

SDS 900891C

\$1.00

DIAGNOSTIC PROGRAM MANUAL
SIGMA 5 AND 7
CPU DIAGNOSTIC SYSTEM (PATTERN)
PROGRAM NO. 704043C

October 1968

This publication supersedes SDS 900891B,
dated June 1968

SCIENTIFIC DATA SYSTEMS • 701 South Aviation Boulevard • El Segundo, Calif., 90245 • 213/772-4511

© 1967, 1968, Scientific Data Systems, Inc.

LIST OF EFFECTIVE PAGES

Total number of pages is 34, as follows:

Page No.	Issue	Page No.	Issue
Title	Original		
A	Original		
i thru ii	Original		
1-1 thru 1-2	Original		
2-1 thru 2-6	Original		
3-1 thru 3-4	Original		
4-1 thru 4-18	Original		

TABLE OF CONTENTS

Section	Title	Page
I	INTRODUCTION	1-1
	1-1 Scope of Manual	1-1
	1-2 Program Objective	1-1
	1-3 General Specifications	1-1
II	OPERATING INSTRUCTIONS	2-1
	2-1 General	2-1
	2-2 Loading Procedure	2-1
	2-3 Success Indications	2-1
	2-4 Error Indications and Recovery Procedures	2-1
	2-5 Options	2-2
	2-6 Loop on Current Test	2-2
	2-7 Repeat Test with Current Parameters	2-2
	2-8 Halt to Change Register Page Limits or Memory Limits	2-2
	2-9 No Halt on Errors	2-2
	2-10 Program Reinitialization	2-2
III	PROGRAM DESCRIPTION	3-1
	3-1 General	3-1
	3-2 Individual Register Test	3-1
	3-3 Register Page Addressing Test	3-1
	3-4 Memory Access Test	3-1
	3-5 Memory Execution Test	3-1
	3-6 Flow Charts	3-1
IV	PROGRAM LISTING	4-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
2-1	Error Analysis Flow Chart	2-4
2-2	Simplified Logic Diagram	2-5
3-1	Pattern Program Flow Chart	3-2

LIST OF TABLES

Table	Title	Page
1-1	General Specifications	1-1
1-2	Testing Prerequisites	1-1
2-1	Switch Settings for Program Loading	2-1
2-2	LT26 Switch Settings	2-2
2-3	Register Block Module Locations	2-3

RELATED PUBLICATIONS

<u>Publication Title</u>	<u>Publication No.</u>
Sigma 5 Computer, Reference Manual	900959
Sigma 5 Computer, Technical Manual	901172
Sigma 7 Computer, Reference Manual	900950
Sigma 7 Computer, Technical Manual	901060
Sigma Symbol and Meta-Symbol, Reference Manual	900952

SECTION I
INTRODUCTION

1-1 SCOPE OF MANUAL

This manual describes the Pattern program of the CPU Diagnostic System designed for the Sigma 5 and Sigma 7 computers, manufactured by Scientific Data Systems.

This manual is made up of four sections. Section I is a general introduction to the Pattern program. Section II contains a discussion of operating procedures. Section III contains a detailed description of the program operation. Section IV contains the program's complete symbolic listing as generated by the Sigma Metasymbol Assembler.

1-2 PROGRAM OBJECTIVE

The purpose of the Pattern program is to detect and report any malfunctioning in any general register of each register page implemented, and to test the ability of the CPU to access and execute instructions from each core memory address.

1-3 GENERAL SPECIFICATIONS

Table 1-1 lists the general specifications for this program.

Table 1-2 shows the testing that must have been successfully completed before the Pattern program is run. Also listed are the testing prerequisites for the other Sigma 5/7 CPU diagnostic programs.

Table 1-1. General Specifications

Computer configuration	Any Sigma 5 or Sigma 7 computer with card reader or paper tape reader for program input
Memory size	4K minimum (4096 words)
Optional equipment	None

Table 1-2. Testing Prerequisites

Program	Prerequisite Program
Sense (Sigma 7 only)	None
Verify	None
Pattern	Verify
Auto	Verify, Pattern*
Suffix	Auto
Float	Auto
Interrupt	Auto
Memory Protect	Suffix
Map (Sigma 7 only)	Suffix
*For the Auto test to run, the block 0 register must be functioning correctly, as tested by the Pattern program	

SECTION II
OPERATING INSTRUCTIONS

2-1 GENERAL

The Pattern diagnostic program consists of two general test sections: the register block test and the memory test. Each of these two sections in turn consists of two subsections. The first subsection of the register block test is to detect any error in any register of register blocks implemented (called type 1 errors). The second subsection is to report any errors (called type 2 errors) caused by interaction between the register blocks.

The memory test consists of the access test (type 3 errors) and the execution test (type 4 errors). The purpose is to reference all available memory locations as operand address and attempt to execute an instruction from each location.

2-2 LOADING PROCEDURE

Table 2-1 shows the control panel switch settings to be used for loading the program.

After the switches have been set as indicated in table 2-1, the following procedure is required:

- a. Place the COMPUTE switch in the IDLE position.
- b. Place the program card deck (or paper tape) in the appropriate reader and start the reader.
- c. Clear memory by pressing the CPU RESET/CLEAR and SYS RESET/CLEAR buttons simultaneously.
- d. Set the UNIT ADDRESS switches to the address of the input device.

