038 ;
13039 060624 125252 ; UWDEND
13040 ;
13041 ;
13042 ;
13043 ; *****
13044 ; STORE T<1: 2>A, T<1: 2>B, FDST<0: 3> TO LOCAL STORE
13045 ; *****
13046 ;
13047 060626 ; MTMPLS:
```



```
13048 ; 6002:
13049 ; UDISP:
13050 ; D_R2 MINUS 1, ; LOCAL STORE ADDRESS
13051 ; J/760
13052
13053 060626 030760 010000 156212 . WORD UWRD1, UWRD2, UWRD3
13054
13055 ;
13056 ; 6760:
13057 ; ENTER. TMS, TMSPTR_STORE. TEMP(740), ; ENTER INTO TMS ROUTINE
13058 ; J/761
13059
13060 060634 030761 140660 020014 . WORD UWRD1, UWRD2, UWRD3
13061
13062 ;
13063 ; 6761: ; NULL WORD #1
13064 ; RETURN_BRA05, ; WILL EXIT WCS THRU HERE
13065 ; GOTO-PAGE(6),
13066 ; J/762
13067
13068 060642 034762 100000 000007 . WORD UWRD1, UWRD2, UWRD3
13069
13070 ;
13071 ; 6762: ; NULL WORD #2
13072 ; D_0, ; ZAP ADDR TO MARGIN WCS
13073 ; J/763
13074
13075 060650 030763 010000 030000 . WORD UWRD1, UWRD2, UWRD3
13076
13077 ;
13078 ; 6763:
13079 ; BUT(RETURN) ; EXIT TO "BRA05" WHICH DOES "BUT(SERVICE)"
13080
13081 060656 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13082
13083 ;
13084 060664 125252 ; UWDEND
13085 ;
13086 ;
13087 ; *****
13088 ; STORE GR'S IN LS
13089 ; *****
13090
13091 060666 ; MGRLS:
13092 ;
13093 ; 6002:
13094 ; UDISP:
13095 ; J/LSGR1
13096 060666 030026 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13097
13098 ;
13099 ; 6026:
13100 ; LSGR1:
13101 ; BUT(SUBRB),
13102 ; RETURN6_TMSXT1, PAGE(6),
13103 ; J/LSGR2
```

13104						
13105	060674	034027	100000	060073	. WORD	UWRD1, UWRD2, UWRD3
13106						
13107						
13108					6027:	
13109					LSGR2:	
13110						MD_EMIT,
13111						EMITH/O, EMITML/777,
13112						J/LSGR3
13113						
13114	060702	030030	140050	004177	. WORD	UWRD1, UWRD2, UWRD3
13115						
13116						
13117					6030:	
13118					LSGR3:	
13119						D_MD,
13120						J/LSGR4
13121						
13122	060710	030031	010000	127000	. WORD	UWRD1, UWRD2, UWRD3
13123						
13124						
13125					6031:	
13126					LSGR4:	
13127						ENTER. TMS, TMSPTR_STORE. GRS(060),
13128						J/NULL0
13129						
13130	060716	030032	000660	120004	. WORD	UWRD1, UWRD2, UWRD3
13131						
13132						
13133					6032:	
13134					NULL0:	
13135						J/NULL1
13136						
13137	060724	030033	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13138						
13139						
13140					6033:	
13141					NULL1:	
13142						J/TMSXTO
13143						
13144	060732	030034	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13145						
13146						
13147					6034:	
13148					TMSXTO:	
13149						BUT(RETURN)
13150						
13151	060740	037000	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13152						
13153						
13154					6035:	
13155					TMSXT1:	
13156						BUT(SERVICE),
13157						J/NOSVC3
13158						
13159	060746	020036	000000	000000	. WORD	UWRD1, UWRD2, UWRD3



```
13160
13161 ;
13162 ; 6036:
13163 ; NOSVC3:
13164 ; GOTO-PAGE(0),
13165 ; J/FETO1
13166
13167 060754 034702 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13168
13169 ;
13170 ; 6037:
13171 ; SVC3:
13172 ; GOTO-PAGE(0),
13173 ; J/SERO2
13174
13175 060762 034703 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13176
13177 ;
13178 060770 125252 UWDEND
13179 ;
13180 ; *****
13181 ; LOAD LS INTO GR'S
13182 ; *****
13183 ;
13184 060772 MLSGR:
13185 ; 6002:
13186 ; UDISP:
13187 ; J/GRLS1
13188
13189 060772 030136 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13190
13191 ;
13192 ;
13193 ; 6136:
13194 ; GRLS1:
13195 ; BUT(SUBRB),
13196 ; RETURN6_TMSXT3, PAGE(6),
13197 ; J/GRLS2
13198
13199 061000 034137 100000 060313 . WORD UWRD1,UWRD2,UWRD3
13200
13201 ;
13202 ; 6137:
13203 ; GRLS2:
13204 ; MD_EMIT,
13205 ; EMITH/O,EMITML/2524,
13206 ; J/GRLS3
13207
13208 061006 030140 000050 004525 . WORD UWRD1,UWRD2,UWRD3
13209
13210 ;
13211 ; 6140:
13212 ; GRLS3:
13213 ; D_MD,
13214 ; J/GRLS4
13215
```

Address	Op1	Op2	Op3	Op4	Op5	Op6
13216	061014	030141	010000	127000	. WORD	UWRD1, UWRD2, UWRD3
13217						
13218						
13219						
13220					6141:	
13221					GRLS4:	
13222						ENTER. TMS, TMSPTR_LOAD. GRS(042),
13223						J/NULL2
13224	061022	030142	000660	020404	. WORD	UWRD1, UWRD2, UWRD3
13225						
13226						
13227					6142:	
13228					NULL2:	
13229						J/NULL3
13230						
13231	061030	030143	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13232						
13233						
13234					6143:	
13235					NULL3:	
13236						J/TMSXT2
13237						
13238	061036	030144	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13239						
13240						
13241					6144:	
13242					TMSXT2:	
13243						BUT(RETURN)
13244						
13245	061044	037000	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13246						
13247						
13248					6145:	
13249					TMSXT3:	
13250						BUT(SERVICE),
13251						J/NOSVC4
13252						
13253	061052	020146	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13254						
13255						
13256					6146:	
13257					NOSVC4:	
13258						GOTO-PAGE(0),
13259						J/FET01
13260						
13261	061060	034702	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13262						
13263						
13264					6147:	
13265					SV4:	
13266						GOTO-PAGE(0),
13267						J/SER02
13268						
13269	061066	034703	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13270						
13271						





```

13328 ; NULL5:
13329 ; J/TMSXT4
13330 ;
13331 061142 030054 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13332 ;
13333 ;
13334 ; 6054:
13335 ; TMSXT4:
13336 ; BUT(RETURN)
13337 ;
13338 061150 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13339 ;
13340 ;
13341 ; 6055:
13342 ; TMSXT5:
13343 ; BUT(SERVICE),
13344 ; J/NOSVC5
13345 ;
13346 061156 020056 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13347 ;
13348 ;
13349 ; 6056:
13350 ; NOSVC5:
13351 ; GOTO-PAGE(0),
13352 ; J/FETO1
13353 ;
13354 061164 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13355 ;
13356 ;
13357 ; 6057:
13358 ;
13359 ; SVCS:
13360 ; GOTO-PAGE(0),
13361 ; J/SERO2
13362 ;
13363 061172 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13364 ;
13365 ;
13366 061200 125252 ; UNDEND
13367 ;
13368 ; *****
13369 ; LOAD LS INTO WFP REGISTERS
13370 ; *****
13371 ;
13372 061202 ; MLSFP:
13373 ; 6002:
13374 ; UDISP:
13375 ; J/WFLS1
13376 ;
13377 061202 030256 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13378 ;
13379 ;
13380 ; 6256:
13381 ; WFLS1:
13382 ; BUT(SUBRB),
13383 ; RETURN6_TMSXT7, PAGE(6).
    
```



13384						J/WFLS2
13385						
13386	061210	034257	100000	060553	. WORD	UWRD1, UWRD2, UWRD3
13387						
13388						
13389						6257:
13390						WFLS2:
13391						MD_EMIT,
13392						EMITH/O, EMITML/1077,
13393						J/WFLS3
13394						
13395	061216	030260	140050	004217	. WORD	UWRD1, UWRD2, UWRD3
13396						
13397						
13398						6260:
13399						WFLS3:
13400						
13401						D_MD,
13402						J/WFLS4
13403	061224	030261	010000	127000	. WORD	UWRD1, UWRD2, UWRD3
13404						
13405						
13406						6261:
13407						WFLS4:
13408						ENTER. TMS, TMSPTR_LOAD. FP(076),
13409						J/NULL6
13410						
13411	061232	030262	000660	123404	. WORD	UWRD1, UWRD2, UWRD3
13412						
13413						
13414						6262:
13415						NULL6:
13416						
13417						J/NULL7
13418	061240	030263	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13419						
13420						
13421						6263:
13422						NULL7:
13423						
13424						J.TMSXT6
13425	061246	030264	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13426						
13427						
13428						6264:
13429						TMSXT6:
13430						BJT(RETURN)
13431						
13432	061254	037000	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13433						
13434						
13435						6265:
13436						TMSXT7:
13437						
13438						BUT(SERVICE),
13439						J/NOSVC6

```
13440 061262 020266 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13441
13442 ;
13443 ; 6266:
13444 ; NOSVC6:
13445 ; GOTO-PAGE(0),
13446 ; J/FETO1
13447
13448 061270 034702 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13449
13450 ;
13451 ; 6267:
13452 ; SVC6:
13453 ; GOTO-PAGE(0),
13454 ; J/SERO2
13455
13456 061276 034703 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13457
13458 ;
13459 061304 125252 UWDEND
13460 ;
13461 ; *****
13462 ; STORE CSP IN LS
13463 ; *****
13464 ;
13465 061306 MCSLS:
13466 ; 6002:
13467 ; UDISP:
13468 ; J/LSCSP1
13469
13470 061306 030066 000000 000000 . WORD UWRD1,UWRD2,UWRD3
13471
13472 ;
13473 ; 6066:
13474 ; LSCSP1:
13475 ; BUT(SUBRB),
13476 ; RETURN6_TMSXT9, PAGE(6),
13477 ; J/LSCSP2
13478
13479 061314 034067 100000 060173 . WORD UWRD1,UWRD2,UWRD3
13480
13481 ;
13482 ; 6067:
13483 ; LSCSP2:
13484 ; MD_EMIT,
13485 ; EMITH/O,EMITML/1037,
13486 ; J/LSCSP3
13487
13488 061322 030070 140050 004207 . WORD UWRD1,UWRD2,UWRD3
13489
13490 ;
13491 ; 6070:
13492 ; LSCSP3:
13493 ; D_MD,
13494 ; J/LSCSP4
13495
```



13496	061330	030071	010000	127000	. WORD	UWRD1, UWRD2, UWRD3
13497						
13498						
13499					6071:	
13500					LSCSP4:	
13501						ENTER.TMS, TMSPTR_STORE.(CSP(210),
13502						J/NULL8
13503						
13504	061336	030072	040660	022000	. WORD	UWRD1, UWRD2, UWRD3
13505						
13506						
13507						
13508					6072:	
13509					NULL8:	
13510						J/NULL9
13511						
13512	061344	030073	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13513						
13514						
13515					6073:	
13516					NULL9:	
13517						J/TMSXT8
13518						
13519	061352	030074	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13520						
13521						
13522					6074:	
13523					TMSXT8:	
13524						BUT(RETURN)
13525						
13526	061360	037000	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13527						
13528						
13529					6075:	
13530					TMSXT9:	
13531						BUT(SERVICE),
13532						J/NSVC7
13533						
13534	061366	020076	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13535						
13536						
13537					6076:	
13538					NSVC7:	
13539						GOTO-PAGE(0),
13540						J/FETO1
13541						
13542	061374	034702	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13543						
13544						
13545					6077:	
13546					SVC7:	
13547						GOTO-PAGE(0),
13548						J/SERO2
13549						
13550	061402	034703	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13551						

```
13552 ;
13553 061410 125252 ; UWDEND
13554 ;
13555 ; *****
13556 ; LOAD LS IN CSP
13557 ; *****
13558 ;
13559 061412 ; MLSCS:
13560 ; 6002:
13561 ; UDISP:
13562 ; J/CSPLS1
13563 ;
13564 061412 030476 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13565 ;
13566 ;
13567 ; 6476:
13568 ; CSPLS1:
13569 ; BUT(SUBRB),
13570 ; RETURN6_TMSXT11, PAGE(6),
13571 ; J/CSPLS2
13572 ;
13573 061420 034477 100000 061213 . WORD UWRD1, UWRD2, UWRD3
13574 ;
13575 ;
13576 ; 6477:
13577 ; CSPLS2:
13578 ; MD_EMIT,
13579 ; EMITH/O, EMITML/1017,
13580 ; J/CSPLS3
13581 ;
13582 061426 030500 140050 004203 . WORD UWRD1, UWRD2, UWRD3
13583 ;
13584 ;
13585 ; 6500:
13586 ; CSPLS3:
13587 ; D_MD,
13588 ; J/CSPLS4
13589 ;
13590 061434 030501 010000 127000 . WORD UWRD1, UWRD2, UWRD3
13591 ;
13592 ;
13593 ; 6501:
13594 ; CSPLS4:
13595 ; ENTER.TMS, TMSPTR_LOAD.CSP(170),
13596 ; J/NULL10
13597 ;
13598 061442 030502 000660 122014 . WORD UWRD1, UWRD2, UWRD3
13599 ;
13600 ;
13601 ; 6502:
13602 ; NULL10:
13603 ; J/NULL11
13604 ;
13605 061450 030503 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13606 ;
13607 ;
```



```
13608 ; 6503:  
13609 ; NULL11:  
13610 ; J/TMSXT10  
13611 ;  
13612 061456 030504 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13613 ;  
13614 ;  
13615 ; 6504:  
13616 ; TMSXT10:  
13617 ; BUT(RETURN)  
13618 ;  
13619 061464 037000 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13620 ;  
13621 ;  
13622 ; 6505:  
13623 ; TMSXT11:  
13624 ; BUT(SERVICE),  
13625 ; J/NOSVC8  
13626 ;  
13627 061472 020506 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13628 ;  
13629 ;  
13630 ; 6506:  
13631 ; NOSVC8:  
13632 ; GOTO-PAGE(0),  
13633 ; J/FET01  
13634 ;  
13635 061500 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13636 ;  
13637 ;  
13638 ; 6507:  
13639 ; SVC8:  
13640 ; GOTO-PAGE(0),  
13641 ; J/SER02  
13642 ;  
13643 061506 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13644 ;  
13645 ;  
13646 061514 125252 ; UNDEND  
13647 ;  
13648 ; *****  
13649 ; STORE WCS USER SP IN LS  
13650 ;  
13651 ; *****  
13652 ;  
13653 061516 ; MWSLS:  
13654 ; 6002:  
13655 ; UDISP:  
13656 ; J/LSWS1  
13657 ;  
13658 061516 030106 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13659 ;  
13660 ;  
13661 ; 6106:  
13662 ; LSWS1:  
13663 ; BUT(SUBRB),
```

13664						RETURN6_TMSXT13, PAGE(6),
13665						J/LSWS2
13666						
13667	061524	034107	100000	060233	. WORD	UWRD1, UWRD2, UWRD3
13668						
13669						
13670						6107:
13671						LSWS2:
13672						
13673						MD_EMIT,
13674						EMITH/O, EMITML/1007,
13675						J/LSWS3
13676	061532	030110	140050	004201	. WORD	UWRD1, UWRD2, UWRD3
13677						
13678						
13679						6110:
13680						LSWS3:
13681						
13682						D_MD,
13683						J/LSWS4
13684	061540	030111	010000	127000	. WORD	UWRD1, UWRD2, UWRD3
13685						
13686						
13687						6111:
13688						LSWS4:
13689						
13690						ENTER_TMS, TMSPTR_STORE, WCS, AB(241),
13691						J/NULL12
13692	061546	030112	040660	020204	. WORD	UWRD1, UWRD2, UWRD3
13693						
13694						
13695						6112:
13696						NULL12:
13697						
13698						J/NULL13
13699	061554	030113	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13700						
13701						
13702						6113:
13703						NULL13:
13704						
13705						J/TMSXT12
13706	061562	030114	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13707						
13708						
13709						6114:
13710						TMSXT12:
13711						
13712						BUT(RETURN)
13713	061570	037000	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13714						
13715						
13716						6115:
13717						TMSXT13:
13718						
13719						BUT(SERVICE), J/NOSVC10



```

13720
13721 061576 020116 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13722
13723 ;
13724 ; 6116:
13725 ; NOSVC10:
13726 ; GOTO-PAGE(0),
13727 ; J/FETO1
13728
13729 061604 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13730
13731 ;
13732 ; 6117:
13733 ; SVC10:
13734 ; GOTO-PAGE(0),
13735 ; J/SERD2
13736
13737 061612 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13738
13739 ;
13740 061620 125252 UWDEND
13741 ;
13742 ; *****
13743 ; LOAD LS IN WCS USER SP
13744 ; *****
13745 ;
13746 061622 MLSWS:
13747 ; 6002:
13748 ; UDISP:
13749 ; J/WLS1
13750
13751 061622 030716 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13752
13753 ;
13754 ; 6716:
13755 ; WLS1:
13756 ; BUT(SUBRB),
13757 ; RETURN6_TMSXT15, PAGE(6),
13758 ; J/WLS2
13759
13760 061630 034717 100000 061653 . WORD UWRD1, UWRD2, UWRD3
13761
13762 ;
13763 ; 6717:
13764 ; WLS2:
13765 ; MD_EMIT,
13766 ; EMITH/0, EMITML/1003,
13767 ; J/WLS3
13768
13769 061636 030720 140050 004200 . WORD UWRD1, UWRD2, UWRD3
13770
13771 ;
13772 ; 6720:
13773 ; WLS3:
13774 ; D_MD,
13775 ; J/WLS4
    