Table 2-1. Switch Settings for Program Loading

Switch	Setting
CONTROL MODE	LOCAL
WATCHDOG TIMER	NORMAL
INTERLEAVE SELECT	NORMAL
PARITY ERROR MODE	CONT
AUDIO	ON
CLOCK MODE	CONT
ADDR STOP	Off
SENSE Switches	0

e. Press the LOAD switch.

f. Place the COMPUTE switch in the RUN position.

If the program is loaded with the switches set according to table 2-1, it will automatically branch to the starting location and begin running.

2-3 SUCCESS INDICATIONS

Provided that no errors occur, and if SENSE switches 1, 3, and 4 are all reset during the execution, the program will run continuously in cyclical fashion. A more positive determination of successful operation is obtained by setting SENSE switch 3, which brings the program to halt at a specified location, and then examining the pass count in general register 4.

See pages 2 and 3 of the program listing for a summary of SENSE switch indications and other displayed information.

2-4 ERROR INDICATIONS AND RECOVERY PROCEDURES

The program requires that general register 0 of page 0 is functioning properly. Any errors associated with this register will halt the program at error waits, the locations of which are specified on page 5 of the program listing under the heading of Additional Error Wait Locations.

The program cannot be continued further until register 0 is corrected. If replacing page 0 modules does not correct the malfunction, there could be interaction with either the REU (Register Extension Unit) or other internal pages.

Whenever program halt occurs due to a possible REU interaction, general register 0 will contain zeros in one or more bits of the interacting byte (or bytes). The operator is advised to pull out the BCR module in the CPU associated with failing byte. If the problem is still present, remove the modules for pages 1, 2, or 3 or the CPU. Sigma 5 may be affected by IOFM modules (see table 2-3 for all module locations).

If removing the BCR module does rectify the register 0 error, reinsert the module and follow the procedure given below to pinpoint the fault in the REU which causes the problem.

- a. Ensure that the LT26 switches are set according to table 2-2 (LT26 module sets REU address).
- b. If only some but not all bits in register 0 are failing, the most probable error is interference from page 4, 5, 6, or

7 in the REU at the faulty bits. Refer to table 2-3 for module locations. If, however, all bits of a byte (or bytes) of register 0 are in error, proceed according to the error analysis flow chart, figure 2-1, and perform the following test whenever called for after removing a given module.

TEST: Check register 0 to see if the error is corrected. This is done by storing FFFFFFFF₁₆ in register 0, followed by a display of register 0. (It must be ensured that none of the display lamps are burned out.)

If register 0 still shows the error, reinsert the removed module and continue with the procedure in the flow chart from the point the branch was made to this test.

If, however, register 0 appears to be corrected, the removed module is a possible source of the error. Replace the module with a new one and examine register 0 again.

If the error reappears after replacing the module, refer to Logic Equations (Dwg. No. 124817) or the simplified logic diagram, figure 2-2, for further troubleshooting.

Any error involving other registers in any page will terminate the program at error wait location 1E3. A complete set of error information as to the types and the source is available to the operator on page 4 of the program listing. Module locations are given in table 2-3.

Similarly, any core memory operand access errors or instruction execution errors will halt the program at the same wait location, 1E3. See page 5 of the listing for information associated with these errors.

Table 2-2. LT26 Switch Settings

Register Block Numbers	S3-2	S3-1	S2-2
4 through 7	0	0	1
8 through 11	0	1	0
12 through 15	0	1	1
16 through 19	1	0	0
20 through 23	1	0	1
24 through 27	1	1	0
28 through 31	1	1	1

Note: S2-1, S1-2, S1-1, S4-1, and S4-2 settings are irrelevant

If the loop on current test option (see paragraph 2-6) is invoked before clearing the wait, the error halt will not occur during the loop.

2-5 OPTIONS

Options by means of SENSE switches are incorporated into the Pattern program to give the operator a more flexible tool for diagnosing failures.

2-6 LOOP ON CURRENT TEST

If SENSE switch 1 is set following an error, the test that detected a failure continues to repeat until SENSE switch 1 is reset.

2-7 REPEAT TEST WITH CURRENT PARAMETERS

If SENSE switch 2 is reset, the program calculates the number of implemented register pages and the core memory configuration on the first pass only and uses these values in the subsequent passes. If, however, SENSE switch 2 is set, the program recalculates on every pass. Switch 2 should thus be reset whenever the parameter change option is used (see paragraph 2-8).

2-8 HALT TO CHANGE REGISTER PAGE LIMITS OR MEMORY LIMITS

By setting SENSE switch 3, the operator halts the program at the end of the current program execution pass at location 1AB. At this time, the following information is displayed on the specified registers:

Register	Contents
0	Value of starting test page shown in bits 23-27
1	Last page implemented shown in bits 23-27
2	Minimum test core address
3	Maximum test core address
4	Pass count
5	Error count

Before proceeding, the contents of registers 0, 1, 2, or 3 may be altered to suit the operator's immediate need. The previous option (paragraph 2-7) is normally invoked in conjunction with this option.

2-9 NO HALT ON ERRORS

If SENSE switch 4 is set, no error wait will be recognized by the program except errors associated with register 0. The error count is maintained independent of this option.