```

13776						
13777	061644	030721	010000	127000	. WORD	UWRD1, UWRD2, UWRD3
13778						
13779						
13780					6721:	
13781					WLS4:	
13782						
13783						ENTER. TMS, TMSPTR_LOAD. WCS. AB(231),
13784						J/NULL14
13785						
13786	061652	030722	040660	122200	. WORD	UWRD1, UWRD2, UWRD3
13787						
13788						
13789					6722:	
13790					NULL14:	
13791						J/NULL15
13792						
13793	061660	030723	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13794						
13795						
13796					6723:	
13797					NULL15:	
13798						J/TMSXT14
13799						
13800	061666	030724	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13801						
13802						
13803					6724:	
13804					TMSXT14:	
13805						BUT (RETURN)
13806						
13807	061674	037000	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13808						
13809						
13810					6725:	
13811					TMSXT15:	
13812						BUT (SERVICE),
13813						J/NOSVC9
13814						
13815	061702	020726	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13816						
13817						
13818					6726:	
13819					NOSVC9:	
13820						GOTO-PAGE (0),
13821						J/FETO1
13822						
13823	061710	034702	000000	000000	. WORD	UWRD1, UWRD2, UWRD3
13824						
13825						
13826					6727:	
13827					SVC9:	
13828						GOTO-PAGE (0),
13829						J/SERO2
13830						
13831	061716	034703	000000	000000	. WORD	UWRD1, UWRD2, UWRD3



```
13832  
13833 ;  
13834 061724 125252 ; UWDEND  
13835 ;  
13836 ; ;*****  
13837 ; ;USER DISPATCH-XFC. WCS ENTRY POINT 6002  
13838 ; ;*****  
13839 ;  
13840 061726 ; MXFC2:  
13841 ; 6002:  
13842 ; UDISP:  
13843 ; R3_MD,  
13844 ; J/CHSVCO  
13845 ;  
13846 061726 130241 010001 127032 . WORD UWRD1, UWRD2, UWRD3  
13847 ;  
13848 ;  
13849 ; 6241:  
13850 ; CHSVCO:  
13851 ; BUT(SERVICE),  
13852 ; J/NOSVCO  
13853 ;  
13854 061734 020242 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13855 ;  
13856 ;  
13857 ; 6242:  
13858 ; NOSVCO:  
13859 ; GOTO-PAGE(0),  
13860 ; J/FETO1  
13861 ;  
13862 061742 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13863 ;  
13864 ;  
13865 ; 6243:  
13866 ; SVCO:  
13867 ; GOTO-PAGE(0),  
13868 ; J/SERO2  
13869 ;  
13870 061750 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13871 ;  
13872 ; UWDEND  
13873 061756 125252 ;  
13874 ; ;*****  
13875 ; ;*****  
13876 ; ;OTHER USER DISPATCHES-XFC. WCS ENTRY POINTS 6001,6011,...,6015  
13877 ; ;*****  
13878 ; ;*****  
13879 061760 ; MXFC1:  
13880 ; 6001:  
13881 ; OTHDIS:  
13882 ; R3_MD,  
13883 ; J/CHSVC1  
13884 ;  
13885 061760 130151 010001 127032 . WORD UWRD1, UWRD2, UWRD3
```

```
13886  
13887 ;  
13888 ; 6151:  
13889 ; CHSVC1:  
13890 ; BUT(SERVICE),  
13891 ; J/NOSVC1  
13892  
13893 061766 020152 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13894  
13895 ;  
13896 ; 6152:  
13897 ;  
13898 ; NOSVC1:  
13899 ; GOTO-PAGE(0),  
13900 ; J/FETO1  
13901  
13902 061774 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13903  
13904 ;  
13905 ; 6153:  
13906 ; SVC1:  
13907 ; GOTO-PAGE(0),  
13908 ; J/SERO2  
13909  
13910 062002 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13911  
13912 ;  
13913 062010 125252 ; UWDEND  
13914 ;  
13915 ; *****  
13916 ; XFC OPCODE ENTRY AT 6011  
13917 ; *****  
13918 ;  
13919 062012 ; MXFC11:  
13920 ; 6011:  
13921 ; OTH11:  
13922 ; R3_MD,  
13923 ; J/CHSVC13  
13924  
13925 062012 130351 010001 127032 . WORD UWRD1, UWRD2, UWRD3  
13926  
13927 ;  
13928 ; 6351:  
13929 ; CHSVC13:  
13930 ; BUT(SERVICE),  
13931 ; J/NOSVC13  
13932  
13933 062020 020352 000000 000000 . WORD UWRD1, UWRD2, UWRD3  
13934  
13935 ;  
13936 ; 6352:  
13937 ; NOSVC13:  
13938 ; GOTO-PAGE(0),  
13939 ; J/FETO1  
13940  
13941 062026 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
```



```
13942
13943 ;
13944 ; 6353:
13945 ; SVC13:
13946 ; GOTO-PAGE(0),
13947 ; J/SER02
13948
13949 062034 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13950
13951 ;
13952 062042 125252 ; UWDEND
13953 ;
13954 ; *****
13955 ; XFC OPCODE ENTRY AT 6012
13956 ; *****
13957 ;
13958 062044 ; MXFC12:
13959 ; 6012:
13960 ; OTH12:
13961 ; R3_MD,
13962 ; J/CHSVC14
13963
13964 062044 130361 010001 127032 . WORD UWRD1, UWRD2, UWRD3
13965
13966 ;
13967 ; 6361:
13968 ; CHSVC14:
13969 ; BUT(SERVICE),
13970 ; J/NOSVC14
13971
13972 062052 020362 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13973
13974 ;
13975 ; 6362:
13976 ; NOSVC14:
13977 ; GOTO-PAGE(0),
13978 ; J/FETO1
13979
13980 062060 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13981
13982 ;
13983 ; 6363:
13984 ; SVC14:
13985 ; GOTO-PAGE(0),
13986 ; J/SER02
13987
13988 062066 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
13989
13990 ;
13991 062074 125252 ; UWDEND
13992 ;
13993 ; *****
13994 ; XFC OPCODE ENTRY AT 6013
13995 ; *****
13996 062076 ; MXFC13:
13997 ; 6013:
```

```

13998 ; OTH13:
13999 ; R3_MD,
14000 ; J/CHSVC15
14001 ;
14002 062076 130371 010001 127032 . WORD UWRD1, UWRD2, UWRD3
14003 ;
14004 ;
14005 ; 6371:
14006 ; CHSVC15:
14007 ; BUT(SERVICE),
14008 ;
14009 ; J/NOSVC15
14010 ;
14011 062104 020372 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14012 ;
14013 ;
14014 ; 6372:
14015 ; NOSVC15:
14016 ; GOTO-PAGE(0),
14017 ; J/FETO1
14018 ;
14019 062112 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14020 ;
14021 ;
14022 ; 6373:
14023 ; SVC15:
14024 ; GOTO-PAGE(0),
14025 ; J/SERO2
14026 ;
14027 062120 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14028 ;
14029 ;
14030 062126 125252 ; UWDEND
14031 ;
14032 ; *****
14033 ; XFC OPCODE ENTRY AT 6014
14034 ; *****
14035 062130 ; MXFC14:
14036 ; 6014:
14037 ; OTH14:
14038 ; R3_MD,
14039 ; J/CHSVC16
14040 ;
14041 062130 130401 010001 127032 . WORD UWRD1, UWRD2, UWRD3
14042 ;
14043 ;
14044 ; 6401:
14045 ; CHSVC16:
14046 ; BUT(SERVICE),
14047 ; J/NOSVC16
14048 ;
14049 062136 020402 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14050 ;
14051 ;
14052 ; 6402:
14053 ; NOSVC16:
    
```



```
14054 ; GOTO-PAGE(0),
14055 ; J/FETO1
14056 ;
14057 062144 034702 000000 000000 . WORD UWRD1,UWRD2,UWRD3
14058 ;
14059 ;
14060 ; 6403:
14061 ; SVC16:
14062 ; GOTO-PAGE(0),
14063 ; J/SERO2
14064 ;
14065 062152 034703 000000 000000 . WORD UWRD1,UWRD2,UWRD3
14066 ;
14067 ;
14068 062160 125252 ; UWDEND
14069 ;
14070 ; ;*****
14071 ; ;XFC OPCODE ENTRY AT 6015
14072 ; ;*****
14073 062162 ; MXFC15:
14074 ; 6015:
14075 ; OTH15:
14076 ; R3_MD,
14077 ; J/CHSVC16A
14078 ;
14079 062162 130405 010001 127032 . WORD UWRD1,UWRD2,UWRD3
14080 ;
14081 ;
14082 ; 6405:
14083 ; CHSVC16A:
14084 ; BUT(SERVICE),
14085 ; J/NOSVC16A
14086 ;
14087 062170 020406 000000 000000 . WORD UWRD1,UWRD2,UWRD3
14088 ;
14089 ;
14090 ; 6406:
14091 ; NOSVC16A:
14092 ; GOTO-PAGE(0),
14093 ; J/FETO1
14094 ;
14095 062176 034702 000000 000000 . WORD UWRD1,UWRD2,UWRD3
14096 ;
14097 ;
14098 ; 6407:
14099 ; SVC16A:
14100 ; GOTO-PAGE(0),
14101 ; J/SERO2
14102 ;
14103 062204 034703 000000 000000 . WORD UWRD1,UWRD2,UWRD3
14104 ;
14105 ;
14106 062212 125252 ; UWDEND
14107 ;
14108 ; ;*****
14109 ; ;ODD PC ENTRY INTO WCS AT 6004
```

```
14110 ; ; *****
14111 ; ;
14112 ; ;
14113 062214 ; MODDPC:
14114 ; ; 6004:
14115 ; ; ODDPC:
14116 ; ; R3_R3 PLUS 1,
14117 ; ; J/CHSVC17
14118 ; ;
14119 062214 130161 010001 116232 . WORD UWRD1, UWRD2, UWRD3
14120 ; ;
14121 ; ;
14122 ; ; 6161:
14123 ; ; CHSVC17:
14124 ; ; BUT(SERVICE),
14125 ; ; J/NOSVC17
14126 ; ;
14127 062222 020162 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14128 ; ;
14129 ; ;
14130 ; ; 6162:
14131 ; ; NOSVC17:
14132 ; ; GOTO-PAGE(0),
14133 ; ; J/FETO1
14134 ; ;
14135 062230 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14136 ; ;
14137 ; ;
14138 ; ; 6163:
14139 ; ; SVC17:
14140 ; ; GOTO-PAGE(0),
14141 ; ; J/SERO2
14142 ; ;
14143 062236 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14144 ; ;
14145 ; ;
14146 062244 125252 ; UWDEND
14147 ; ;
14148 ; ; *****
14149 ; ; ;BASE MACHINE MICRO-BREAK ENTPY INTO WCS AT 6000
14150 ; ; *****
14151 ; ;
14152 062246 ; MUBRK:
14153 ; ; 6000:
14154 ; ; UBRK:
14155 ; ; R3_R3 PLUS 1,
14156 ; ; J/CHSVC11
14157 ; ;
14158 062246 130171 010001 116232 . WORD UWRD1, UWRD2, UWRD3
14159 ; ;
14160 ; ;
14161 ; ; 6171:
14162 ; ; CHSVC11:
14163 ; ; BUT(SERVICE),
14164 ; ; J/NOSVC11
14165 ; ;
```



```
14166 062254 020172 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14167
14168 ;
14169 ; 6172:
14170 ; NOSVC11:
14171 ; GOTO-PAGE(0),
14172 ; J/FETO1
14173
14174 062262 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14175
14176 ;
14177 ; 6173:
14178 ; SVC11:
14179 ; GOTO-PAGE(0),
14180 ; J/SERO2
14181
14182 062270 034703 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14183
14184 ;
14185 062276 125252 UWDEND
14186 ;
14187 ; *****
14188 ; RESERVED INSTRUCTION ENTRY INTO WCS AT 6003
14189 ; *****
14190 ;
14191 062300 MRESIN:
14192 ; 6003:
14193 ; URESIN:
14194 ; R3_MD,
14195 ; J/CHSVC12
14196
14197 062300 130201 010001 127032 . WORD UWRD1, UWRD2, UWRD3
14198
14199 ;
14200 ; 6201:
14201 ; CHSVC12:
14202 ; BUT(SERVICE),
14203 ; J/NOSVC12
14204
14205 062306 020202 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14206
14207 ;
14208 ; 6202:
14209 ;
14210 ; NOSVC12:
14211 ; GOTO-PAGE(0),
14212 ; J/FETO1
14213
14214 062314 034702 000000 000000 . WORD UWRD1, UWRD2, UWRD3
14215
14216 ;
14217 ; 6203:
14218 ; SVC12:
14219 ; GOTO-PAGE(0),
14220 ; J/SERO2
14221
```

```
14222 062322 034703 000000 000000 . WORD UWRD1,UWRD2,UWRD3
14223
14224
14225 062330 125252 ; UWDEND
14226 ;
14227 ;
14228 ; *****
14229 ; FLAG<7> WCS-SERVICE ENTRY INTO WCS AT 6005
14230 ; *****
14231 ;
14232 062332 MF7SVC:
14233 ; 6005:
14234 ; F7SVC:
14235 ; RETURN_FETO1, ;. RETURN W/O SERVICE CHECK
14236 ; GOTO-PAGE(6),
14237 ; J/770
14238
14239 062332 034770 100000 001605 . WORD UWRD1,UWRD2,UWRD3
14240
14241 ;
14242 ; 6770:
14243 ; R3_R3 PLUS 1,
14244 ; BUT(RETURN)
14245
14246 062340 137000 010001 116232 . WORD UWRD1,UWRD2,UWRD3
14247
14248 ;
14249 062346 125252 ; UWDEND
14250 ;
14251 ;
14252 ; *****
14253 ; END OF MICROCODE
```



PDP-11/60 WCS DIAGNOSTIC  
DQKUA.MIC 24-OCT-77 14:22

MACY11 30(1046) 25-OCT-77 20:12<sup>E 6</sup> PAGE 264  
MICRO-CODE

SEQ 0265  
SEQ 0276

14254

000001

.END





ATMP1	003044	1072#	1536*	1598*	1686*	1708								
AUNIT =	000000	235	242											
AUSWR =	000000	235	249											
AVECT1=	000000	235	274											
AVECT2=	000000	235	275											
BADPAR	003344	1261#	1352	2072	2278	2329	2442	2550	2658	2733	2734	2737	2814	2815
		2819	2896	2897	2901	2999	3043	3173	3217	3370	3389	3396	3518	3536
		3543	3663	3679	3686	3813	3830	3837	3961	3976	3983	4108	4124	4130
		4621	4636	4703	4705	4774	4795	4872	4893	4971	4992	5064	5085	5157
		5178	5255	5276	5353	5374	5441	5463	5546	5559	5634	5647	5735	5807
		5827	5847	5930	6080	6237	6369	6515	6650	6785	6913	7045	7163	7274
		7392	7510	7628	7749	7874	7999	8123	8245	8367	8510	8683	8849	9033
		9244	9452	9614	9781	9893	10850							
BADTMO	003276	1234#	1351	1381	1388	5449	5472	9892						
BITPR	037036	10018#	10078											
BIT0 =	000001	110#	1036	1355	2708	2710	2788	2790	2870	2873	10126			
BIT00 =	000001	100#	110											
BIT01 =	000002	99#	109											
BIT02 =	000004	98#	108											
BIT03 =	000010	97#	107											
BIT04 =	000020	96#	106											
BIT05 =	000040	95#	105											
BIT06 =	000100	94#	104											
BIT07 =	000200	93#	103											
BIT08 =	000400	92#	102	10371										
BIT09 =	001000	91#	101	10379	10445									
BIT1 =	000002	109#	1035											
BIT10 =	002000	90#	2959	3133	4662	10325	10423							
BIT11 =	004000	89#	8811	9393	10325	10386								
BIT12 =	010000	88#	5784											
BIT13 =	020000	87#	5784	10430										
BIT14 =	040000	86#	1037	5618	5627	10357								
BIT15 =	100000	85#	1028	1264	4662	5528								
BIT2 =	000004	108#	1034											
BIT3 =	000010	107#	1033	10129										
BIT4 =	000020	106#	1032											
BIT5 =	000040	105#	1031	1262	1421									
BIT6 =	000100	104#	1030	1262										
BIT7 =	000200	103#	1029	1262	1427	5776	5850	8474	8504	8534	8535	8537	8538	8539
		8542	8543	8544	8545	8546	8547	8548	8549	8550	8551	8553	8557	8558
		8559	8560	8561	8562	8563	8564	8565	8650	8677	8707	8708	8710	8711
		8712	8715	8716	8717	8718	8719	8720	8721	8722	8723	8724	8725	8729
		8731	8732	8733	8734	8735	8736	8737	8738	8828	8877	8878	8879	8880
		8881	8882	8883	8884	8885	8886	8887	8888	9002	9213	9426	9579	9581
		9583	9585	9590	9591	9593	9594	9638	9639	9640	9641	9642	9643	9644
		9645	9747											
BIT8 =	000400	102#	10124											
BIT9 =	001000	101#	1050	5523	10127									
BPTVEC=	000014	117#												
CESTAT	037252	1548*	1574*	1642*	1668*	1758*	2149*	2381*	2489*	2597*	10024	10075	10084	10092#
CHKSTA	037152	1551	1577	1645	1671	1761	2152	2384	2492	2600	10068#			
CLRWCS	036624	3284	3429	3582	3729	3878	4024	9916#						
CMASK	037256	1549*	1575*	1643*	1669*	1759*	2150*	2382*	2490*	2598*	10073	10080	10094#	
COUNT	003052	1075#	3747*	3772*	3906*	3921*	4042*	4068*	4198*	4224*	4282*	4308*	4370*	4400*
		4497*	4526*											
COUNT0	003054	1076#	1862*	1878*	2949*	3035*	3050*	4199*	4251*	4283*	4335*	4371*	4424*	4498*

















MSG17	042652	1242	1276	2324	2432	2540	2648	10335	10928#												
MSG2	042264	1085	10875#																		
MSG20	042670	10931#																			
MSG21	042701	10182	10933#																		
MSG22	042725	3401	10937#																		
MSG23	042757	10195	10942#																		
MSG24	042766	10202	10944#																		
MSG25	042774	3548	3697	10945#																	
MSG26	043026	3845	3992	10950#																	
MSG27	043060	4139	10955#																		
MSG28	043112	1384	1713	10960#																	
MSG29	043134	1268	10963#																		
MSG3	042271	1086	10876#																		
MSG30	043172	1271	10968#																		
MSG31	043233	1238	1272	10002	10974#																
MSG4	042276	1087	10877#																		
MSG5	042303	1088	10878#																		
MSG6	042322	1089	10881#																		
MSG7	042345	1090	10885#																		
MSG8	042362	1091	10888#																		
MSG80	042403	1092	10891#																		
MSG81	042410	1093	10892#																		
MSG82	042415	1094	10893#																		
MSG83	042423	1095	10895#																		
MSG84	042431	1096	10897#																		
MSG85	042437	1097	10899#																		
MSG86	042445	1098	10901#																		
MSG9	042453	1099	10903#																		
MTMPLS	060626	9571	13047#																		
MUBRK	062246	5535	14152#																		
MWR IND	057554	7483	12268#																		
MWR2	057704	7252	12389#																		
MWSLS	061516	6772	13653#																		
MXFC1	061760	4759	13879#																		
MXFC11	062012	4857	13919#																		
MXFC12	062044	4956	13958#																		
MXFC13	062076	5049	13996#																		
MXFC14	062130	5142	14035#																		
MXFC15	062162	5240	14073#																		
MXFC2	061726	5338	13840#																		
NEWPAS	004054	1350#	9882																		
NXTST	003056	1077#	1243	1277	1370*	1418*	1462*	1533*	1634*	1739*	1853*	1943*	2032*	2135*							
		2368*	2476*	2584*	2678*	2759*	2840*	2928*	3103*	3278*	3428*	3581*	3728*	3876*							
		4023*	4150*	4197*	4280*	4369*	4495*	4609*	4643*	4754*	4852*	4951*	5044*	5137*							
		5235*	5333*	5424*	5518*	5609*	5703*	5770*	5904*	6041*	6202*	6320*	6481*	6600*							
		6753*	6866*	7019*	7133*	7248*	7362*	7479*	7597*	7711*	7837*	7962*	8087*	8209*							
		8331*	8455*	8631*	8805*	8958*	9170*	9387*	9564*	9716*	10339										
ODDPC =	006004	1060#	5432																		
OTHDIS=	006001	1051#	4760																		
OTH11 =	006011	1053#	4858																		
OTH12 =	006012	1054#	4957																		
OTH13 =	006013	1055#	5050																		
OTH14 =	006014	1056#	5143																		
OTH15 =	006015	1057#	5241																		
PARDIS=	000020	1032#	1440	1467	1469	1490	1491	1504	1543	1638	1744	1857	1950	2033							
		2142	2168	2172	2207	2268	2326	2375	2390	2439	2483	2498	2547	2591							





RESTAR 004006  
RESVEC= 000010  
RO =%000000

1339#	10864												
114#													
35#	1355*	1372*	1377	1378	1391	1421	1423	1427*	1431*	1432*	1433	1435	
1440	1443	1446	1448	1464*	1465*	1467*	1469	1471	1476	1478	1482	1484	
1490*	1491	1493	1497*	1498	1500	1504	1506	1511*	1512	1514	1740*	1741*	
1742*	1743*	1749*	1753	1755	1757	1763	1769*	1772	1774	1780	1785	1787	
1793	1797	1799	1805	1806	1807	1808	1809*	1813	1859*	1864	1877*	1952*	
1953*	1956	1971*	2038*	2045	2050	2063*	2068	2079	2137*	2147	2154	2200	
2223	2245	2256	2261	2280	2298	2369*	2379	2386	2402	2410	2413	2419	
2477*	2487	2494	2510	2518	2521	2527	2585*	2595	2602	2618	2626	2629	
2635	2682*	2690	2692	2697	2705*	2708*	2713*	2727	2763*	2771	2773	2778	
2785*	2788*	2793*	2808	2844*	2852	2854	2859	2867*	2870*	2876*	2890	2945*	
2958	3049*	3120*	3121*	3132	3210*	3223*	3286*	3288	3290*	3298*	3313	3325*	
3341	3379	3399*	3432*	3435	3437*	3446*	3461	3474*	3489	3526	3546*	3584*	
3587	3589*	3598*	3612	3622*	3636	3671	3689*	3691*	3693*	3694*	3732*	3735	
3737*	3745*	3760	3770*	3784	3821	3840*	3842*	3880*	3884	3886*	3894*	3909	
3919*	3933	3967	3986*	3988*	3989*	4027*	4030	4032*	4040*	4055	4066*	4080	
4116	4134*	4136*	4218*	4236*	4245*	4262*	4301*	4320*	4330*	4346*	4375*	4398	
4422	4434*	4501*	4524	4548	4559*	4662*	4671*	4673	4675*	4676	4691	4755*	
4786*	4853*	4884*	4952*	4983*	5045*	5076*	5138*	5169*	5236*	5267*	5334*	5365*	
5523*	5528*	5539*	5618*	5627*	5708*	5731*	5772*	5776*	5779	5811*	5812	5814	
5850*	5906*	5935*	5936	5941	5946*	5947	5951	5955*	5956	5961	5965*	5968	
5971	5973	5978*	6048*	6051*	6053*	6055*	6057*	6059*	6061*	6063*	6065*	6067*	
6083	6087	6212*	6243*	6247	6329*	6346*	6356*	6390	6397	6489*	6522*	6525	
6540*	6612*	6629*	6638*	6670	6676	6760*	6790*	6792	6875*	6892*	6901*	6930	
6936	7033*	7146*	7262*	7375*	7490*	7492*	7516*	7517	7518	7608*	7633*	7634	
7635	7724*	7732*	7733*	7734	7849*	7857*	7858*	7859	7974*	7982*	7983*	7984	
8099*	8109*	8110*	8111	8134	8138	8221*	8231*	8232*	8233	8256	8260	8343*	
8353*	8354*	8355	8378	8382	8472*	8474*	8475	8499	8502	8504*	8648*	8650*	
8651	8675	8677*	8811*	8826*	8828*	8829	8970*	8973	8975*	8983	8984	8988	
8989	9000*	9002*	9003	9004*	9036*	9059	9184*	9187	9189*	9199	9200	9211*	
9213*	9214	9215*	9247	9248*	9275	9393*	9413*	9415	9417*	9424*	9426*	9427	
9428*	9467	9469	9579*	9581*	9583*	9585*	9590*	9597*	9599*	9601*	9603*	9730*	
9732*	9733	9745*	9747*	9748	9749*	9763*	9772	9774	9776	9778	9794	9796	
9845*	9873*	9876	9916	9918*	9919	9921*	9922	9925*	10018	10028*	10030*	10033	
10040	10053*	10173	10203	10257	10264*	10266*	10267*	10268	10284*	10306*	10314*	10466	
10467*	10468*	10475*	10476*	10477*	10478*	10479*	10480	10485	10490*	10492*	10496	10498	
10594	10604*	10608	10624	10625	10638*	10669	10670*	10675	10680	10683*	10735	10743*	
10747	10748	10750*	10751*	10752	10774*	10792	10793*	10794	10795*	10796*	10797*	10798*	
10829	10848*	10858*											
36#	1373*	1379*	1805*	1809	2035*	2048	2066*	2138*	2149	2281	2370*	2381	
2422*	2478*	2489	2530*	2586*	2597	2638*	2685*	2689	2766*	2770	2847*	2851	
2934*	2957	2959	2978	2993	3003	3024	3045	3070*	3071	3109*	3131	3133	
3152	3167	3177	3199	3219	3245*	3246	3287*	3289	3291*	3292	3299*	3322	
3326*	3327	3342	3352	3380	3400*	3433*	3436	3438*	3439	3447*	3471	3475*	
3476	3490	3500	3527	3547*	3586*	3588	3590*	3591	3599*	3620	3623*	3624	
3637	3645	3672	3690*	3692*	3695*	3696*	3733*	3736	3738*	3746*	3768	3771*	
3786	3794	3822	3841*	3843*	3882*	3885	3887*	3895*	3917	3920*	3934	3942	
3968	3987*	3990*	3991*	4028*	4031	4033*	4041*	4064	4067*	4082	4089	4117	
4135*	4137*	4138*	4155*	4156	4157	4158*	4159	4162*	4163	4165	4169	4170	
4174*	4175	4207*	4211*	4219*	4221	4226	4235*	4244*	4247	4253	4261*	4290*	
4294*	4302*	4304	4310	4319*	4329*	4332	4337	4345*	4380*	4382	4384*	4393*	
4395	4402	4407*	4416*	4419	4426	4430*	4506*	4508	4510*	4519*	4521	4528	
4532*	4542*	4545	4552	4556*	4649*	4650	4652*	4653	4674*	4756*	4762	4775	
4778	4785*	4798	4854*	4860	4873	4876	4883*	4896	4953*	4959	4972	4975	
4982*	4995	5046*	5052	5065	5068	5075*	5088	5139*	5145	5158	5161	5168*	

R1 =%000001



5181	5237*	5243	5256	5259	5266*	5279	5335*	5341	5354	5357	5364*	5375
5709*	5715	5722	5726	5730*	5779*	5780*	5781	5782*	5783*	5784*	5785	5907*
5967*	5968	5972	6047*	6049*	6052*	6054*	6056*	6058*	6060*	6062*	6064*	6066*
6091	6095	6204*	6217*	6245*	6249	6256	6263*	6325*	6327	6347*	6351	6355*
6374*	6415*	6483*	6495*	6523*	6527	6534	6542*	6606*	6609	6630*	6634	6637*
6655*	6692*	6756*	6765*	6791*	6793	6800	6807*	6871*	6873	6893*	6897	6900*
6918*	6951*	7034*	7147*	7263*	7376*	7493*	7520*	7521	7522	7609*	7637*	7638
7639	7757*	7758*	7760	7764	7882*	7883*	7885	7889	8007*	8008*	8010	8014
8129*	8130*	8131	8134	8137	8251*	8252*	8253	8256	8259	8373*	8374*	8375
8378	8381	8971*	8976*	9037*	9185*	9190*	9249	9250*	9459*	9460	9462	9471
9591*	9597	9604*	9764*	9772*	10019	10023*	10026*	10033	10052*	10068	10071*	10072
10073*	10074	10075	10080*	10081*	10083	10089*	10174	10196	10258	10263*	10264	10265
10269*	10270	10272	10274*	10275	10279*	10308	10313*	10595	10608*	10609	10613	10637*
10736	10773*	10830	10857*									
37#	1375*	1382	1390*	1806*	1807*	1808*	1875	1886	1969	1980	2044*	2060
2062*	2139*	2259	2283*	2371*	2424*	2479*	2532*	2587*	2640*	2691*	2692	2698
2710*	2711	2772*	2773	2779	2790*	2791	2853*	2854	2860	2873*	2874	2942*
2943	2969	2971*	2972	3117*	3118	3143	3145*	3146	3244*	3314*	3315	3316*
3318	3320*	3322	3343	3354	3359	3361*	3363	3462*	3464	3465*	3467	3469*
3471	3491	3502	3508	3509*	3511	3613*	3614	3615*	3617	3619*	3620	3638
3647	3653	3655*	3657	3734*	3739*	3761*	3762	3763*	3765	3767*	3768	3787
3796	3802	3804*	3806	3883*	3888*	3910*	3911	3912*	3914	3916*	3917	3935
3944	3950	3952*	3954	4029*	4034*	4056*	4057	4059*	4061	4063*	4064	4083
4091	4097	4099*	4101	4164*	4165	4171	4222*	4228	4248*	4249	4255	4305*
4306	4312	4333*	4339	4396*	4420*	4441*	4442*	4443*	4444	4458*	4459*	4460*
4461	4522*	4546*	4566*	4567*	4568*	4569	4583*	4584*	4585*	4586	4611*	4616*
4644*	4688	5706*	5738*	5908*	6099	6103	6206*	6210	6247*	6251	6257	6259
6324*	6333*	6349*	6352	6375*	6380	6485*	6487	6525*	6529	6535	6605*	6616*
6632*	6635	6636*	6656*	6664	6757*	6758	6760	6792*	6795	6801	6803*	6804
6870*	6879*	6895*	6898	6919*	6924	7032*	7149*	7150	7261*	7378*	7379	7497*
7498	7613*	7614	7739*	7754	7763	7768*	7864*	7879	7888	7893*	7989*	8004
8013	8018*	8107*	8108	8127*	8229*	8230	8249*	8351*	8352	8371*	8471*	8472
8494*	8647*	8648	8670*	8825*	8826	8843*	8999*	9000	9023*	9038*	9210*	9211
9234*	9251	9252*	9423*	9424	9442*	9461*	9467	9470	9592*	9744*	9745	9762*
9774*	9948*	9951*	9954*	10020	10024*	10026	10040	10051*	10186*	10187*	10190	10192
10259	10281*	10289	10295	10296	10299	10300	10302	10312*	10327	10596	10607*	10611*
10614	10621*	10622*	10623	10628*	10636*	10831	10856*					
38#	1745*	1758	1810	1886*	1888*	1889*	1891	1980*	1982*	1983*	1986	2039*
2070*	2140*	2197	2242	2285*	2680*	2703*	2761*	2784*	2842*	2865*	2964*	2965*
2966	3138*	3139*	3140	3308*	3358*	3397	3506*	3544	3651*	3687	3800*	3838
3948*	3984	4025*	4095*	4132	4154*	4167*	4203*	4233	4249	4254	4286*	4292
4306	4311	4344	4373*	4405	4499*	4509	4555	4775	4779	4873	4877	4972
4976	5065	5069	5158	5162	5256	5260	5354	5358	5425*	5442	5444	5519*