2-10 PROGRAM REINITIALIZATION

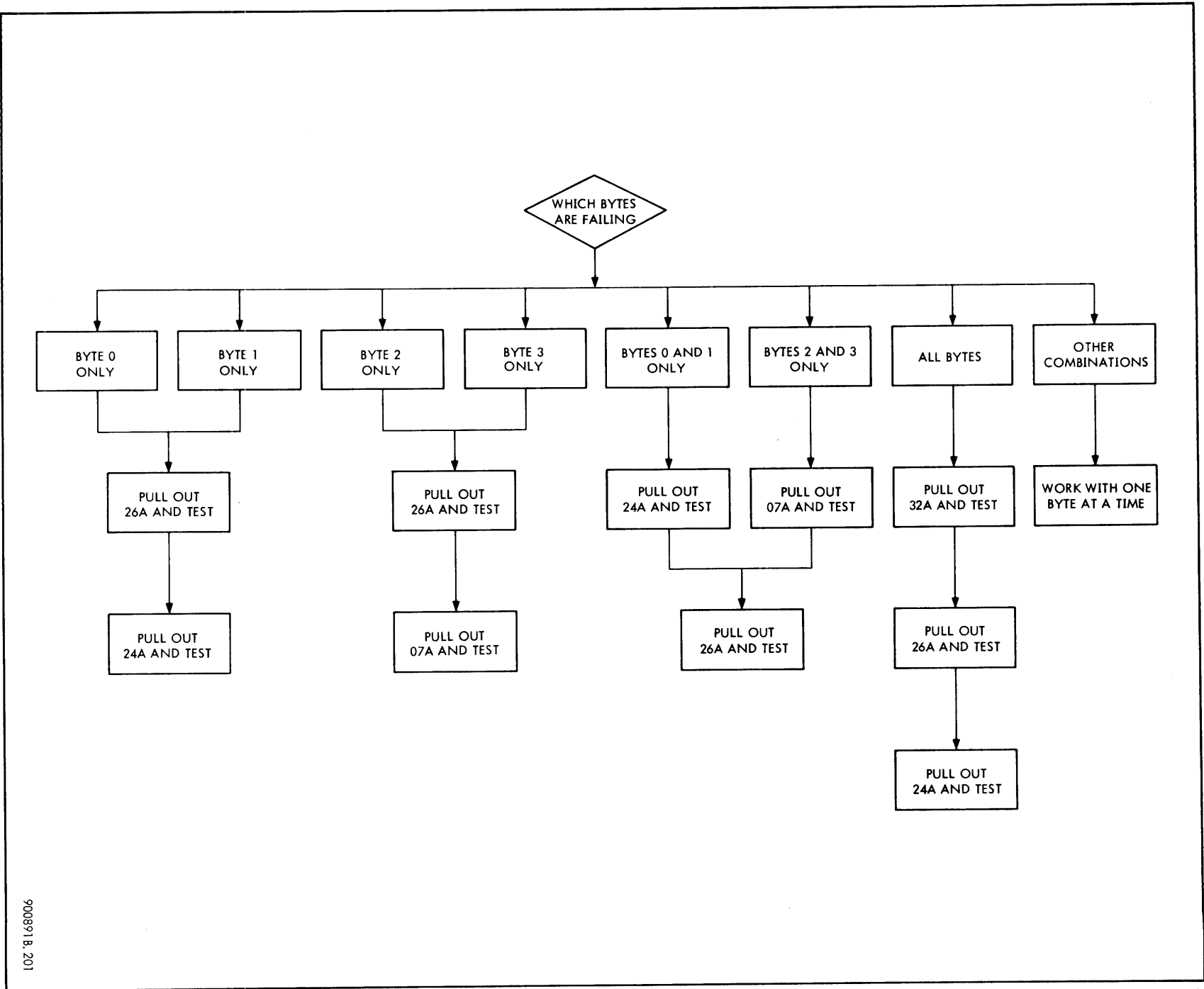
With the COMPUTE switch at IDLE position, pressing the SYSTEM RESET/CLEAR button returns the program to the starting point. Reinitialization is also achieved by the execution of PCP interrupt.

Table 2-3. Register Block Module Locations

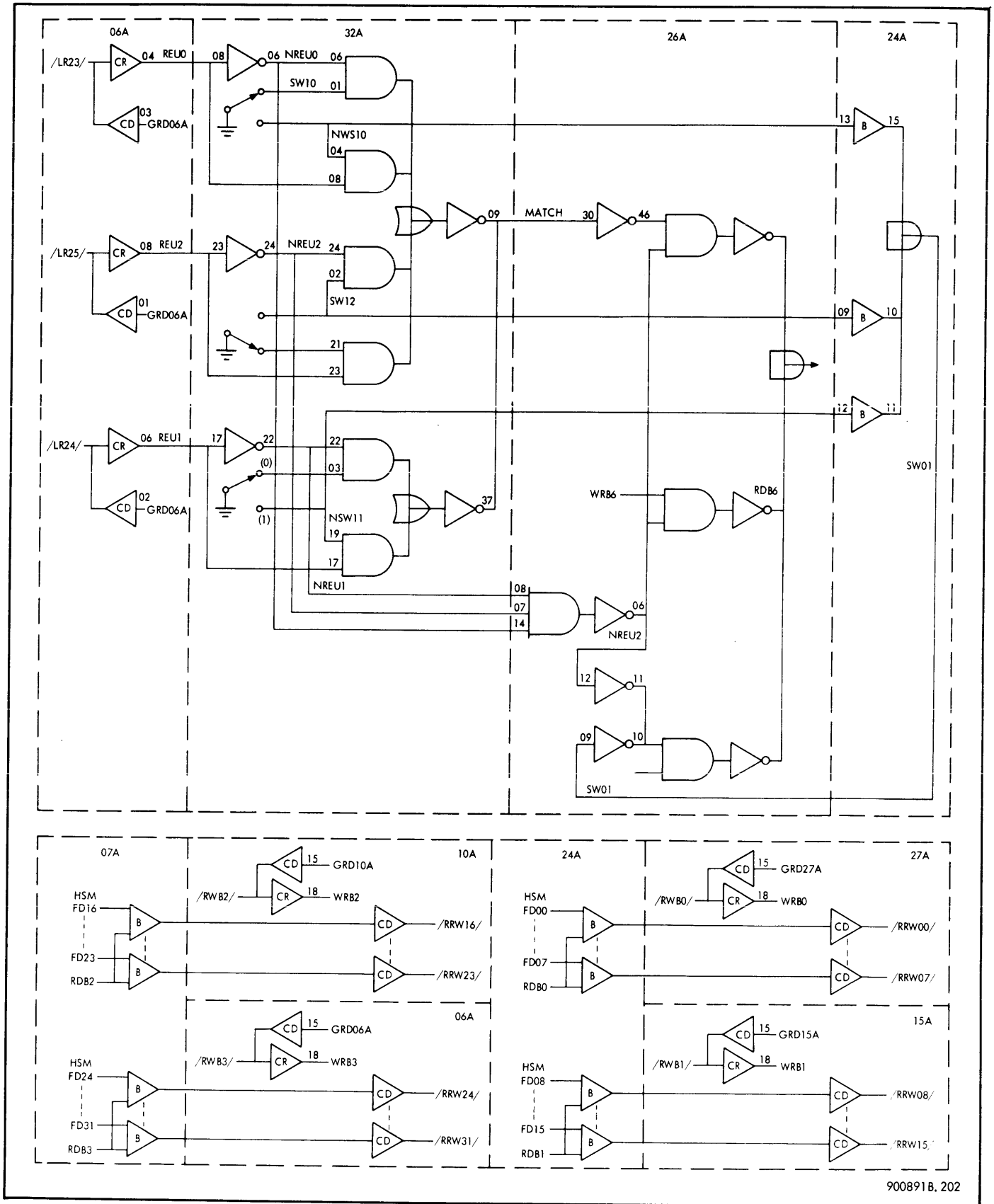
CPU									REU				
Reg. Page No.	Sigma 5				Sigma 7				Register Page (Block) Number	Sigma 5/7			
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 0	Byte 1	Byte 2	Byte 3		Byte 0	Byte 1	Byte 2	Byte 3
0	09K	19K	05S	09T	01T	15T	01L	01Q	4, 8, 12, 16, 20, 24, 28	23A	19A	14A	04A
1	10K	18K	06S	10T	02T	16T	02L	02Q	5, 9, 13, 17, 21, 25, 29	22A	18A	13A	03A
2	11K	17K	07S	11T	03T	17T	03L	03Q	6, 10, 14, 18, 22, 26, 30	21A	17A	12A	02A*
3	12K	16K	08S	12T	04T	18T	04L	04Q	7, 11, 15, 19, 23, 27, 31	20A	16A	11A	01A†
BCR	13K	15K	09S	13T	05T	19T	05L	05Q	*Remove XT10 from 05A † Remove XT10's from 05A, 09A				

IOFM	05K	23K	01S	01T
	06K	22K	02S	02T
	07K	21K	03S	03T
	08K	20K	04S	04T
				05T
				06T
				07T
				08T

Figure 2-1. Error Analysis Flow Chart



900891 B, 201



900891B.202

Figure 2-2. Simplified Logic Diagram

SECTION III PROGRAM DESCRIPTION

3-1 GENERAL

After the program load, program execution begins with testing the proper operation of register 0 and then initialization and determination of the number of register pages implemented in the test computer. This is accomplished by loading a register in each page with X'555555', then examining the register for the content. A positive nonzero content indicates implementation. The value of the maximum implemented pages is stored for future reference.

The core memory size is determined by accessing each location until a nonexistent memory trap occurs at maximum address plus 1. The maximum address is also stored for future reference.

All above parameters can be altered by the operator (see paragraph 2-8).

The actual Pattern program tests are described in paragraphs 3-2 through 3-5.

3-2 INDIVIDUAL REGISTER TEST

Register 0 is independently tested at the beginning of the program. All other registers of all pages are then tested individually in turn. This is done by first storing a number pattern (called test pattern) in the register under test, followed by a routine which loads all remaining registers of the test page with the complement of the test pattern. The test register is then examined for containing the test pattern.

3-3 REGISTER PAGE ADDRESSING TEST

The purpose of this test is to detect errors arising when one page is affected by addressing another page. The test begins by loading all registers of all the implemented pages with ones. Then starting with the first page (test page), registers 0 and 8 of this page are cleared to zero. The

register pointer is then advanced to scan succeeding pages one by one with registers 0 and 8 of each page tested for retaining the previously stored ones.

If no error is reported, the test page is incremented by one and the register pointer is set to the test page. Once again, registers 0 and 8 are cleared, followed by examination of succeeding pages. The process is repeated until all the implemented pages are tested.

3-4 MEMORY ACCESS TEST

Each memory location, from address 300_{16} to maximum, is set equal to its own address. All test memory is then sequentially read, with the content of each location compared against an independent counter that is incremented for each sequential memory access. Errors precipitate a halt at the common error wait.

3-5 MEMORY EXECUTION TEST

All of test memory is preloaded with an unconditional branch to an error routine. Starting with the lowest test address and proceeding sequentially through core to the maximum test address, each test location is loaded with a BIR instruction whose R field equals C and whose effective address is set to the normal program return. Register C is set to $7FFFFFFF_{16}$ and program control is transferred to the test location.