5547	5549	5610*	5635	5637	5722	5727	5909*	6107	6111	6207*	6212	6250*
6251	6258	6352*	6353*	6354	6376*	6386	6486*	6489	6494*	6528*	6529	6536
6659*	6673	6794*	6795	6802	7030*	7144*	7166	7170	7259*	7373*	7395	7399
7494*	7610*	7631	7641	7755*	7760	7765	7880*	7885	7890	8005*	8010	8015
9039*	9253	9254*	9593*	9599	9605*	9765*	9776*	9940	9941*	9944	9945*	9950*
9952*	9956*	9979	9983*	9984*	9985*	9986*	9989	9990*	10021	10027*	10032*	10035*
10036	10037*	10050*	10115	10117*	10118*	10119	10120*	10123*	10126*	10129*	10131	10133*
10260	10289*	10290	10311*	10537	10546*	10552*	10553*	10556*	10561*	10562*	10563	10572*
10597	10605*	10606*	10620*	10623*	10632*	10633*	10635*	10832	10855*			
39#	1747*	1750*	1754*	1756*	1770*	1783*	1795*	1858*	1859	1864*	1867	1890
1951*	1952	1956*	1960	1984	2037*	2066	2681*	2706	2711*	2762*	2786	2791*
2843*	2868	2874*	2944*	2945	2958*	2962	2979	3004	3019	3025	3046	3119*
3120	3132*	3136	3153	3178	3193	3200	3220	3300*	3315*	3448*	3464*	3600*

R2 =%000002

R3 =%000003

R4 =%000004



3614*	3748*	3762*	3896*	3911*	4043*	4057*	4151*	4156*	4163*	4201*	4208*	4210*
4221*	4247*	4284*	4291*	4293*	4304*	4332*	4381*	4385*	4392*	4408*	4417*	4431*
4507*	4511*	4518*	4533*	4543*	4557*	5705*	5706	5724*	5771*	5840*	5841	5910*
6115	6119	6241*	6264*	6386*	6387*	6390	6393	6394*	6395	6519*	6544*	6657*
6670	6675	6691*	6789*	6809*	6924*	6925*	6926	6930	6934	6935	7031*	7145*
7174	7178	7260*	7374*	7403	7407	7495*	7611*	7721*	7727*	7731*	7737*	7752*
7769*	7846*	7852*	7856*	7862*	7877*	7894*	7971*	7977*	7981*	7987*	8002*	8019*
8096*	8102*	8106*	8113*	8126*	8141*	8218*	8224*	8228*	8235*	8248*	8263*	8340*
8346*	8350*	8357*	8370*	8385*	8514*	8528*	8687*	8701*	8853*	8867*	9040*	9255
9256*	9412*	9418*	9456*	9475*	9594*	9601	9606*	9618*	9631*	9766*	9778*	9947
9949*	9955*	9980	9983	9987*	9988*	9989*	10117	10124	10127	10130*	10131*	10173*
10174*	10175*	10176*	10178	10538	10540*	10541*	10542*	10543	10544*	10558	10560*	10568*
10571*	10833	10854*										
40#	1748*	1753*	1755*	1757*	1771	1784	1796	1868*	1870*	1875	1961*	1963*
1969	2046*	2047*	2048	2195*	2196*	2197	2201*	2240*	2241*	2242	2246*	2257*
2258*	2259	2282*	2283	2284*	2285	2393*	2410	2414	2421*	2422	2501*	2518
2522	2529*	2530	2609*	2626	2630	2637*	2638	2684*	2709*	2765*	2789*	2846*
2872*	2963*	2966*	2972	2994*	3019	3026	3053*	3054	3137*	3140*	3146	3168*
3193	3201	3227*	3228	3302*	3306	3317*	3318*	3319*	3450*	3454	3466*	3467*
3468*	3602*	3606	3616*	3617*	3618*	3750*	3754	3764*	3765*	3766*	3898*	3902
3913*	3914*	3915*	4045*	4049	4060*	4061*	4062*	4152*	4157*	4164	4202*	4209*
4222	4233*	4248	4260*	4285*	4292*	4305	4318*	4333	4344*	4397*	4398	4403
4421*	4422	4427	4523*	4524	4529	4547*	4548	4553	4757*	4855*	4954*	5047*
5140*	5238*	5336*	5429*	5533*	5612*	5710*	5773*	5792	5800	5808*	5809	5816*
5830	5911*	5914*	6042*	6123	6127	6222*	6339*	6378*	6398	6414*	6500*	6520*
6537	6541*	6622*	6660*	6677	6690*	6770*	6885*	6920*	6937	6950*	7021*	7035*
7135*	7148*	7250*	7264*	7364*	7377*	7481*	7496*	7599*	7612*	7714*	7720*	7722
7730*	7732	7751*	7757	7762	7839*	7845*	7847	7855*	7857	7876*	7882	7887
7964*	7970*	7972	7980*	7982	8001*	8007	8012	8089*	8095*	8097	8105*	8109
8125*	8129	8136	8211*	8217*	8219	8227*	8231	8247*	8251	8258	8333*	8339*
8341	8349*	8353	8369*	8373	8380	8460*	8513*	8519	8636*	8686*	8692	8814*
8852*	8859	8963*	9041*	9175*	9257	9258*	9396*	9455*	9459	9461	9569*	9607*
9617*	9622	9623	9721*	9728*	9730	9785*	9789	9790	9978	9980*	9981*	9982*
9988	9991*	10069	10082*	10083*	10084*	10085	10088*	10116	10119*	10121*	10122*	10123
10132*	10178*	10188*	10279	10281	10283	10320*	10339*	10539	10545*	10547*	10549*	10550*
10551*	10552	10570*	10598	10600*	10602*	10609*	10613*	10628	10634*	10834	10853*	
41#	1289*	1290*	1291	1339*	1340*	1341	5922*	5924*	6074*	6081	6082*	6132
42#												
43#	1234	1236*	1239*	1240*	1243*	1261	1273*	1274*	1277*	1281*	1293*	1311*
1319*	1323	1343*	1350*	1387	1600*	1601	1602	1688*	1689	1690	2186	2293
2294	2310	2311	2320	2321	2429	2430	2435	2537	2538	2543	2645	2646
2651	2720	2800	2885	3042	3216	3352*	3353*	3354*	3355*	3356	3374	3401*
3402*	3500*	3501*	3502*	3503*	3504	3521	3548*	3549*	3645*	3646*	3647*	3648*
3649	3667	3697*	3698*	3794*	3795*	3796*	3797*	3798	3816	3845*	3846*	3942*
3943*	3944*	3945*	3946	3964	3992*	3993*	4089*	4090*	4091*	4092*	4093	4111
4139*	4140*	4633	4686	4793	4891	4990	5083	5176	5274	5372	5450	5451
5464	5560	5648	5734	5748	5804	5826	5987	6153	6269	6426	6549	6700
6814	6959	7074	7188	7303	7417	7534	7651	7777	7902	8027	8149	8271
8393	8496	8572	8672	8745	8894	8978	8992	8993	9042*	9068	9111	9180
9203	9204	9247*	9248	9249*	9250	9251*	9252	9253*	9254	9255*	9256	9257*
9258	9284	9327	9502	9651	9822	9870*	9907	9910*	9916*	9925	9940*	9944*
9947*	9955	9956	9978*	9979*	9990	9991	10003*	10004*	10018*	10019*	10020*	10021*
10050	10051	10052	10053	10068*	10069*	10087*	10088	10089	10115*	10116*	10132	10133
10176	10181	10190*	10196*	10197*	10203*	10204*	10209*	10210	10257*	10258*	10259*	10260*
10311	10312	10313	10314	10336*	10362*	10365	10367	10368	10397	10398	10402*	10427
10447*	10450*	10466*	10471*	10492	10496*	10529*	10530	10531	10532*	10537*	10538*	10539*

R5 =%000005

R6 =%000006

R7 =%000007

SP =%000006

	10545	10570	10571	10572	10573*	10574*	10594*	10595*	10596*	10597*	10598*	10599*	10600
	10603*	10616	10618*	10620	10630	10632	10634	10635	10636	10637	10638	10640*	10641*
	10669*	10670	10680*	10682	10683	10684*	10686	10688	10690	10696	10698*	10700*	10708*
	10712	10716	10717	10721	10735*	10736*	10743	10744*	10755	10756*	10757*	10767	10768*
	10773	10774	10792*	10793	10803*	10804*	10829*	10830*	10831*	10832*	10833*	10834*	10835*
	10836	10844*	10848	10852	10853	10854	10855	10856	10857	10858	10863*		
STACK = 001100	18#	1293	1343	1350									
START 003450	136	1289#											
STKMT= 177774	29#												
STWRDO 006044	1745	1825#											
SWB01 = 000071	1048#	5520											
SWR 001140	197#	1291	1313*	1315	1321*	1328*	1341	9905	9998	10357	10371	10373	10379
	10386	10423	10430	10442	10445	10835	10852*						
SWPEG 000176	134#	1321											
SW0 = 000001	82#												
SW00 = 000001	72#	82											
SW01 = 000002	71#	81											
SW02 = 000004	70#	80											
SW03 = 000010	69#	79											
SW04 = 000020	68#	78											
SW05 = 000040	67#	77											
SW06 = 000100	66#	76											
SW07 = 000200	65#	75											
SW08 = 000400	64#	74											
SW09 = 001000	63#	73											
SW1 = 010000	81#	12123#	12130#	12138#	12145#	12152#	12160#	12178#	12185#	2192#	12199#	12206#	12224#
	12231#	12240#	12247#	12255#	12273#	12280#	12288#	12295#	12302	2318#	12325#	12333#	12340#
	12347#	12353#	12360#	12367#	12375#	12394#	12401#	12409#	12416#	12423#	12430#	12437#	12443#
	12461#	12468#	12476#	12483#	12489#	12507#	12514#	12522#	12529#	12535#	12552#	12559#	12567#
	12574#	12580#	12597#	12604#	12612#	12619#	12625#	12643#	12650#	12658#	12665#	12671#	12688#
	12695#	12703#	12710#	12716#	12733#	12741#	12749#	12756#	12762#	12779#	12786#	12794#	12801#
	12807#	12824#	12831#	12839#	12847#	12853#	12870#	12877#	12885#	12892#	12898#	12915#	12922#
	12930#	12937#	12944#	12961#	12968#	12976#	12983#	12989#	13006#	13013#	13021#	13029#	13035#
	13052#	13059#	13067#	13074#	13080#	13095#	13104#	13113#	13121#	13129#	13136#	13143#	13150#
	13158#	13166#	13174#	13188#	13198#	13207#	13215#	13223#	13230#	13237#	13244#	13252#	13260#
	13268#	13282#	13291#	13300#	13308#	13316#	13323#	13330#	13337#	13345#	13353#	13362#	13376#
	13385#	13394#	13402#	13410#	13417#	13424#	13431#	13439#	13447#	13455#	13469#	13478#	13487#
	13495#	13503#	13511#	13518#	13525#	13533#	13541#	13549#	13563#	13572#	13581#	13589#	13597#
	13604#	13611#	13618#	13626#	13634#	13642#	13657#	13666#	13675#	13683#	13691#	13698#	13705#
	13712#	13720#	13728#	13736#	13750#	13759#	13768#	13776#	13785#	13792#	13799#	13806#	13814#
	13822#	13830#	13845#	13853#	13861#	13869#	13884#	13892#	13901#	13909#	13924#	13932#	13940#
	13948#	13963#	13971#	13979#	13987#	14001#	14010#	14018#	14026#	14040#	14048#	14056#	14064#
	14078#	14086#	14094#	14102#	14118#	14126#	14134#	14142#	14157#	14165#	14173#	14181#	14196#
	14204#	14213#	14221#	14238#	14245#								
SW10 = 002000	62#												
SW11 = 004000	61#												
SW12 = 010000	60#												
SW13 = 020000	59#	9905	9998										
SW14 = 040000	58#												
SW15 = 100000	57#												
SW2 = 000001	80#	12123#	12130#	12138#	12145#	12152#	12160#	12178#	12185#	12192#	12199#	12206#	12224#
	12231#	12240#	12247#	12255#	12273#	12280#	12288#	12295#	12302#	12318#	12325#	12333#	12340#
	12347#	12353#	12360#	12367#	12375#	12394#	12401#	12409#	12416#	12423#	12430#	12437#	12443#
	12461#	12468#	12476#	12483#	12489#	12507#	12514#	12522#	12529#	12535#	12552#	12559#	12567#
	12574#	12580#	12597#	12604#	12612#	12619#	12625#	12643#	12650#	12658#	12665#	12671#	12688#
	12695#	12703#	12710#	12716#	12733#	12741#	12749#	12756#	12762#	12779#	12786#	12794#	12801#









12240#	12247#	12255#	12273#	12280#	12288#	12295#	12302#	12318#	12325#	12333#	12340#	12347#
12353#	12360#	12367#	12375#	12394#	12401#	12409#	12416#	12423#	12430#	12437#	12443#	12461#
12468#	12476#	12483#	12489#	12507#	12514#	12522#	12529#	12535#	12552#	12559#	12567#	12574#
12580#	12597#	12604#	12612#	12619#	12625#	12643#	12650#	12658#	12665#	12671#	12688#	12695#
12703#	12710#	12716#	12733#	12741#	12749#	12756#	12762#	12779#	12786#	12794#	12801#	12807#
12824#	12831#	12839#	12847#	12853#	12870#	12877#	12885#	12892#	12898#	12915#	12922#	12930#
12937#	12944#	12961#	12968#	12976#	12983#	12989#	13006#	13013#	13021#	13029#	13035#	13052#
13059#	13067#	13074#	13080#	13095#	13104#	13113#	13121#	13129#	13136#	13143#	13150#	13158#
13166#	13174#	13188#	13198#	13207#	13215#	13223#	13230#	13237#	13244#	13252#	13260#	13268#
13282#	13291#	13300#	13308#	13316#	13323#	13330#	13337#	13345#	13353#	13362#	13376#	13385#
13394#	13402#	13410#	13417#	13424#	13431#	13439#	13447#	13455#	13469#	13478#	13487#	13495#
13503#	13511#	13518#	13525#	13533#	13541#	13549#	13563#	13572#	13581#	13589#	13597#	13604#
13611#	13618#	13626#	13634#	13642#	13657#	13666#	13675#	13683#	13691#	13698#	13705#	13712#
13720#	13728#	13736#	13750#	13759#	13768#	13776#	13785#	13792#	13799#	13806#	13814#	13822#
13830#	13845#	13853#	13861#	13869#	13884#	13892#	13901#	13909#	13924#	13932#	13940#	13948#
13963#	13971#	13979#	13987#	14001#	14010#	14018#	14026#	14040#	14048#	14056#	14064#	14078#
14086#	14094#	14102#	14118#	14126#	14134#	14142#	14157#	14165#	14173#	14181#	14196#	14204#
14213#	14221#	14238#	14245#									
9871	10191	10819#										
1235	1238	1242	1268	1271	1272	1276	1332	1384	1713	9869	9872	9908
10001	10002	10335	10425	10433	10465	10482	10484	10487	10489	10493	10500	10564
10639	10691	10815#	10861									
1592	1594	1597	1680	1682	1685	2324	2432	2540	2648	10038	10043	10046
10182	10183	10195	10202	10922#								
1237	1241	1275	10005	10337	10473	10497	10816#					
10818#												
10198	10205	10817#										
9905#	10822											
1061#	5536											
1059#	5339	5917	6045	6225	6342	6503	6625	6773	6888	7024	7138	7253
7367	7484	7602	7717	7842	7967	8092	8214	8336	8463	8639	8817	8966
9178	9399	9572	9724									
1058#	5713											
1063#	10302	12164	12210	12259	12306	12380	12447	12493	12539	12584	12630	12675
12720	12766	12811	12857	12902	12948	12993	13039	13084	13178	13272	13366	13459
13553	13646	13740	13834	13873	13913	13952	13991	14030	14068	14106	14146	14185
14225	14249											
12123#	12124	12130#	12131	12138#	12139	12145#	12146	12152#	12153	12160#	12161	12178#
12179	12185#	12186	12192#	12193	12199#	12200	12206#	12207	12224#	12225	12231#	12232
12240#	12241	12247#	12248	12255#	12256	12273#	12274	12280#	12281	12288#	12289	12295#
12296	12302#	12303	12318#	12319	12325#	12326	12333#	12334	12340#	12341	12347#	12348
12353#	12354	12360#	12361	12367#	12368	12375#	12376	12394#	12395	12401#	12402	12409#
12410	12416#	12417	12423#	12424	12430#	12431	12437#	12438	12443#	12444	12461#	12462
12468#	12469	12476#	12477	12483#	12484	12489#	12490	12507#	12508	12514#	12515	12522#
12523	12529#	12530	12535#	12536	12552#	12553	12559#	12560	12567#	12568	12574#	12575
12580#	12581	12597#	12598	12604#	12605	12612#	12613	12619#	12620	12625#	12626	12643#
12644	12650#	12651	12658#	12659	12665#	12666	12671#	12672	12688#	12689	12695#	12696
12703#	12704	12710#	12711	12716#	12717	12733#	12734	12741#	12742	12749#	12750	12756#
12757	12762#	12763	12779#	12780	12786#	12787	12794#	12795	12801#	12802	12807#	12808
12824#	12825	12831#	12832	12839#	12840	12847#	12848	12853#	12854	12870#	12871	12877#
12878	12885#	12886	12892#	12893	12898#	12899	12915#	12916	12922#	12923	12930#	12931
12937#	12938	12944#	12945	12961#	12962	12968#	12969	12976#	12977	12983#	12984	12989#
12990	13006#	13007	13013#	13014	13021#	13022	13029#	13030	13035#	13036	13052#	13053
13059#	13060	13067#	13068	13074#	13075	13080#	13081	13095#	13096	13104#	13105	13113#
13114	13121#	13122	13129#	13130	13136#	13137	13143#	13144	13150#	13151	13158#	13159
13166#	13167	13174#	13175	13188#	13189	13198#	13199	13207#	13208	13215#	13216	13223#

TYPDS = 104405  
TYPE = 104401

TYPMSG= 104406

TYPOC = 104402  
TYPON = 104404  
TYPOS = 104403  
TY. MSG 036574  
UBRK = 006000  
UDISP = 006002

URESIN= 006003  
UWDEND= 125252

UWRD1 = 137000

13224	13230#	13231	13237#	13238	13244#	13245	13252#	13253	13260#	13261	13268#	13269
13282#	13283	13291#	13292	13300#	13301	13308#	13309	13316#	13317	13323#	13324	13330#
13331	13337#	13338	13345#	13346	13353#	13354	13362#	13363	13376#	13377	13385#	13386
13394#	13395	13402#	13403	13410#	13411	13417#	13418	13424#	13425	13431#	13432	13439#
13440	13447#	13448	13455#	13456	13469#	13470	13478#	13479	13487#	13488	13495#	13496
13503#	13504	13511#	13512	13518#	13519	13525#	13526	13533#	13534	13541#	13542	13549#
13550	13563#	13564	13572#	13573	13581#	13582	13589#	13590	13597#	13598	13604#	13605