Successful execution of the BIR increments register C once and causes a branch back to the test program, where the process is repeated for the next memory location. Failure of the BIR to execute properly results in a branch to error; the program ultimately halts at the common error wait.

3-6 FLOW CHARTS

Figure 3-1 is a flow chart of the entire Pattern program.

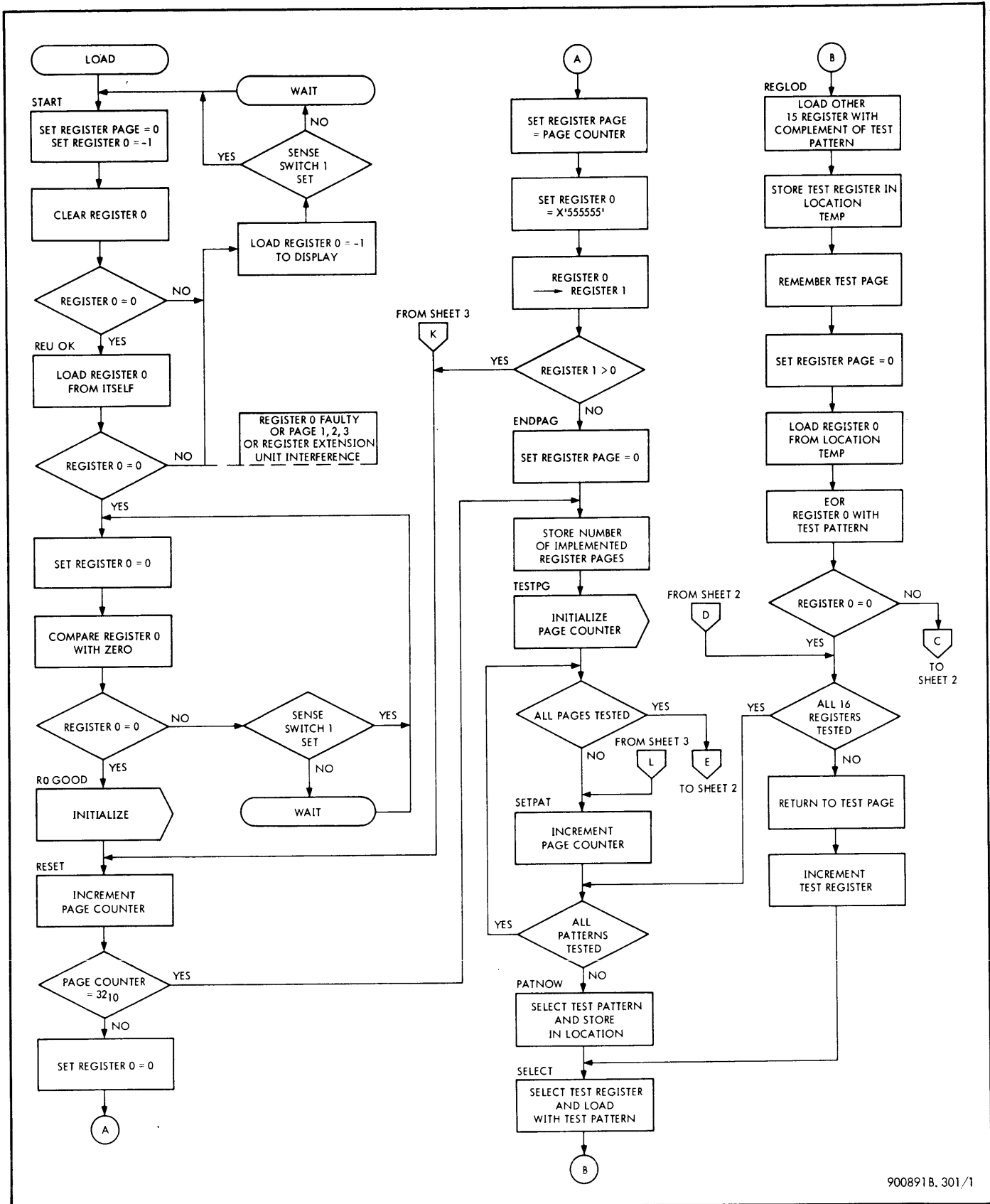
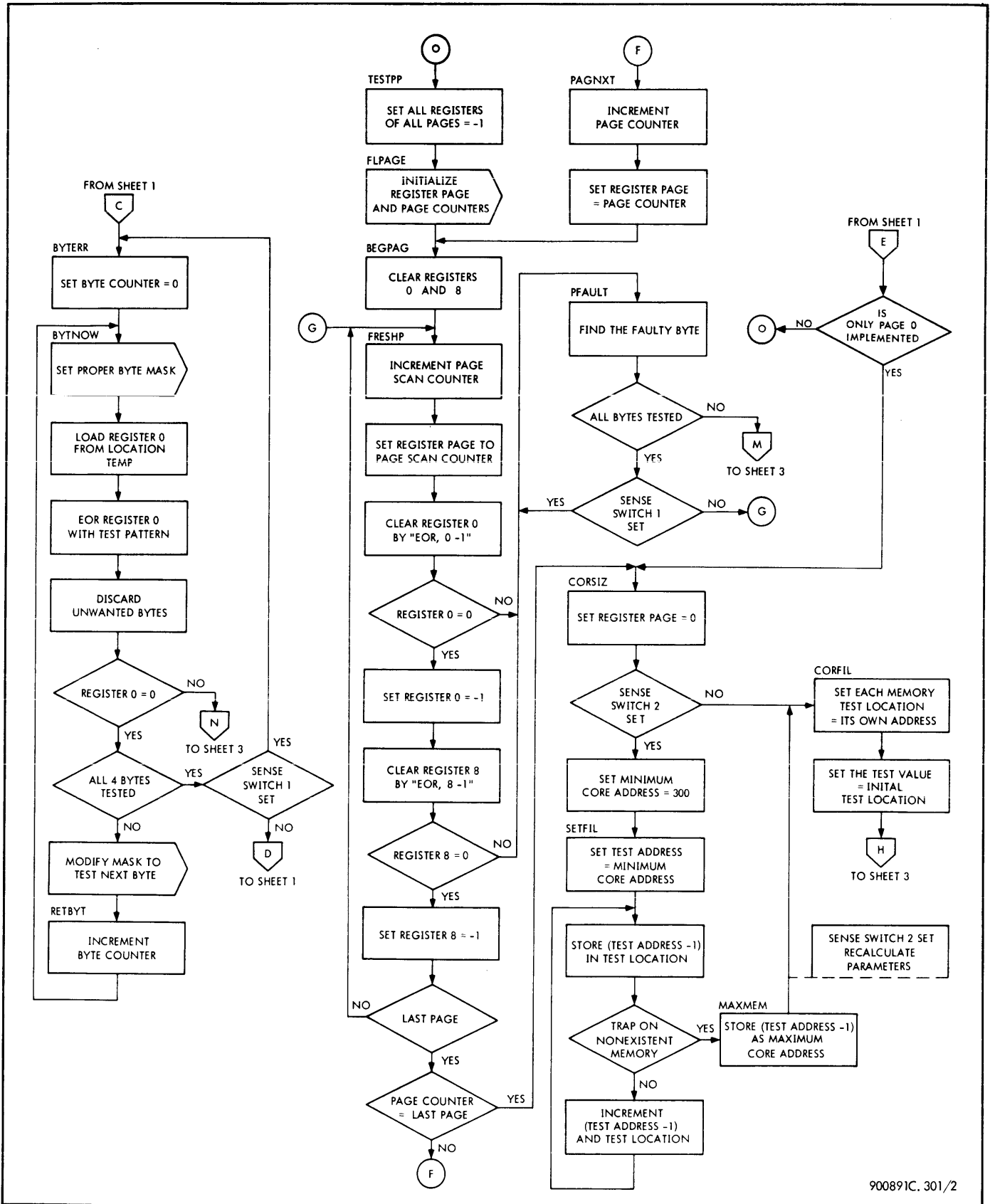