13611#	13612	13618#	13619	13626#	13627	13634#	13635	13642#	13643	13657#	13658	13666#
13667	13675#	13676	13683#	13684	13691#	13692	13698#	13699	13705#	13706	13712#	13713
13720#	13721	13728#	13729	13736#	13737	13750#	13751	13759#	13760	13768#	13769	13776#
13777	13785#	13786	13792#	13793	13799#	13800	13806#	13807	13814#	13815	13822#	13823
13830#	13831	13845#	13846	13853#	13854	13861#	13862	13869#	13870	13884#	13885	13892#
13893	13901#	13902	13909#	13910	13924#	13925	13932#	13933	13940#	13941	13948#	13949
13963#	13964	13971#	13972	13979#	13980	13987#	13988	14001#	14002	14010#	14011	14018#
14019	14026#	14027	14040#	14041	14048#	14049	14056#	14057	14064#	14065	14078#	14079
14086#	14087	14094#	14095	14102#	14103	14118#	14119	14126#	14127	14134#	14135	14142#
14143	14157#	14158	14165#	14166	14173#	14174	14181#	14182	14196#	14197	14204#	14205
14213#	14214	14221#	14222	14238#	14239	14245#	14246					
12123#	12124	12130#	12131	12138#	12139	12145#	12146	12152#	12153	12160#	12161	12178#
12179	12185#	12186	12192#	12193	12199#	12200	12206#	12207	12224#	12225	12231#	12232
12240#	12241	12247#	12248	12255#	12256	12273#	12274	12280#	12281	12288#	12289	12295#
12296	12302#	12303	12318#	12319	12325#	12326	12333#	12334	12340#	12341	12347#	12348
12353#	12354	12360#	12361	12367#	12368	12375#	12376	12394#	12395	12401#	12402	12409#
12410	12416#	12417	12423#	12424	12430#	12431	12437#	12438	12443#	12444	12461#	12462
12468#	12469	12476#	12477	12483#	12484	12489#	12490	12507#	12508	12514#	12515	12522#
12523	12529#	12530	12535#	12536	12552#	12553	12559#	12560	12567#	12568	12574#	12575
12580#	12581	12597#	12598	12604#	12605	12612#	12613	12619#	12620	12625#	12626	12643#
12644	12650#	12651	12658#	12659	12665#	12666	12671#	12672	12688#	12689	12695#	12696
12703#	12704	12710#	12711	12716#	12717	12733#	12734	12741#	12742	12749#	12750	12756#
12757	12762#	12763	12779#	12780	12786#	12787	12794#	12795	12801#	12802	12807#	12808
12824#	12825	12831#	12832	12839#	12840	12847#	12848	12853#	12854	12870#	12871	12877#
12878	12885#	12886	12892#	12893	12898#	12899	12915#	12916	12922#	12923	12930#	12931
12937#	12938	12944#	12945	12961#	12962	12968#	12969	12976#	12977	12983#	12984	12989#
12990	13006#	13007	13013#	13014	13021#	13022	13029#	13030	13035#	13036	13052#	13053
13059#	13060	13067#	13068	13074#	13075	13080#	13081	13095#	13096	13104#	13105	13113#
13114	13121#	13122	13129#	13130	13136#	13137	13143#	13144	13150#	13151	13158#	13159
13166#	13167	13174#	13175	13188#	13189	13198#	13199	13207#	13208	13215#	13216	13223#
13224	13230#	13231	13237#	13238	13244#	13245	13252#	13253	13260#	13261	13268#	13269
13282#	13283	13291#	13292	13300#	13301	13308#	13309	13316#	13317	13323#	13324	13330#
13331	13337#	13338	13345#	13346	13353#	13354	13362#	13363	13376#	13377	13385#	13386
13394#	13395	13402#	13403	13410#	13411	13417#	13418	13424#	13425	13431#	13432	13439#
13440	13447#	13448	13455#	13456	13469#	13470	13478#	13479	13487#	13488	13495#	13496
13503#	13504	13511#	13512	13518#	13519	13525#	13526	13533#	13534	13541#	13542	13549#
13550	13563#	13564	13572#	13573	13581#	13582	13589#	13590	13597#	13598	13604#	13605
13611#	13612	13618#	13619	13626#	13627	13634#	13635	13642#	13643	13657#	13658	13666#
13667	13675#	13676	13683#	13684	13691#	13692	13698#	13699	13705#	13706	13712#	13713
13720#	13721	13728#	13729	13736#	13737	13750#	13751	13759#	13760	13768#	13769	13776#
13777	13785#	13786	13792#	13793	13799#	13800	13806#	13807	13814#	13815	13822#	13823
13830#	13831	13845#	13846	13853#	13854	13861#	13862	13869#	13870	13884#	13885	13892#
13893	13901#	13902	13909#	13910	13924#	13925	13932#	13933	13940#	13941	13948#	13949
13963#	13964	13971#	13972	13979#	13980	13987#	13988	14001#	14002	14010#	14011	14018#
14019	14026#	14027	14040#	14041	14048#	14049	14056#	14057	14064#	14065	14078#	14079
14086#	14087	14094#	14095	14102#	14103	14118#	14119	14126#	14127	14134#	14135	14142#
14143	14157#	14158	14165#	14166	14173#	14174	14181#	14182	14196#	14197	14204#	14205
14213#	14214	14221#	14222	14238#	14239	14245#	14246					
12123#	12124	12130#	12131	12138#	12139	12145#	12146	12152#	12153	12160#	12161	12178#

UWRD2 = 010001

UWRD3 = 116232



12179	12185#	12186	12192#	12193	12199#	12200	12206#	12207	12224#	12225	12231#	12232
12240#	12241	12247#	12248	12255#	12256	12273#	12274	12280#	12281	12288#	12289	12295#
12296	12302#	12303	12318#	12319	12325#	12326	12333#	12334	12340#	12341	12347#	12348
12353#	12354	12360#	12361	12367#	12368	12375#	12376	12394#	12395	12401#	12402	12409#
12410	12416#	12417	12423#	12424	12430#	12431	12437#	12438	12443#	12444	12461#	12462
12468#	12469	12476#	12477	12483#	12484	12489#	12490	12507#	12508	12514#	12515	12522#
12523	12529#	12530	12535#	12536	12552#	12553	12559#	12560	12567#	12568	12574#	12575
12580#	12581	12597#	12598	12604#	12605	12612#	12613	12619#	12620	12625#	12626	12643#
12644	12650#	12651	12658#	12659	12665#	12666	12671#	12672	12688#	12689	12695#	12696
12703#	12704	12710#	12711	12716#	12717	12733#	12734	12741#	12742	12749#	12750	12756#
12757	12762#	12763	12779#	12780	12786#	12787	12794#	12795	12801#	12802	12807#	12808
12824#	12825	12831#	12832	12839#	12840	12847#	12848	12853#	12854	12870#	12871	12877#
12878	12885#	12886	12892#	12893	12898#	12899	12915#	12916	12922#	12923	12930#	12931
12937#	12938	12944#	12945	12961#	12962	12968#	12969	12976#	12977	12983#	12984	12989#
12990	13006#	13007	13013#	13014	13021#	13022	13029#	13030	13035#	13036	13052#	13053
13059#	13060	13067#	13068	13074#	13075	13080#	13081	13095#	13096	13104#	13105	13113#
13114	13121#	13122	13129#	13130	13136#	13137	13143#	13144	13150#	13151	13158#	13159
13166#	13167	13174#	13175	13188#	13189	13198#	13199	13207#	13208	13215#	13216	13223#
13224	13230#	13231	13237#	13238	13244#	13245	13252#	13253	13260#	13261	13268#	13269
13282#	13283	13291#	13292	13300#	13301	13308#	13309	13316#	13317	13323#	13324	13330#
13331	13337#	13338	13345#	13346	13353#	13354	13362#	13363	13376#	13377	13385#	13386
13394#	13395	13402#	13403	13410#	13411	13417#	13418	13424#	13425	13431#	13432	13439#
13440	13447#	13448	13455#	13456	13469#	13470	13478#	13479	13487#	13488	13495#	13496
13503#	13504	13511#	13512	13518#	13519	13525#	13526	13533#	13534	13541#	13542	13549#
13550	13563#	13564	13572#	13573	13581#	13582	13589#	13590	13597#	13598	13604#	13605
13611#	13612	13618#	13619	13626#	13627	13634#	13635	13642#	13643	13657#	13658	13666#
13667	13675#	13676	13683#	13684	13691#	13692	13698#	13699	13705#	13706	13712#	13713
13720#	13721	13728#	13729	13736#	13737	13750#	13751	13759#	13760	13768#	13769	13776#
13777	13785#	13786	13792#	13793	13799#	13800	13806#	13807	13814#	13815	13822#	13823
13830#	13831	13845#	13846	13853#	13854	13861#	13862	13869#	13870	13884#	13885	13892#
13893	13901#	13902	13909#	13910	13924#	13925	13932#	13933	13940#	13941	13948#	13949
13963#	13964	13971#	13972	13979#	13980	13987#	13988	14001#	14002	14010#	14011	14018#
14019	14026#	14027	14040#	14041	14048#	14049	14056#	14057	14064#	14065	14078#	14079
14086#	14087	14094#	14095	14102#	14103	14118#	14119	14126#	14127	14134#	14135	14142#
14143	14157#	14158	14165#	14166	14173#	14174	14181#	14182	14196#	14197	14204#	14205
14213#	14214	14221#	14222	14238#	14239	14245#	14246					
1546	1563	1621	1641	1658	1710	9892#						
1068#	1544*	1561*	1573*	1639*	1656*	1667*	1747	1867*	1960*	2045*	2146*	2161*
2174*	2194*	2238*	2256*	2327*	2378*	2392*	2440*	2486*	2500*	2548*	2594*	2608*
2656*	2689*	2770*	2851*	2957*	2993*	3045*	3131*	3167*	3219*	3288*	3313*	3435*
3461*	3587*	3612*	3735*	3760*	3884*	3909*	4030*	4055*	4151	4201	4284	4382*
4395*	4419*	4508*	4521*	4545*	4650*	4673*	5781*	5787*	5789*	5931*	5939	5945*
5949	5954*	5959	5966*	5970	6047	6249*	6351*	6527*	6634*	6793*	6897*	7048*
7054	7058*	7064	7150*	7152*	7277*	7283	7287*	7293	7379*	7381*	7498*	7500*
7513*	7518*	7522*	7614*	7616*	7618*	7635*	7639*	7754*	7879*	8004*	8108*	8112*
8230*	8234*	8352*	8356*	8516*	8523	8527*	8689*	8696	8700*	8856*	8862	8866*
8969*	8974*	8982*	8987*	8991*	8995*	9183*	9188*	9194*	9198*	9202*	9206*	9402*
9404*	9406*	9408*	9410*	9416*	9420*	9619*	9627	9630*	9727*	9734*	9737*	9919*
10265*	10272*	10283*	10285*	10294*	10298*	10305*	10308*	10324	10325			
1214#	6757	6790	6871	6895	6919							
1069#	1545*	1562	1640*	1657	1748	2147*	2162	2175	2328*	2379*	2393	2441*
2487*	2501	2549*	2595*	2609	2657*	2690*	2691	2771*	2772	2852*	2853	2962*
2994	3046*	3136*	3168	3220*	3289*	3314	3436*	3462	3588*	3613	3736*	3761
3885*	3910	4031*	4056	4152	4202	4285	4383*	4396	4405*	4420	4429*	4509*
4522	4531*	4546	4555*	4615	4651*	4674	4766	4864	4963	5056	5149	5247
5345	5434	5538	5620	5718	5785*	5788*	5790*	5796	5935	5946	5955	5967

VECTRS 036550  
WCSAR 003036

WCSAOW 003270  
WCSOR 003040

	6048	6250	6354*	6528	6635*	6794	6898*	6899*	7049	7053	7059	7063	7151*
	7153*	7278	7282	7288	7292	7380*	7382*	7499*	7514	7519	7523	7615*	7617*
	7636	7640	7755	7880	8005	8111*	8233*	8355*	8518	8691	8858	8973*	8983*
	8988*	8992*	8996*	9187*	9195*	9199*	9203*	9207*	9403*	9405*	9407*	9409*	9415*
	9621	9733*	9920*	10268*	10273*	10291*	10292	10295*	10296	10299*	10300	10328	
WCSENT 003124	1108#	5779											
WCSPRS= 000001	1036#												
WCSSST 003034	1067#	1266	1372	1431	1439*	1464	1543*	1638*	1744*	1857*	1868	1950*	1961
	2033*	2046	2051	2078	2142*	2168*	2172*	2188	2190	2195	2201	2207*	2208
	2210	2224	2229	2231	2240	2246	2257	2262	2268*	2270	2272	2297	2326*
	2375*	2390*	2403	2439*	2483*	2498*	2511	2547*	2591*	2606*	2619	2655*	2679*
	2718	2723	2724*	2725*	2760*	2798	2804	2805*	2806*	2841*	2881	2886	2887*
	2888*	2956*	2963	2980	2992*	3005	3044*	3130*	3137	3154	3166*	3179	3218*
	3285*	3297*	3372	3375	3376*	3431*	3445*	3519	3522	3523*	3583*	3597*	3665
	3668	3669*	3670*	3731*	3744*	3814	3817	3818*	3879*	3893*	3962*	3965	3966*
	4026*	4039*	4109	4112	4113*	4131*	4153*	4178*	4200*	4281*	4372*	4397	4421
	4496*	4523	4547	4613*	4620*	4634*	4635*	4648*	4655*	4670*	4682	4684*	4685*
	4687	4765*	4773*	4796*	4797*	4863*	4871*	4894*	4895*	4962*	4970*	4993*	4994*
	5055*	5063*	5086*	5087*	5148*	5156*	5179*	5180*	5246*	5254*	5277*	5278*	5344*
	5352*	5433*	5440*	5465*	5466*	5537*	5545*	5561*	5562*	5619*	5633*	5649*	5650*
	5717*	5721*	5736*	5737*	5795*	5799*	5805*	5828*	5829*	5918*	5929*	5986*	5988*
	6071*	6079*	6152*	6154*	6227*	6236*	6270*	6271*	6359*	6368*	6427*	6428*	6505*
	6514*	6550*	6551*	6641*	6649*	6701*	6702*	6775*	6784*	6815*	6816*	6904*	6912*
	6960*	6961*	7038*	7044*	7073*	7075*	7156*	7162*	7187*	7189*	7267*	7273*	7302*
	7304*	7385*	7391*	7416*	7418*	7503*	7509*	7533*	7535*	7621*	7627*	7650*	7652*
	7742*	7748*	7776*	7778*	7867*	7873*	7901*	7903*	7992*	7998*	8026*	8028*	8116*
	8122*	8148*	8150*	8238*	8244*	8270*	8272*	8360*	8366*	8392*	8394*	8466*	8509*
	8571*	8573*	8642*	8682*	8744*	8746*	8820*	8848*	8893*	8895*	9026*	9032*	9110*
	9112*	9237*	9243*	9326*	9328*	9445*	9451*	9501*	9503*	9575*	9613*	9650*	9652*
	9740*	9780*	9821*	9823*	9917*	9924*	10071						
WRFLAG= 000344	1046#	5530	5622	5630	8812	9394							
WRLWHA= 000305	1042#												
WRPAT 006622	2038	2096#	2369	2477	2585								
WRTEN = 000100	1030#												
WRWHAM= 000222	1041#	1357	1429	5525	5778	5852							
WWP = 000040	1031#	1446	1476	1497	1498	1511	1512	2033	2142	2270	2375	2483	2591
	2956	3130	4648	4655									
XFCUDI= 076700	1062#	5926	5989	6075	6155	6230	6362	6508	6643	6778	6906	7041	7159
	7270	7388	7506	7624	7745	7870	7995	8119	8241	8363	8506	8679	8845
	8875	9029	9240	9448	9609	9769							
XFRCHK 037262	2961	3135	10115#										
XFRPGN 036734	1866	1959	9978#										
\$APTHD 000204	147	153#											
\$ASTAT= ***** U	10761	10776											
\$ATYC 041572	10732	10734#											
\$ATY1 041546	10730#												
\$ATY3 041554	10676	10731#											
\$ATY4 041564	10438	10733#											
\$AUTOB 001134	194#												
\$BASE 001312	276#												
\$BDADR 001122	189#												
\$BDDAT 001126	191#												
\$BELL 001226	227#	10425	10457										
\$CDW1 001316	278#												
\$CDW2 001320	279#												
\$CHARC 041542	10693*	10703*	10710	10719*	10724#								

















COMENT	1359#	1362	1399#	1402	1453#	1456	1518#	1521	1624#	1627	1715#	1718	1829#	1832	1915#
	1918	2009#	2012	2106#	2109	2339#	2342	2448#	2451	2556#	2559	2663#	2666	2739#	2742
	2820#	2823	2902#	2905	3076#	3080	3251#	3253	3404#	3407	3554#	3556	3701#	3703	3849#
	3851	3996#	3998	4143#	4146	4180#	4182	4264#	4266	4348#	4350	4474#	4476	4599#	4601
	4638#	4713#	4715	4811#	4813	4909#	4911	5002#	5004	5095#	5097	5193#	5195	5291#	5293
	5386#	5388	5478#	5480	5573#	5575	5662#	5664	5757#	5759	5861#	5863	5996#	5999	6162#
	6164	6275#	6278	6435#	6441	6554#	6558	6711#	6713	6822#	6824	6969#	6971	7084#	7086
	7198#	7200	7313#	7315	7427#	7429	7544#	7546	7661#	7663	7787#	7789	7912#	7914	8037#
	8039	8159#	8161	8281#	8283	8406#	8408	8582#	8584	8755#	8757	8909#	8911	9121#	9123
	9337#	9339	9514#	9516	9666#	9668									
COMMEN	125#														
ENDCOM	125#														
EPROR	19#	1393	1424	1436	1444	1449	1472	1479	1485	1494	1501	1507	1515	1554	1569
	1580	1609	1615	1648	1663	1674	1695	1702	1764	1776	1789	1801	1892	1988	2052
	2080	2155	2191	2202	2211	2225	2232	2247	2263	2273	2299	2387	2404	2415	2495
	2512	2523	2603	2620	2631	2699	2728	2780	2809	2861	2891	2981	3006	3027	3155
	3180	3202	3344	3381	3492	3528	3639	3673	3788	3823	3936	3969	4084	4118	4172
	4229	4256	4313	4340	4445	4449	4453	4462	4466	4470	4570	4574	4578	4587	4591
	4595	4622	4692	4780	4799	4878	4897	4977	4996	5070	5089	5163	5182	5261	5280
	5359	5376	5445	5452	5467	5550	5563	5638	5651	5728	5740	5750	5801	5831	5943
	5953	5963	5975	5990	6089	6097	6105	6113	6121	6129	6140	6147	6156	6260	6272
	6400	6429	6538	6552	6679	6704	6805	6817	6939	6963	7055	7065	7076	7171	7179
	7190	7284	7294	7305	7400	7408	7419	7525	7536	7642	7653	7766	7779	7891	7904