Figure 3-1. Pattern Program Flow Chart (Sheet 1 of 3)



900891C. 301/2

Figure 3-1. Pattern Program Flow Chart (Sheet 2 of 3)

SDS 900891

SECTION IV
PROGRAM LISTING

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

0000000A
0000000B
0000000C
0000000D
0000000E
0000000F

SYSTEM SIGZEDP
 * REVISION C00 = REVISION DATE 1 OCT, 1968 PROGRAM REVISED TO RUN WITH
 * 1 REGISTER PAGE (PAGE 0), OR MORE. PREVIOUSLY IT REQUIRED MULTIPLE
 * PAGES;

A EQU X'1A'
 B EQU X'1B'
 C EQU X'1C'
 D EQU X'1D'
 E EQU X'1E'
 F EQU X'1F'

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

* PROGRAM CONTROL AND DISPLAY INFORMATION

SENSE SWITCH	CONDITION	ACTION
--------------	-----------	--------

1	RESET SET	NORMAL OPERATION LOOP ON CURRENT TEST (SS4 MUST BE ON TO BYPASS THE ERROR-WAIT)
2	RESET SET	REPEAT TEST WITH CURRENT PAGE LIMITS AND/OR WITH CURRENT CASE PARAMETERS RECALCULATE REGISTER-PAGE LIMITS AND/OR CORE SIZE LIMITS
3	RESET SET	NORMAL OPERATION HALT TO CHANGE MEMORY LIMITS AND/OR RP LIMITS (SEE NEXT PAGE)
4	RESET SET	HALT ON ERRORS NO HALT ON ERRORS

39 PAGE

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

*
* PCP INTERRUPT - REINITIALIZE AND EXECUTE PROGRAM.* SENSE SWITCH 3 SET HALTS THE PROGRAM AT LOCATION LAB1. THE FOLLOWING
* REGISTERS WILL DISPLAY THE SPECIFIED INFORMATION :*

* REGISTER CONTENTS

* -----

* 0 VALUE OF STARTING PAGE SHOWN IN BITS 20-27
* 1 MAXIMUM PAGES IMPLEMENTED (BITS 20-27)
* 2 MINIMUM TEST CORE ADDRESS
* 3 MAXIMUM TEST CORE ADDRESS
* 4 PASS COUNT
* 5 ERROR COUNT

* REGISTERS 0-3 MAY BE ALTERED PRIOR TO CLEARING THE HALT

58 PAGE

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

* AS A RESULT OF ANY REGISTER-PAGE ERRORS :-
* THE FOLLOWING ERROR INFORMATION IS DISPLAYED IN THE SPECIFIED
* REGISTERS SUBSEQUENT TO THE ERROR HALT AT LOCATION LABELLED 'ERHALT'