	8016	8029	8139	8151	8261	8273	8383	8395	8524	8574	8697	8747	8863	8896	9062
	9113	9278	9329	9472	9504	9628	9653	9799	9824	10329					
ESCAPE	125#														
GETPRI	125#														
GETSWR	125#														
GTNXTN	1225#	1370	1418	1462	1533	1634	1739	1853	1943	2032	2135	2368	2476	2584	2678
	2759	2840	2928	3103	3278	3428	3581	3728	3876	4023	4150	4197	4280	4369	4495
	4609	4643	4754	4852	4951	5044	5137	5235	5333	5424	5518	5609	5703	5770	5904
	6041	6202	6320	6481	6600	6753	6866	7019	7133	7248	7362	7479	7597	7711	7837
	7962	8087	8209	8331	8455	8631	8805	8958	9170	9387	9564	9716			
GTNXTS	1225#	1370	1418	1462	1533	1634	1739	1853	1943	2032	2135	2368	2476	2584	2678
	2759	2840	2928	3103	3278	3428	3581	3728	3876	4023	4150	4197	4280	4369	4495
	4609	4643	4754	4852	4951	5044	5137	5235	5333	5424	5518	5609	5703	5770	5904
	6041	6202	6320	6481	6600	6753	6866	7019	7133	7248	7362	7479	7597	7711	7837
	7962	2087	8209	8331	8455	8631	8805	8958	9170	9387	9564	9716			
MULT	125#														
NEWTST	125#	1360	1400	1454	1519	1625	1716	1830	1916	2010	2107	2340	2449	2557	2664
	2740	2821	2903	3078	3251	3405	3554	3701	3849	3996	4144	4180	4264	4348	4474
	4599	4639	4713	4811	4909	5002	5095	5193	5291	5386	5478	5573	5662	5757	5861
	5997	6162	6276	6439	6556	6711	6822	6969	7084	7198	7313	7427	7544	7661	7787
	7912	8037	8159	8281	8406	8582	8755	8909	9121	9337	9514	9666	9837		
POP	125#	10634	10773	10774	10852	10853									
PUSH	125#	10593	10734	10736	10757	10829	10835								
REPORT	125#														
SCOPE	20#	1369	1417	1461	1532	1633	1738	1852	1942	2031	2134	2367	2475	2583	2677
	2758	2839	2927	3102	3277	3427	3580	3727	3875	4022	4149	4196	4279	4368	4494
	4608	4642	4753	4851	4950	5043	5136	5234	5332	5423	5517	5608	5702	5769	5903
	6040	6201	6319	6480	6599	6752	6865	7018	7132	7247	7361	7478	7596	7710	7836
	7961	8086	8208	8330	8454	8630	8804	8957	9169	9386	9563	9715	9840	9858	
SETPRI	125#														
SETTRA	10807#	10816	10817	10818	10819	10822									
SETUP	125#														
SKIP	125#	1383	1447	1513	1712	1819	1906	2003	2073	2330	3072	3247	3398	3545	3688



PDP-11/60 WCS DIAGNOSTIC		MACY11		30(1046)		25-OCT-77		20:12		PAGE 292		E 8		SEQ 0291	
DQKUA.A.MIC		24-OCT-77 14:22		CROSS REFERENCE TABLE		--		MACRO NAMES				SEQ 0302			
	3839	3985	4133	4252	4336	4425	4438	4551	4564	4629	4787	4885	4984	5077	5170
	5268	5366	5458	5548	5554	5636	5642	5725	5732	5739	5981	6149	6417	6545	6694
	6953	7067	7181	7296	7410	7527	7644	7771	7896	8021	8143	8265	8387	8530	8703
	8869	9069	9285	9477	9634	9805									
SLASH	125#														
SPACE	125#														
STARS	125#	140	142	149	162	172	231	234	1360	1368	1400	1416	1454	1460	1519
	1531	1625	1632	1716	1737	1830	1851	1916	1941	2010	2030	2107	2133	2340	2366
	2449	2474	2557	2582	2664	2676	2740	2757	2821	2838	2903	2926	3078	3101	3251
	3276	3405	3426	3554	3579	3701	3726	3849	3874	3996	4021	4144	4148	4180	4195
	4264	4278	4348	4367	4474	4493	4599	4607	4639	4641	4713	4752	4811	4850	4909
	4949	5002	5042	5095	5135	5193	5233	5291	5331	5386	5422	5478	5516	5573	5607
	5662	5701	5757	5768	5861	5902	5997	6039	6162	6200	6276	6318	6439	6479	6556
	6598	6711	6751	6822	6864	6969	7017	7084	7131	7198	7246	7313	7360	7427	7477
	7544	7595	7661	7709	7787	7835	7912	7960	8037	8085	8159	8207	8281	8329	8406
	8453	8582	8629	8755	8803	8909	8956	9121	9168	9337	9385	9514	9562	9666	9714
	9837	9839	9851	10344	10407	10459	10506	10583	10650	10729	10786	10825	10841		
SWRSU	125#	1309#													
TRMTRP	10807#														
TYPBIN	125#														
TYPDEC	125#	9870													
TYPNAM	125#														
TYPNUM	125#														
TYPOCS	125#														
TYPOCT	125#	10471	10495												
TYPTXT	125#	1332													
U	1219#	12123	12130	12138	12145	12152	12160	12178	12185	12192	12199	12206	12224	12231	12240
	12247	12255	12273	12280	12288	12295	12302	12318	12325	12333	12340	12347	12353	12360	12367
	12375	12394	12401	12409	12416	12423	12430	12437	12443	12461	12468	12476	12483	12489	12507
	12514	12522	12529	12535	12552	12559	12567	12574	12580	12597	12604	12612	12619	12625	12643
	12650	12658	12665	12671	12688	12695	12703	12710	12716	12733	12741	12749	12756	12762	12779
	12786	12794	12801	12807	12824	12831	12839	12847	12853	12870	12877	12885	12892	12898	12915
	12922	12930	12937	12944	12961	12968	12976	12983	12989	13006	13013	13021	13029	13035	13052
	13059	13067	13074	13080	13095	13104	13113	13121	13129	13136	13143	13150	13158	13166	13174
	13188	13198	13207	13215	13223	13230	13237	13244	13252	13260	13268	13282	13291	13300	13308
	13316	13323	13330	13337	13345	13353	13362	13376	13385	13394	13402	13410	13417	13424	13431
	13439	13447	13455	13469	13478	13487	13495	13503	13511	13518	13525	13533	13541	13549	13563
	13572	13581	13589	13597	13604	13611	13618	13626	13634	13642	13657	13666	13675	13683	13691
	13698	13705	13712	13720	13728	13736	13750	13759	13768	13776	13785	13792	13799	13806	13814
	13822	13830	13845	13853	13861	13869	13884	13892	13901	13909	13924	13932	13940	13948	13963
	13971	13979	13987	14001	14010	14018	14026	14040	14048	14056	14064	14078	14086	14094	14102
	14118	14126	14134	14142	14157	14165	14173	14181	14196	14204	14213	14221	14238	14245	
UPCODE	10823#	10848													
\$\$CMRE	170#	209	210	211	212	213	214	215	216						
\$\$CMTM	170#	217	218	219	220	221	222	223	224						
\$\$ESCA	125#														
\$\$NEWT	125#	1360	1400	1454	1519	1625	1716	1830	1916	2010	2107	2340	2449	2557	2664
	2740	2821	2903	3078	3251	3405	3554	3701	3849	3996	4144	4180	4264	4348	4474
	4599	4639	4713	4811	4909	5002	5095	5193	5291	5386	5478	5573	5662	5757	5861
	5997	6162	6276	6439	6556	6711	6822	6969	7084	7198	7313	7427	7544	7661	7787
	7912	8037	8159	8281	8406	8582	8755	8909	9121	9337	9514	9666	9837		
\$\$SET	10807#	10816	10817	10818	10819	10822									
\$\$SETM	1325#														
\$\$SKIP	125#	1383	1447	1513	1712	1819	1906	2003	2073	2330	3072	3247	3398	3545	3688
	3839	3985	4133	4252	4336	4425	4438	4551	4564	4629	4787	4885	4984	5077	5170
	5268	5366	5458	5548	5554	5636	5642	5725	5732	5739	5981	6150	6417	6545	6694





ADC	9950														
ADD	1877	1907	2004	3049	3070	3245	3589	3622	3738	3771	3886	3887	3919	3920	4033
	4067	4433	4558	5787	5789	5811	9050	9053	9266	9269	9910	10294	10298	10479	10532
	10542	10613	10684	10744	10756	10768									
ASH	4136	4137													
ASL	2705	2785	2867	3069	3243	4434	4442	4443	4459	4460	4559	4567	4568	4584	4585
	9984	9985	10035	10118	10121	10476	10477	10478	10796						
ASLB	10618														
ASR	3693	3694	3695	3696	3988	3989	3990	3991	4138	10037	10751				
BCC	9953	10189	10619												
BEQ	1267	1383	1441	1447	1477	1483	1492	1505	1513	1566	1660	1707	1709	1773	1786
	1798	1884	1978	2049	2198	2209	2243	2260	2411	2519	2627	2693	2695	2707	2712
	2719	2722	2774	2776	2787	2792	2799	2855	2857	2869	2875	2882	2884	2960	2970
	2973	2976	3001	3020	3022	3057	3072	3134	3144	3147	3150	3175	3194	3196	3231
	3247	3339	3357	3373	3378	3398	3487	3505	3520	3525	3545	3635	3650	3666	3688
	3783	3799	3815	3820	3839	3932	3947	3963	3985	4079	4094	4110	4115	4133	4166
	4223	4250	4307	4334	4399	4549	4777	4875	4974	5067	5160	5258	5356	5443	5548
	5636	5723	5937	5948	5957	5969	6084	6092	6100	6108	6116	6124	6135	6142	6254
	6392	6532	6672	6798	6932	7050	7060	7167	7175	7279	7289	7396	7404	7515	7632
	7761	7886	8011	8135	8257	8379	8520	8522	8693	8695	8861	9061	9277	9468	9626
	9795	9874	10031	10034	10076	10125	10128	10303	10326	10372	10374	10376	10380	10389	10421
	10424	10446	10449	10481	10486	10499	10559	10674	10687	10722	10738	10742	10762	10764	
BGE	5813	10392													
BGT	9066	9282	9632	9803	9865	10566	10627								
BHI	4689	5842	10378												
BHIS	1604	1693													
BIC	1490	1511	1870	1963	2047	2172	2196	2241	2258	2390	2498	2606	2966	2992	3140
	3166	3353	3355	3501	3503	3646	3648	3795	3797	3943	3945	4090	4092	4140	4620
	4655	4773	4871	4970	5063	5156	5254	5352	5440	5523	5545	5627	5633	5721	5776
	5780	5783	5799	5805	5929	6079	6236	6368	6381	6394	6514	6649	6665	6784	6803
	6899	6912	6925	7044	7162	7273	7391	7509	7627	7748	7873	7998	8122	8244	8366
	8509	8682	8848	9032	9243	9451	9613	9780	9798	9862	9982	9986	9987	10073	10083
	10120	10122	10130	10176	10197	10267	10285	10556							
BIS	1355	1427	2168	2207	2268	2708	2788	2870	3315	3318	3359	3363	3464	3467	3508
	3511	3614	3617	3653	3657	3762	3765	3802	3806	3911	3914	3950	3954	3962	4057
	4061	4097	4101	5528	5618	5784	5850	6353	6387	7733	7758	7858	7883	7983	8008
	8110	8130	8232	8252	8354	8374	8474	8650	8811	8828	9393	9732	9988	9989	10084
	10123	10126	10129	10131	10204	10561	10562	10621	10622						
BISB	7723	7848	7973	8098	8220	8342	9002	9213	9426	9747	10468				
BIT	1262	1264	1266	1421	1433	1440	1446	1469	1476	1482	1491	1498	1504	1512	2208
	2270	2718	2798	2881	2959	3133	3372	3519	3665	3814	4109	4398	4422	4524	4548
	5809	9905	9998	10033	10040	10124	10127	10325	10357	10371	10379	10386	10423	10430	10445
BITB	10673	10678	10710	10741											
BLOS	10193														
BLT	10567	10610	10626	10701											
BMI	2189	2230	8473	8649	8827	9001	9055	9212	9271	9425	9463	9731	9746	10617	
BNE	1263	1265	1292	1316	1327	1342	1380	1422	1434	1470	1499	1814	1876	1879	1906
	1970	1973	2003	2061	2071	2271	2286	2425	2533	2641	2802	3051	3224	3293	3323
	3328	3440	3472	3477	3592	3621	3625	3769	3773	3918	3922	4065	4069	4160	4176
	4423	4437	4525	4562	4617	4654	4677	5810	5815	9906	9923	9999	10041	10271	10276
	10293	10297	10301	10307	10358	10387	10431	10436	10453	10469	10491	10557	10615	10672	10679
	10681	10689	10697	10711	10718	10740	10746	10749	10766	10847					
BPL	4168	4225	4252	4309	4336	4401	4425	4527	4551	4683	5725	5739	5979	10443	10555
	10601	10631	10666	10715											
BPT	5435														
BR	1269	1318	1333	1397	1585	1593	1596	1613	1678	1681	1684	1700	1712	1819	1880

	1899	1974	1996	2064	2073	2091	2166	2179	2214	2314	2330	2427	2433	2445	2535
	2541	2552	2643	2649	2660	2714	2732	2794	2813	2877	2895	3032	3036	3207	3211
	3226	3330	3366	3387	3478	3513	3534	3626	3659	3677	3774	3808	3828	3923	3956
	3974	4070	4103	4122	4404	4428	4438	4448	4452	4456	4465	4469	4473	4530	4554
	4564	4573	4577	4581	4590	4594	4598	4629	4678	4699	4787	4885	4984	5077	5170
	5268	5366	5448	5458	5554	5642	5732	5746	5802	5817	5843	5981	6150	6265	6417
	6545	6694	6810	6953	7068	7182	7297	7411	7528	7645	7771	7896	8021	8143	8265
	8387	8478	8530	8654	8703	8832	8869	9007	9069	9218	9285	9431	9477	9634	9735
	9752	9805	9942	10044	10048	10086	10309	10340	10360	10366	10369	10382	10385	10441	10474
	10501	10533	10548	10569	10612	10629	10668	10694	10704	10713	10720	10732	10754	10839	10867
CLC	1887	1981													
CLP	1290	1304	1306	1325	1340	1344	1375	1432	1439	1465	1535	1536	1544	1545	1552
	1553	1561	1567	1568	1578	1579	1639	1646	1656	1661	1672	1759	1854	1944	2035
	2044	2328	2441	2549	2657	2679	2680	2681	2685	2696	2725	2726	2760	2761	2762
	2806	2841	2842	2843	2888	2930	2934	3105	3109	3285	3286	3287	3298	3299	3300
	3304	3308	3431	3433	3447	3448	3452	3583	3586	3597	3599	3600	3604	3670	3731
	3733	3746	3748	3752	3879	3882	3895	3896	3900	4025	4026	4028	4041	4043	4047
	4131	4155	4162	4178	4208	4209	4219	4227	4260	4291	4302	4318	4338	4374	4376
	4383	4429	4500	4502	4531	4635	4644	4649	4651	4670	4671	4685	4797	4895	4994
	5087	5180	5278	5425	5466	5519	5562	5610	5650	5737	5771	5772	5788	5790	5829
	5906	5923	5988	6133	6154	6205	6271	6326	6329	6377	6428	6484	6520	6551	6607
	6612	6658	6659	6702	6816	6875	6961	7033	7034	7035	7075	7144	7145	7146	7147
	7148	7189	7262	7263	7264	7304	7373	7374	7375	7376	7377	7418	7492	7493	7494
	7495	7496	7500	7535	7608	7609	7610	7611	7612	7618	7652	7712	7724	7778	7849
	7903	7974	8028	8099	8150	8221	8272	8343	8394	8515	8573	8746	8895	9004	9112
	9195	9215	9328	9420	9428	9503	9603	9604	9605	9606	9607	9652	9737	9749	9763
	9764	9765	9766	9823	9842	9859	9860	9918	9920	9924	9945	10028	10263	10273	10384
CLRB	10399	10467	10546	10604	10607	10845									
CMP	9843	10383	10633	10693	10719	10770	10771	10772							
	1291	1315	1341	1387	1602	1603	1659	1690	1692	1706	1708	1772	1785	1797	1813
	1875	1969	2048	2186	2197	2242	2259	2294	2311	2321	2410	2430	2435	2518	2538
	2543	2626	2646	2651	2692	2720	2773	2800	2854	2885	2972	3019	3042	3071	3146
	3193	3216	3246	3292	3322	3327	3356	3374	3439	3471	3476	3504	3521	3591	3620
	3624	3649	3667	3768	3798	3816	3917	3946	3964	4064	4093	4111	4159	4165	4175
	4249	4306	4436	4561	4633	4653	4676	4686	4688	4775	4793	4873	4891	4972	4990
	5065	5083	5158	5176	5256	5274	5354	5372	5442	5451	5464	5547	5560	5635	5648
	5722	5734	5748	5804	5812	5814	5826	5841	5936	5947	5956	5968	5987	6083	6091
	6099	6107	6115	6123	6134	6141	6153	6251	6269	6390	6426	6529	6549	6670	6700
	6795	6814	6930	6959	7049	7059	7074	7166	7174	7188	7278	7288	7303	7395	7403
	7417	7514	7534	7631	7651	7760	7777	7885	7902	8010	8027	8134	8149	8256	8271
	8378	8393	8521	8572	8694	8745	8860	8894	8978	9060	9068	9111	9180	9276	9284
	9327	9467	9502	9625	9651	9794	9822	9922	10075	10192	10210	10270	10275	10292	10296
CMPB	10300	10302	10367	10391	10452	10625									
COM	1326	10373	10377	10435	10671	10686	10688	10696	10717	10721	10739				
	2063	2965	3139	3316	3317	3319	3320	3361	3362	3364	3465	3466	3468	3469	3509
	3510	3512	3615	3616	3618	3619	3655	3656	3658	3763	3764	3766	3767	3804	3805
	3807	3912	3913	3915	3916	3952	3953	3955	4059	4060	4062	4063	4099	4100	4102
	5782	10081	10175	10266											
DEC	3772	3921	4068	4261	4345	4416	4430	4542	4556	8865	9065	9281	9417	9631	9802
	9863	10475													
DECB	10554	10565	10700	10703											
EMT	19														
HALT	132	6076	10444	10454	10667	10838	10866								
INC	1379	1390	1595	1598	1683	1686	1878	2062	2070	2424	2532	2640	2703	2784	2865
	3050	3290	3291	3325	3326	3358	3437	3438	3474	3475	3506	3590	3623	3651	3737
	3770	3800	3948	4032	4066	4095	4158	4167	4174	4210	4224	4235	4251	4293	4308



	4319	4335	4384	4400	4407	4424	4435	4510	4526	4532	4550	4560	4616	4652	4675
	4785	4883	4982	5075	5168	5266	5364	5724	5730	5738	5808	5816	5840	5945	5954
	5966	6052	6054	6056	6058	6060	6062	6064	6066	6263	6355	6414	6494	6540	6541
	6542	6636	6637	6690	6691	6807	6900	6950	7152	7381	7768	7893	8018	8112	8234
	8356	8526	8527	8699	8700	8866	8974	8975	9064	9188	9189	9280	9404	9406	9408
	9410	9416	9474	9630	9734	9801	9861	9921	10032	10187	10269	10274	10306	10390	10426
INCB	10560	10568	10611	10769	10846										
	1885	1971	1979	2694	2721	2775	2801	2856	2883	2977	3002	3023	3151	3176	3223
	3338	3377	3486	3524	3634	3782	3819	3931	4078	4114	10395	10420	10723		
IOT	20														
JMP	136	1385	1714	1908	2005	2287	2305	2734	2803	2815	2897	3058	3073	3232	3248
	3389	3536	3679	3830	3976	4124	4444	4461	4569	4586	4703	9881			
JEP	1546	1551	1563	1577	1621	1641	1645	1658	1671	1710	1761	1866	1873	1959	1967
	2152	2295	2312	2322	2384	2431	2492	2539	2600	2647	2961	2967	3135	3141	3284
	3403	3429	3550	3582	3699	3729	3847	3878	3994	4024	4141	4757	4855	4954	5047
	5140	5238	5336	5429	5533	5612	5710	5914	6042	6222	6339	6500	6622	6770	6885
	7021	7135	7250	7364	7481	7599	7714	7839	7964	8089	8211	8333	8460	8636	8814
MOV	8963	9175	9396	9569	9721	9876	10078	10432	10438	10676	10695	10702	10709	10758	
	1234	1236	1239	1243	1261	1273	1277	1281	1289	1293	1295	1296	1297	1298	1299
	1300	1301	1302	1303	1307	1308	1311	1312	1313	1314	1319	1321	1322	1323	1328
	1339	1343	1346	1347	1350	1351	1352	1370	1372	1373	1374	1376	1381	1388	1391
	1418	1423	1431	1435	1443	1448	1462	1464	1466	1467	1471	1478	1484	1493	1497
	1500	1506	1514	1533	1539	1540	1541	1543	1548	1549	1550	1558	1559	1560	1562
	1573	1574	1575	1576	1601	1634	1635	1636	1637	1638	1640	1642	1643	1644	1647
	1653	1654	1655	1657	1662	1667	1668	1669	1670	1673	1689	1739	1740	1741	1742
	1743	1744	1745	1746	1747	1748	1749	1750	1753	1754	1755	1756	1757	1758	1760
	1762	1763	1768	1769	1770	1771	1774	1775	1781	1783	1784	1787	1788	1794	1795
	1796	1799	1800	1805	1806	1807	1808	1809	1853	1857	1858	1859	1861	1862	1864
	1867	1868	1886	1890	1891	1943	1950	1951	1952	1954	1956	1960	1961	1980	1984
	1986	2032	2033	2037	2038	2039	2040	2042	2045	2046	2050	2051	2072	2078	2079
	2135	2137	2138	2139	2140	2142	2143	2144	2146	2147	2149	2150	2151	2153	2154
	2161	2165	2171	2174	2187	2190	2194	2195	2199	2200	2201	2210	2221	2222	2223
	2224	2231	2237	2238	2240	2244	2245	2246	2255	2256	2257	2261	2262	2272	2278
	2282	2284	2293	2297	2298	2310	2320	2326	2327	2329	2368	2369	2370	2371	2373
	2375	2376	2378	2379	2381	2382	2383	2385	2386	2389	2392	2393	2400	2401	2402
	2403	2408	2412	2413	2414	2421	2429	2439	2440	2442	2476	2477	2478	2479	2481
	2483	2484	2486	2487	2489	2490	2491	2493	2494	2497	2500	2501	2508	2509	2510
	2511	2516	2520	2521	2522	2529	2537	2547	2548	2550	2584	2585	2586	2587	2589
	2591	2592	2594	2595	2597	2598	2599	2601	2602	2605	2608	2609	2616	2617	2618
	2619	2624	2628	2629	2630	2637	2645	2655	2656	2658	2678	2682	2683	2684	2686
	2689	2690	2691	2697	2698	2710	2713	2723	2724	2727	2733	2737	2759	2763	2764
	2765	2766	2767	2770	2771	2772	2777	2778	2779	2790	2793	2804	2805	2807	2808
	2814	2819	2840	2844	2845	2846	2847	2848	2851	2852	2853	2858	2859	2860	2873
	2876	2886	2887	2889	2890	2896	2901	2928	2942	2943	2944	2945	2948	2949	2956
	2957	2958	2962	2963	2964	2971	2978	2979	2980	2991	2993	2994	2999	3003	3004
	3005	3024	3025	3026	3035	3043	3044	3045	3046	3053	3103	3117	3118	3119	3120
	3123	3130	3131	3132	3136	3137	3138	3145	3152	3153	3154	3165	3167	3168	3173
	3177	3178	3179	3199	3200	3201	3217	3218	3219	3220	3227	3244	3278	3288	3289
	3297	3302	3306	3311	3312	3313	3314	3341	3342	3343	3352	3354	3370	3375	3376
	3379	3380	3396	3399	3400	3401	3402	3428	3432	3435	3436	3445	3446	3450	3454
	3458	3459	3461	3462	3489	3490	3491	3500	3502	3518	3522	3523	3526	3527	3543
	3546	3547	3548	3549	3581	3584	3587	3588	3598	3602	3606	3610	3611	3612	3613
	3636	3637	3638	3645	3647	3663	3668	3669	3671	3672	3686	3689	3690	3697	3698
	3728	3732	3734	3735	3736	3744	3745	3747	3750	3754	3758	3759	3760	3761	3784
	3786	3787	3794	3796	3813	3817	3818	3821	3822	3837	3840	3841	3845	3846	3876
	3880	3883	3884	3885	3893	3894	3898	3902	3906	3907	3908	3909	3910	3933	3934

3935	3942	3944	3961	3965	3966	3967	3968	3983	3986	3987	3992	3993	4023	4027
4029	4030	4031	4039	4040	4042	4045	4049	4053	4054	4055	4056	4080	4082	4083
4089	4091	4108	4112	4113	4116	4117	4130	4134	4135	4139	4150	4151	4152	4153
4154	4156	4157	4163	4164	4169	4170	4171	4197	4198	4199	4200	4201	4202	4203
4207	4218	4220	4221	4222	4226	4228	4233	4244	4245	4246	4247	4248	4253	4254
4255	4280	4281	4282	4283	4284	4285	4286	4290	4292	4301	4303	4304	4305	4310
4311	4312	4329	4330	4331	4332	4333	4337	4339	4344	4369	4370	4371	4372	4373
4375	4380	4381	4382	4392	4393	4394	4395	4396	4397	4402	4403	4405	4417	4418
4419	4420	4421	4426	4427	4441	4458	4495	4496	4497	4498	4499	4501	4506	4507
4508	4509	4518	4519	4520	4521	4522	4523	4528	4529	4543	4544	4545	4546	4547
4552	4553	4555	4566	4583	4609	4611	4612	4613	4621	4634	4636	4643	4648	4650
4662	4669	4673	4674	4684	4687	4691	4705	4754	4755	4756	4762	4763	4765	4774
4778	4779	4795	4796	4798	4852	4853	4854	4860	4861	4863	4872	4876	4877	4893
4894	4896	4951	4952	4953	4959	4960	4962	4971	4975	4976	4992	4993	4995	5044
5045	5046	5052	5053	5055	5064	5068	5069	5085	5086	5088	5137	5138	5139	5145
5146	5148	5157	5161	5162	5178	5179	5181	5235	5236	5237	5243	5244	5246	5255
5259	5260	5276	5277	5279	5333	5334	5335	5341	5342	5344	5353	5357	5358	5374
5375	5424	5426	5427	5428	5433	5441	5444	5449	5450	5463	5465	5472	5518	5520
5531	5537	5546	5549	5559	5561	5609	5611	5619	5634	5637	5647	5649	5703	5705
5706	5707	5708	5709	5715	5716	5717	5726	5727	5735	5736	5749	5770	5773	5779
5781	5785	5792	5793	5794	5795	5800	5806	5807	5827	5828	5830	5846	5847	5904
5907	5908	5909	5910	5911	5912	5918	5921	5922	5924	5930	5931	5935	5939	5940
5941	5942	5946	5949	5950	5951	5952	5955	5959	5960	5961	5962	5965	5967	5970
5971	5972	5973	5986	5989	6041	6047	6048	6049	6051	6053	6055	6057	6059	6061
6063	6065	6067	6068	6071	6074	6080	6081	6082	6085	6086	6087	6088	6093	6094
6095	6096	6101	6102	6103	6104	6109	6110	6111	6112	6117	6118	6119	6120	6125
6126	6127	6128	6131	6132	6136	6137	6138	6139	6143	6144	6145	6146	6152	6155
6202	6204	6206	6207	6226	6227	6237	6241	6243	6245	6249	6250	6256	6257	6258
6259	6270	6320	6324	6325	6346	6347	6349	6351	6354	6358	6359	6369	6374	6375
6376	6378	6393	6395	6397	6398	6427	6481	6483	6485	6486	6489	6504	6505	6515
6519	6522	6523	6525	6527	6528	6534	6535	6536	6537	6550	6600	6605	6606	6629
6630	6632	6634	6635	6640	6641	6650	6655	6656	6657	6660	6673	6675	6676	6677
6701	6753	6756	6757	6758	6760	6774	6775	6785	6789	6790	6791	6792	6793	6794
6800	6801	6802	6804	6815	6866	6870	6871	6873	6892	6893	6895	6897	6898	6903
6904	6913	6918	6919	6920	6924	6926	6934	6935	6936	6937	6960	7019	7030	7031
7032	7037	7038	7045	7048	7051	7052	7053	7054	7058	7061	7062	7063	7064	7073
7133	7149	7150	7151	7153	7155	7156	7163	7168	7169	7170	7176	7177	7178	7187
7248	7259	7260	7261	7266	7267	7274	7277	7280	7281	7282	7283	7287	7290	7291
7292	7293	7302	7362	7378	7379	7380	7382	7384	7385	7392	7397	7398	7399	7405
7406	7407	7416	7479	7490	7497	7498	7499	7502	7503	7510	7513	7516	7517	7518
7519	7520	7521	7522	7523	7524	7533	7597	7613	7614	7615	7616	7617	7620	7621
7628	7633	7634	7635	7636	7637	7638	7639	7640	7641	7650	7711	7720	7721	7730
7731	7739	7741	7742	7749	7751	7752	7754	7755	7763	7764	7765	7776	7837	7845
7846	7855	7856	7864	7866	7867	7874	7876	7877	7879	7880	7888	7889	7890	7901
7962	7970	7971	7980	7981	7989	7991	7992	7999	8001	8002	8004	8005	8013	8014
8015	8026	8087	8095	8096	8105	8106	8107	8108	8111	8115	8116	8123	8125	8126
8127	8137	8138	8148	8209	8217	8218	8227	8228	8229	8230	8233	8237	8238	8245
8247	8248	8249	8259	8260	8270	8331	8339	8340	8349	8350	8351	8352	8355	8359
8360	8367	8369	8370	8371	8381	8382	8392	8455	8465	8466	8471	8494	8496	8499
8502	8504	8510	8513	8514	8516	8518	8519	8523	8571	8631	8641	8642	8647	8670
8672	8675	8677	8683	8686	8687	8688	8689	8691	8692	8696	8744	8805	8819	8820
8825	8843	8849	8852	8853	8854	8856	8858	8859	8862	8893	8958	8969	8970	8971
8973	8982	8983	8984	8987	8988	8989	8991	8992	8993	8995	8996	8999	9023	9025
9026	9033	9036	9037	9038	9039	9040	9041	9042	9045	9046	9047	9049	9052	9059
9110	9170	9183	9184	9185	9187	9194	9198	9199	9200	9202	9203	9204	9206	9207
9210	9234	9236	9237	9244	9247	9248	9249	9250	9251	9252	9253	9254	9255	9256



	9257	9258	9261	9262	9263	9265	9268	9275	9326	9387	9402	9403	9405	9407	9409
	9412	9413	9415	9423	9442	9444	9445	9452	9455	9456	9457	9459	9461	9469	9470
	9471	9501	9564	9574	9575	9579	9581	9583	9585	9590	9591	9592	9593	9594	9597
	9599	9601	9614	9617	9618	9619	9621	9622	9623	9627	9650	9716	9727	9728	9733
	9739	9740	9744	9762	9772	9774	9776	9778	9781	9785	9786	9787	9789	9790	9796
	9797	9821	9845	9866	9870	9873	9892	9893	9894	9907	9916	9917	9919	9925	9940
	9941	9944	9947	9948	9954	9955	9956	9978	9979	9980	9983	9990	9991	10003	10018
	10019	10020	10021	10023	10024	10027	10036	10050	10051	10052	10053	10068	10069	10071	10072
	10074	10080	10082	10085	10087	10088	10089	10115	10116	10117	10119	10132	10133	10173	10178
	10181	10186	10190	10196	10203	10209	10257	10258	10259	10260	10264	10265	10268	10272	10279
	10281	10283	10284	10288	10289	10291	10295	10299	10308	10311	10312	10313	10314	10324	10327
	10328	10336	10339	10362	10363	10365	10368	10381	10393	10394	10397	10398	10401	10402	10422
	10427	10447	10450	10466	10471	10480	10485	10490	10492	10496	10529	10537	10538	10539	10545
	10552	10570	10571	10572	10573	10574	10594	10595	10596	10597	10598	10599	10600	10605	10608
	10628	10634	10635	10636	10637	10638	10640	10641	10669	10670	10675	10683	10698	10735	10736
	10743	10747	10752	10753	10755	10757	10767	10773	10774	10792	10793	10797	10803	10804	10827
	10828	10829	10830	10831	10832	10833	10834	10835	10836	10837	10843	10844	10848	10850	10852
	10853	10854	10855	10856	10857	10858	10859	10860	10863						
MOVB	1305	1345	1863	1955	2687	2688	2768	2769	2849	2850	2951	3125	3210	3309	3310
	3456	3457	3608	3609	3756	3757	3904	3905	4051	4052	6210	6212	6247	6327	6352
	6380	6386	6487	6609	6664	7722	7732	7734	7757	7762	7847	7857	7859	7882	7887
	7972	7982	7984	8007	8012	8097	8109	8129	8131	8136	8219	8231	8251	8253	8258
	8341	8353	8373	8375	8380	8472	8475	8648	8651	8826	8829	9000	9003	9051	9211
	9214	9267	9424	9427	9460	9730	9745	9748	10396	10400	10429	10437	10530	10531	10534
	10535	10536	10540	10543	10544	10563	10603	10606	10620	10623	10632	10680	10708	10716	10730
	10731	10733	10795												
NEG	10541	10602													
NOP	2163	2176	2177	2394	2395	2502	2503	2610	2611	2995	3169	9877	9878	9879	
RESET	9875														
ROL	9949	10030	10547	10549	10550	10551	10553								
ROLB	1888	1889	1982	1983											
ROR	9952	10188													
RTI	1245	1279	1282	1320	9911	10403	10456	10575	10642	10685	10805	10865			
RTS	9895	9926	9957	9992	10006	10054	10090	10134	10212	10320	10494	10725	10775	10798	
SEC	10029														
SOB	2709	2789	2872	3739	3888	4034	4211	4236	4262	4294	4320	4346	4385	4408	4431
	4511	4533	4557	4786	4884	4983	5076	5169	5267	5365	5731	6217	6264	6333	6356
	6415	6495	6544	6616	6638	6692	6765	6809	6879	6901	6951	7727	7737	7769	7852
	7862	7894	7977	7987	8019	8102	8113	8141	8224	8235	8263	8346	8357	8385	8528
	8701	8867	8976	9190	9418	9475	9951								
SUB	1240	1274	1600	1688	5978	10004	10305	10428	10609	10750					
SWAB	1953	3121	3691	3692	3842	3843	5539	9981							
TRAP	10807	10816	10817	10818	10819	10822									
TST	1377	1378	1382	1565	1780	1793	1810	1903	2000	2060	2068	2162	2175	2188	2229
	2280	2281	2419	2527	2635	2706	2786	2868	2969	3143	3397	3544	3687	3838	3984
	4132	4615	4682	4766	4864	4963	5056	5149	5247	5345	5434	5538	5620	5718	5796
	9054	9270	9462	10364	10388	10442	10448	10498	10558	10614	10624	10682	10690	10712	10745
	10763	10765	10794												
TSTB	1883	1977	2975	3000	3021	3149	3174	3195	10375	10616	10630	10665	10714	10737	10748
	10761														
XOR	2066	2283	2285	2422	2530	2638	2711	2791	2874	3054	3228	10026	10174	10290	
ASC11	228	229	10931	10933											
ASC12	227	230	1335	9884	10502	10869	10875	10876	10877	10878	10881	10885	10888	10891	10892
	10893	10895	10897	10899	10901	10903	10906	10909	10911	10916	10923	10928	10937	10942	10944
	10945	10950	10955	10960	10963	10968	10974	10979	10985	10991	10996	11002	11008	11013	11018
	11024	11031	11038	11045	11048	11056	11066	11078	11087	11092	11100	11109	11116	11122	11127

	11133	11144	11154	11165	11174	11183	11192	11202	11212	11221	11229	11238	11247	11260	11270
	11281	11291	11303	11312	11323	11333	11343	11353	11365	11375	11387	11398	11406	11416	11423
	11431	11442	11450	11457	11467	11477	11487	11497	11507	11517	11526	11536	11545	11556	11566
	11577	11588	11598	11608	11618	11627	11637	11647	11657	11667	11677	11687	11697	11706	11716
	11725	11735	11744	11755	11765	11776	11786	11797	11807	11818	11828	11839	11849	11860	11870
	11880	11889	11899	11912	11915	11918	11921	11928	11934	11937	11943	11947	11953	11955	11961
	11967	11970	11973	11980	11987	11991	11998	12005	12012	12019	12026	12030	12033	12037	12041
	12048	12054	12062	12069											
BLKW	10647														
BYTE	179	180	185	186	194	195	203	204	205	206	246	247	257	258	265
	266	268	269	271	272	1078	1079	1080	1081	1082	1083	1124	1126	1129	1131
	1134	1136	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156
	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1172	1173	1174	1175
	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190
	1191	1192	1193	1194	1195	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210
	1211	1212	8481	8483	8486	8488	8491	8657	8659	8662	8664	8667	8835	8838	8840
	9010	9012	9015	9017	9020	9221	9223	9226	9228	9231	9434	9437	9439	9755	9757
	9759	9883	10199	10200	10206	10207	10439	10440	10576	10577	10578	10579	10776	10777	10778