* REGISTER CONTENTS

* -----

* 0 ERROR IDENTIFIER AND LOCATION
* 10R00PPR = (TYPE 1) REGISTER PAGE ERROR
* FAULTY BYTE R IN REGISTER R OF PAGE PP
* 20F00PPH = (TYPE 2) MISADDRESSED PAGE PP = ERROR
* IN BYTE R OF REGISTER R* 1 MAXIMUM PAGES IMPLEMENTED (APPLIES TO ALL TYPES)
* 2 CURRENT PATTERN (APPLIES TO ERROR-TYPE 1 ONLY)
* 3 RECEIVED PATTERN (APPLIES TO ERROR-TYPE 1 ONLY)
* 4 DIFFERING BITS IN R2 AND R4 (ERROR-TYPE 1 ONLY)* 5 DESIRED PAGE --- (APPLIES TO ERROR-TYPE 2 ONLY)
* 00000PPD = PAGE DESIRED WAS PP IN ADDRESSING* 'PP' VALUE RANGES ARE 00-1F FOR SIGMA 71 00-0F FOR SIGMA 5
* FOR MODULE LOCATIONS REFER TO TABLE '2-3' OF THE MANUAL

84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119

PAGE

* AS A RESULT OF ANY MEMORY ERROR :-
* THE FOLLOWING ERROR INFORMATION IS DISPLAYED IN THE SPECIFIED
* REGISTERS SUBSEQUENT TO THE ERROR HALT AT LOCATION '1F3'

REGISTER CONTENTS

0	ERROR IDENTIFIER AND LOCATION 300YYYYY * (TYPE 3) ACCESS ERROR AT CORE ADDRESS YYYY
	400YYYYY * (TYPE 4) EXECUTION ERROR AT CORE ADDRESS YYYY
1	MAXIMUM PAGE IMPLEMENTED
2	MAXIMUM CORE ADDRESS
3	ACCUMULATIVE ERROR COUNT
4	NUMBER OF PROGRAM PASSES
5	ERRONEOUS RESULT
6	PREDETERMINED RESULT
7	DIFFERENCE BETWEEN R5 AND R6

ADDITIONAL ERROR WAIT LOCATIONS:

* HALT AT LOCATION '06A' SIGNIFIES FAULTY REG.0, OR PAGES 1,2,OR3, OR
* REU NO.1 INTERFERENCE, FAULTY BITS INDICATED BY ZEROS IN REGISTER 0
* FOR CORRECTIVE ACTION SEE PARAGRAPH 2-4 OF THE MANUAL

* HALT AT LOCATION '071' SIGNIFIES FAULTY R0 AT BITS SHOWN BY 1

* HALTS AT ALL ODD LOCATIONS '24F' TO '277' INDICATE SPECIFIC
* SPURIOUS TRAP OR INTERRUPT, SEE LISTING FOR IDENTIFICATION

120
121 01 00040
122 01 00040
123
124
125 01 00040 0F4001EC
126 01 00041 0F0001FA
127 01 00042 0F0001FF
128 01 00043 0F000202
129 01 00044 0F000206
130 01 00045 0F00020A
131 01 00046 0F00020E
132 01 00047 0F000222
133 01 00048 0F000212
134 01 00049 0F000216
135 01 0004A 0F00021A
136 01 0004B 0F00021E
137 01 0004C 0F000222
138 01 0004D 0F000222
139 01 0004E 0F000222
140 01 0004F 0F000222

PAGE
BRG X'40'

* TRAP LOCATIONS

XPSD,4	NONOP
XPSD,0	UNIMP
XPSD,0	STACK
XPSD,0	8FL0
XPSD,0	FL0AT
XPSD,0	DEC
XPSD,0	TIMER
XPSD,0	TRAPUN
XPSD,0	CALL1
XPSD,0	CALL2
XPSD,0	CALL3
XPSD,0	CALL4
XPSD,0	TRAPUN
XPSD,0	TRAPUN
XPSD,0	TRAPUN

141 PAGE
 142
 143 *
 144 *
 * INTERRUPT LOCATIONS
 *
 145 01 00050 0F000226 XPSD,0 POWON
 146 01 00051 0F00022A XPSD,0 POWOFF
 147 01 00052 3300022E MTW,0 PULSE1
 148 01 00053 3300022F MTW,0 PULSE2
 149 01 00054 33000230 MTW,0 PULSE3
 150 01 00055 33000231 MTW,0 PULSE4
 151 01 00056 0F000232 XPSD,0 MEMPAR
 152 01 00057 0F000236 XPSD,0 INTUN
 153 01 00058 0F00023A XPSD,0 COUNT1
 154 01 00059 0F00023E XPSD,0 COUNT2
 155 01 0005A 0F000242 XPSD,0 COUNT3
 156 01 0005B 0F000246 XPSD,0 COUNT4
 157 01 0005C 0F00024A XPSD,0 INRUT
 158 01 0005D 0F00024E XPSD,0 PCPINT
 159 01 0005E 0F000236 XPSD,0 INTUN
 160 01 0005F 0F000236 XPSD,0 INTUN