ENABL	1														
END	14254														
ENDC	9	19	111	125	137	141	143	150	163	167	169	173	177	179	207
	217	225	226	227	228	232	235	257	265	268	271	274	275	276	277
	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292
	293	294	295	296	300	1297	1299	1301	1303	1304	1306	1307	1309	1330	1335
	1346	1348	1361	1362	1368	1369	1370	1384	1401	1402	1416	1417	1418	1448	1455
	1456	1460	1461	1462	1514	1520	1521	1531	1532	1533	1626	1627	1632	1633	1634
	1713	1717	1718	1737	1738	1739	1820	1831	1832	1851	1852	1853	1907	1917	1918
	1941	1942	1943	2004	2011	2012	2030	2031	2032	2074	2108	2109	2133	2134	2135
	2331	2341	2342	2366	2367	2368	2450	2451	2474	2475	2476	2558	2559	2582	2583
	2584	2665	2666	2676	2677	2678	2741	2742	2757	2758	2759	2822	2823	2838	2839
	2840	2904	2905	2926	2927	2928	3073	3079	3080	3101	3102	3103	3248	3252	3253
	3276	3277	3278	3399	3406	3407	3426	3427	3428	3546	3555	3556	3579	3580	3581
	3689	3702	3703	3726	3727	3728	3840	3850	3851	3874	3875	3876	3986	3997	3998
	4021	4022	4023	4134	4145	4146	4148	4149	4150	4181	4182	4195	4196	4197	4253
	4265	4266	4278	4279	4280	4337	4349	4350	4367	4368	4369	4426	4439	4475	4476
	4493	4494	4495	4552	4565	4600	4601	4607	4608	4609	4630	4640	4641	4642	4643
	4714	4715	4752	4753	4754	4788	4812	4813	4850	4851	4852	4886	4910	4911	4949
	4950	4951	4985	5003	5004	5042	5043	5044	5078	5096	5097	5135	5136	5137	5171
	5194	5195	5233	5234	5235	5269	5292	5293	5331	5332	5333	5367	5387	5388	5422
	5423	5424	5459	5479	5480	5516	5517	5518	5549	5555	5574	5575	5607	5608	5609
	5637	5643	5663	5664	5701	5702	5703	5726	5733	5740	5758	5759	5768	5769	5770
	5862	5863	5902	5903	5904	5982	5998	5999	6039	6040	6041	6151	6163	6164	6200
	6201	6202	6277	6278	6318	6319	6320	6418	6440	6441	6479	6480	6481	6546	6557
	6558	6598	6599	6600	6695	6712	6713	6751	6752	6753	6823	6824	6864	6865	6866
	6954	6970	6971	7017	7018	7019	7069	7085	7086	7131	7132	7133	7183	7199	7200
	7246	7247	7248	7298	7314	7315	7360	7361	7362	7412	7428	7429	7477	7478	7479
	7529	7545	7546	7595	7596	7597	7646	7662	7663	7709	7710	7711	7772	7788	7789
	7835	7836	7837	7897	7913	7914	7960	7961	7962	8022	8038	8039	8085	8086	8087
	8144	8160	8161	8207	8208	8209	8266	8282	8283	8329	8330	8331	8388	8407	8408
	8453	8454	8455	8531	8583	8584	8629	8630	8631	8704	8756	8757	8803	8804	8805
	8870	8910	8911	8956	8957	8958	9070	9122	9123	9168	9169	9170	9286	9338	9339
	9385	9386	9387	9478	9515	9516	9562	9563	9564	9635	9667	9668	9714	9715	9716
	9806	9838	9839	9840	9841	9852	9853	9854	9856	9859	9865	9868	9869	9873	9875
	9881	9883	9884	9887	10345	10348	10353	10357	10359	10370	10373	10374	10375	10377	10379
	10386	10390	10395	10397	10401	10404	10405	10408	10411	10420	10427	10432	10433	10434	10442
	10452	10456	10457	10460	10475	10504	10507	10584	10651	10680	10730	10731	10734	10761	10776



	10787	10793	10796	10815	10816	10817	10818	10819	10820	10821	10822	10826	10835	10836	10842
	10852	10853	10863	10865	10872										
EQUIV	19	20	28	73	74	75	76	77	78	79	80	81	82	101	102
EVEN	103	104	105	106	107	108	109	110							
IF	235	1025	1219	1335	8492	8668	8841	9021	9232	9440	9760	10503	10779	10871	12079
	5	17	83	111	125	135	140	142	149	162	165	167	172	176	178
	207	217	225	226	227	231	232	234	257	265	268	271	274	275	276
	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291
	292	293	294	295	296	300	1295	1297	1299	1301	1303	1304	1306	1307	1325
	1334	1344	1346	1360	1362	1368	1370	1383	1400	1402	1416	1418	1447	1454	1456
	1460	1462	1513	1519	1521	1531	1533	1625	1627	1632	1634	1712	1716	1718	1737
	1739	1819	1830	1832	1851	1853	1906	1916	1918	1941	1943	2003	2010	2012	2030
	2032	2073	2107	2109	2133	2135	2330	2340	2342	2366	2368	2449	2451	2474	2476
	2557	2559	2582	2584	2664	2666	2676	2678	2740	2742	2757	2759	2821	2823	2838
	2840	2903	2905	2926	2928	3072	3078	3080	3101	3103	3247	3251	3253	3276	3278
	3398	3405	3407	3426	3428	3545	3554	3556	3579	3581	3688	3701	3703	3726	3728
	3839	3849	3851	3874	3876	3985	3996	3998	4021	4023	4133	4144	4146	4148	4150
	4180	4182	4195	4197	4252	4264	4266	4278	4280	4336	4348	4350	4367	4369	4425
	4438	4474	4476	4493	4495	4551	4564	4599	4601	4607	4609	4629	4639	4641	4643
	4713	4715	4752	4754	4787	4811	4813	4850	4852	4885	4909	4911	4949	4951	4984
	5002	5004	5042	5044	5077	5095	5097	5135	5137	5170	5193	5195	5233	5235	5268
	5291	5293	5331	5333	5366	5386	5388	5422	5424	5458	5478	5480	5516	5518	5548
	5554	5573	5575	5607	5609	5636	5642	5662	5664	5701	5703	5725	5732	5739	5757
	5759	5768	5770	5861	5863	5902	5904	5981	5997	5999	6039	6041	6150	6162	6164
	6200	6202	6276	6278	6318	6320	6417	6439	6441	6479	6481	6545	6556	6558	6598
	6600	6694	6711	6713	6751	6753	6822	6824	6864	6866	6953	6969	6971	7017	7019
	7068	7084	7086	7131	7133	7182	7198	7200	7246	7248	7297	7313	7315	7360	7362
	7411	7427	7429	7477	7479	7528	7544	7546	7595	7597	7645	7661	7663	7709	7711
	7771	7787	7789	7835	7837	7896	7912	7914	7960	7962	8021	8037	8039	8085	8087
	8143	8159	8161	8207	8209	8265	8281	8283	8329	8331	8387	8406	8408	8453	8455
	8530	8582	8584	8629	8631	8703	8755	8757	8803	8805	8869	8909	8911	8956	8958
	9069	9121	9123	9168	9170	9285	9337	9339	9385	9387	9477	9514	9516	9562	9564
	9634	9666	9668	9714	9716	9805	9837	9839	9841	9851	9852	9853	9854	9855	9856
	9858	9864	9867	9869	9873	9875	9881	9883	9884	10344	10347	10352	10357	10369	10371
	10372	10373	10375	10376	10377	10386	10388	10396	10398	10403	10404	10405	10407	10410	10420
	10423	10430	10432	10433	10435	10442	10445	10452	10456	10457	10459	10474	10490	10506	10583
	10650	10671	10729	10731	10734	10761	10776	10786	10792	10796	10807	10816	10817	10818	10819
IFF	10820	10821	10822	10825	10835	10836	10841	10848	10853	10861	10863	10865	10869		
	17	141	143	150	163	167	169	173	176	179	207	232	235	1361	1362
	1369	1370	1384	1401	1402	1417	1418	1448	1455	1456	1461	1462	1514	1520	1521
	1532	1533	1626	1627	1633	1634	1713	1717	1718	1738	1739	1820	1831	1832	1852
	1853	1907	1917	1918	1942	1943	2004	2011	2012	2031	2032	2074	2108	2109	2134
	2135	2331	2341	2342	2367	2368	2450	2451	2475	2476	2558	2559	2583	2584	2665
	2666	2677	2678	2741	2742	2758	2759	2822	2823	2839	2840	2904	2905	2927	2928
	3073	3079	3080	3102	3103	3248	3252	3253	3277	3278	3399	3406	3407	3427	3428
	3546	3555	3556	3580	3581	3689	3702	3703	3727	3728	3840	3850	3851	3875	3876
	3986	3997	3998	4022	4023	4134	4145	4146	4149	4150	4181	4182	4196	4197	4253
	4265	4266	4279	4280	4337	4349	4350	4368	4369	4426	4439	4475	4476	4494	4495
	4552	4565	4600	4601	4608	4609	4630	4640	4641	4642	4643	4714	4715	4753	4754
	4788	4812	4813	4851	4852	4886	4910	4911	4950	4951	4985	5003	5004	5043	5044
	5078	5096	5097	5136	5137	5171	5194	5195	5234	5235	5269	5292	5293	5332	5333
	5367	5387	5388	5423	5424	5459	5479	5480	5517	5518	5549	5555	5574	5575	5608
	5609	5637	5643	5663	5664	5702	5703	5726	5733	5740	5758	5759	5769	5770	5862
	5863	5903	5904	5982	5998	5999	6040	6041	6151	6163	6164	6201	6202	6277	6278
	6319	6320	6418	6440	6441	6480	6481	6546	6557	6558	6599	6600	6695	6712	6713
	6752	6753	6823	6824	6865	6866	6954	6970	6971	7018	7019	7069	7085	7086	7132







	3	125	232	1309	209	210	211	212	213	214	215	216	217	218	219
MCALL	3	125	232	1309	209	210	211	212	213	214	215	216	217	218	219
NLIST	1	125	132	207	209	210	211	212	213	214	215	216	217	218	219
	220	221	222	223	224	225	232	235	1219	1225	1309	1335	1360	1370	1400
	1418	1454	1462	1519	1533	1625	1634	1716	1739	1830	1853	1916	1943	2010	2032
	2107	2135	2340	2368	2449	2476	2557	2584	2664	2678	2740	2759	2821	2840	2903
	2928	3078	3103	3251	3278	3405	3428	3554	3581	3701	3728	3849	3876	3996	4023
	4144	4150	4180	4197	4264	4280	4348	4369	4474	4495	4599	4609	4639	4643	4713
	4754	4811	4852	4909	4951	5002	5044	5095	5137	5193	5235	5291	5333	5386	5424
	5478	5518	5573	5609	5662	5703	5757	5770	5861	5904	5997	6041	6162	6202	6276
	6320	6439	6481	6556	6600	6711	6753	6822	6866	6969	7019	7084	7133	7198	7248
	7313	7362	7427	7479	7544	7597	7661	7711	7787	7837	7912	7962	8037	8087	8159
	8209	8281	8331	8406	8455	8582	8631	8755	8805	8909	8958	9121	9170	9337	9387
	9514	9564	9666	9716	9837	9841	9859	9875	10352	10452	10807	10815	10816	10817	10818
	10819	10820	10822	10823	12095	12123	12130	12138	12145	12152	12160	12178	12185	12192	12199
	12206	12224	12231	12240	12247	12255	12273	12280	12288	12295	12302	12318	12325	12333	12340
	12347	12353	12360	12367	12375	12394	12401	12409	12416	12423	12430	12437	12443	12461	12468
	12476	12483	12489	12507	12514	12522	12529	12535	12552	12559	12567	12574	12580	12597	12604
	12612	12619	12625	12643	12650	12658	12665	12671	12688	12695	12703	12710	12716	12733	12741
	12749	12756	12762	12779	12786	12794	12801	12807	12824	12831	12839	12847	12853	12870	12877
	12885	12892	12898	12915	12922	12930	12937	12944	12961	12968	12976	12983	12989	13006	13013
	13021	13029	13035	13052	13059	13067	13074	13080	13095	13104	13113	13121	13129	13136	13143
	13150	13158	13166	13174	13188	13198	13207	13215	13223	13230	13237	13244	13252	13260	13268
	13282	13291	13300	13308	13316	13323	13330	13337	13345	13353	13362	13376	13385	13394	13402
	13410	13417	13424	13431	13439	13447	13455	13469	13478	13487	13495	13503	13511	13518	13525
	13533	13541	13549	13563	13572	13581	13589	13597	13604	13611	13618	13626	13634	13642	13657
	13666	13675	13683	13691	13698	13705	13712	13720	13728	13736	13750	13759	13768	13776	13785
	13792	13799	13806	13814	13822	13830	13845	13853	13861	13869	13884	13892	13901	13909	13924
	13932	13940	13948	13963	13971	13979	13987	14001	14010	14018	14026	14040	14048	14056	14064
	14078	14086	14094	14102	14118	14126	14134	14142	14157	14165	14173	14181	14196	14204	14213
	14221	14238	14245												
PAGE	170	300	1025	1286	1359	4706	5854	9849	9887	10342	10873	12094			
REPT	132	209	217												
SBTTL	15	126	135	138	160	170	232	300	1286	1360	1400	1454	1519	1625	1716
	1830	1916	2010	2107	2340	2449	2557	2664	2740	2821	2903	3078	3251	3405	3554
	3701	3849	3996	4144	4180	4264	4348	4474	4599	4639	4706	4713	4811	4909	5002
	5095	5193	5291	5386	5478	5573	5662	5757	5854	5861	5997	6162	6276	6439	6556
	6711	6822	6969	7084	7198	7313	7427	7544	7661	7787	7912	8037	8159	8281	8406
	8582	8755	8909	9121	9337	9514	9666	9837	9849	10342	10405	10457	10504	10581	10648
	10727	10784	10807	10823	10873	10977	11910	12077	12094						
TITLE	4														
WORD	132	133	134	154	155	156	157	158	159	168	178	181	182	183	184
	187	188	189	190	191	192	193	196	197	198	207	209	210	211	212
	213	214	215	216	217	218	219	220	221	222	223	224	237	238	239
	240	241	242	243	244	248	249	250	263	267	270	273	274	275	276
	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291
	292	293	294	295	1071	1072	1073	1074	1075	1076	1077	1085	1086	1087	1088
	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1214	1215	1216	1821
	1822	1823	1825	1826	1827	1911	1912	2096	2097	2098	2099	2332	2333	4768	4866
	4965	5058	5151	5249	5347	5720	5798	6215	6331	6384	6492	6614	6668	6763	6877
	6928	7726	7736	7851	7861	7976	7986	8101	8133	8223	8255	8345	8377	8477	8534
	8535	8536	8537	8538	8539	8540	8541	8542	8543	8544	8545	8546	8547	8548	8549
	8550	8551	8552	8553	8554	8555	8556	8557	8558	8559	8560	8561	8562	8563	8564
	8565	8653	8707	8708	8709	8710	8711	8712	8713	8714	8715	8716	8717	8718	8719
	8720	8721	8722	8723	8724	8725	8726	8727	8728	8729	8730	8731	8732	8733	8734
	8735	8736	8737	8738	8831	8873	8874	8875	8876	8877	8878	8879	8880	8881	8882
	8883	8884	8885	8886	8887	8888	9006	9057	9073	9074	9075	9076	9077	9078	9079

9080	9081	9082	9083	9084	9085	9086	9087	9088	9089	9090	9091	9092	9093	9094
9095	9096	9097	9098	9099	9100	9101	9102	9103	9104	9217	9273	9289	9290	9291
9292	9293	9294	9295	9296	9297	9298	9299	9300	9301	9302	9303	9304	9305	9306
9307	9308	9309	9310	9311	9312	9313	9314	9315	9316	9317	9318	9319	9320	9430
9465	9481	9482	9483	9484	9485	9486	9487	9488	9489	9490	9491	9492	9493	9494
9495	9496	9638	9639	9640	9641	9642	9643	9644	9645	9751	9792	9809	9810	9811
9812	9813	9814	9815	9816	9864	9867	9882	10008	10039	10092	10093	10094	10095	10184
10483	10488	10580	10677	10724	10759	10814	10862	10864	12080	12082	12085	12087	12089	12090
12124	12131	12139	12146	12153	12161	12179	12186	12193	12200	12207	12225	12232	12241	12248
12256	12274	12281	12289	12296	12303	12319	12326	12334	12341	12348	12354	12361	12368	12376
12395	12402	12410	12417	12424	12431	12438	12444	12462	12469	12477	12484	12490	12508	12515
12523	12530	12536	12553	12560	12568	12575	12581	12598	12605	12613	12620	12626	12644	12651
12659	12666	12672	12689	12696	12704	12711	12717	12734	12742	12750	12757	12763	12780	12787
12795	12802	12808	12825	12832	12840	12848	12854	12871	12878	12886	12893	12899	12916	12923
12931	12938	12945	12962	12969	12977	12984	12990	13007	13014	13022	13030	13036	13053	13060
13068	13075	13081	13096	13105	13114	13122	13130	13137	13144	13151	13159	13167	13175	13189
13199	13208	13216	13224	13231	13238	13245	13253	13261	13269	13283	13292	13301	13309	13317
13324	13331	13338	13346	13354	13363	13377	13386	13395	13403	13411	13418	13425	13432	13440
13448	13456	13470	13479	13488	13496	13504	13512	13519	13526	13534	13542	13550	13564	13573
13582	13590	13598	13605	13612	13619	13627	13635	13643	13658	13667	13676	13684	13692	13699
13706	13713	13721	13729	13737	13751	13760	13769	13777	13786	13793	13800	13807	13815	13823
13831	13846	13854	13862	13870	13885	13893	13902	13910	13925	13933	13941	13949	13964	13972
13980	13988	14002	14011	14019	14027	14041	14049	14057	14065	14079	14087	14095	14103	14119
14127	14135	14143	14158	14166	14174	14182	14197	14205	14214	14222	14239	14246		

ABS. 062350 000

ERRORS DETECTED: 0

DSKM: DQKUA.A, DSKW: DQKUA.A=DQKUA.A. P11, DQKUA.A.MIC  
RUN-TIME: 39 41 6 SECONDS  
RUN-TIME RATIO: 307/87=3.5  
CORE USED: 23K (45 PAGES)

DOCUMENT PAGES: 302  
WRAP-AROUND: 0%

USER SYMBOLS: 710  
MACRO NAMES: 58  
UNDF SYMBOLS: 13  
DISK BLOCKS READ: 2214  
DISK BLKS WRITTEN: 1601  
KILO CORE SECONDS: 2922