161 PAGE
 162
 163 *
 164 *
 * REGISTER 0 TEST - ENTRY FROM PROGRAM LOAD OR PCP INTERRUPT
 *
 165 01 00060 2F0002C1 START LRP ZFR0 PAGE 0
 166 01 00061 3200029D REUTST LW,0 MINS1 RO==1
 167 01 00062 4800029D EBR,0 MINS1 IS RO=-1, TEST FOR REU AND RO
 168 01 00063 69300066 RCS,3 *+3 NR, GR WAIT
 169 01 00064 32000000 A LW,0 0 IS RO=0
 170 01 00065 68300068 BCR,3 REUBK YES, NO REU INTERFERENCE
 171 01 00066 3200029D LW,0 MINS1 RO==1
 172 01 00067 6C000010 A RD,0 X'10' SS1 SET
 173 01 00068 69800061 BCS,2 REUTST YES, TEST REU AND RO AGAIN
 174 01 00069 2E000000 A WAIT NO, EITHER REU OR RO FAULTY
 175
 176 *
 177 *
 * FAULTY RO OR REU - SEE PAGE 5 OF THIS LISTING UNDER ADDITIONAL ERROR
 *
 178 01 0006A 68000061 REUBK B REUTST LOOP UNTIL BOTH REU AND RO ARE BK
 179 01 0006B 22000000 A REUBK LI,0 0 TEST RO FOR OPEN CIRCUIT
 180 01 0006C 48000000 A EBR,0 0 IS RO=0
 181 01 0006D 68300072 BCR,3 ROGR0D YES, PROCEED WITH THE TEST
 182 01 0006E 6C000010 A RD,0 X'10' SS1 SET
 183 01 0006F 69800068 BCS,2 REUBK YES, TEST RO FOR OPEN CIRCUITS AGAIN
 184 01 00070 2E000000 A WAIT NO FAULTY AT BITS INDICATED BY 1
 185
 186 *
 187 *
 * FAULTY RO - SEE PAGE 5 OF THIS LISTING UNDER 'ADDITIONAL ERROR WAITS'
 *
 188 01 00071 68000068 B REUBK

			PAGE			
189			* INITIALIZATION PROCEDURE			
190			*			
191			*			
192	01 00072	321002A0	ROG00D	LW,1	ARMINT	ARM AND ENABLE PCP, PARITY INTERRUPT
193	01 00073	6D101200 A		WD,1	X'1200'	
194	01 00074	320002A5		LW,0	HALT	
195	01 00075	35000011 A		STW,0	X'111'	
196	01 00076	320002A9		LW,0	BRST	
197	01 00077	35000026 A		STW,0	X'26'	
198	01 00078	320002C1		LW,0	ZERO	CLEAR PASS COUNTER
199	01 00079	350002BA		STW,0	PASCTR	
200	01 0007A	350002BB		STW,0	PASTAG	CLEAR INITIAL PASSING TAG
201	01 0007B	320002B4		LW,0	ONE	CLEAR ERROR COUNTER TO 1
202	01 0007C	350002A4		STW,0	ERRCTR	
203			*			
204			* PROCEDURE TO CALCULATE THE REGISTER PAGE LIMITS			
205			*			
206	01 0007D	320002BC	RESET	LW,0	REGLIM	BEGIN TO CALCULATE NO. OF PAGES
207	01 0007E	350002B6		STW,0	PAGE	
208	01 0007F	350002B7		STW,0	PAGEP	
209	01 00080	650000A1	PSCAN	BIR,0	*+1	INCREMENT PAGE COUNT *C
210	01 00081	480002RC		EBR,0	REGLIM	
211	01 00082	350002B6		STW,0	PAGE	PAGE=00000PPF
212	01 00083	480002A9		EBR,0	MAXPAG	ARE ALL 32 PAGES IMPLEMENTED
213	01 00084	6830008E		BCR,3	ENDPAG	YES
214	01 00085	320002C1		LW,0	ZERO	
215	01 00086	2F0002FA		LRP	PAGE	
216	01 00087	32000292		LW,0	FIVES	FIVES =X'555555'
217	01 00088	32100000 A		LW,1	0	IS THIS PAGE IMPLEMENTED
218	01 00089	6820008E		BCR,2	ENDPAG	NO
219	01 0008A	2F0002C1		LRP	ZERO	YES, INVESTIGATE NEXT PAGE, BY PAGE 0
220	01 0008B	320002FA		LW,0	PAGE	
221	01 0008C	350002B7		STW,0	PAGEP	
222	01 0008D	68000080		B	PSCAN	SCAN NEXT PAGE
223	01 0008E	2F0002C1	ENDPAG	LRP	ZERO	USE PAGE 0
224	01 0008F	320002B7		LW,0	PAGEP	
225	01 00090	480002FC		EBR,0	REGLIM	MASK OUT 4 LEAST SIG. DIGITS

```

PAGE
*
* SELECT THE PAGE TO BE TESTED
*
231 01 00092 320002C2 TESTPG LW,0 7FSZRO R0=X'FFFFFFF0'
232 01 00093 350002B6 STW,0 PAGE INITIALIZE PAGE COUNTER
233 01 00094 320002B6 NEWPAG LW,0 PAGE GET CURRENT PAGE
234 01 00095 480002B9 EBR,0 PAGLIM IS THIS LAST PAGE
235 01 00096 683000F5 BCR,3 TESTPP YES, GO TO DB PAGE ADDRESSING TEST
236 01 00097 320002B6 LW,0 PAGE GET CURRENT PAGE AGAIN
237 01 00098 480002B0 EBR,0 REGLIM EBR WITH F
238 01 00099 6500009A BIR,0 **1 INCREMENT PAGE COUNTER
239 01 0009A 350002B6 STW,0 PAGE STORE NEW PAGE COUNTER
240
241
242
* SELECT TEST PATTERNS IN CONSECUTIVE ORDER
*
243 01 0009B 2F0002C1 GETPAT LRP ZERO RETURN TO PAGE ZERO
244 01 0009C 3200028C LW,0 PATSCT PATSCT=LW,0 PATTRN
245 01 0009D 350000A5 STW,0 PATNBW SET PATTERN SELECTOR CODE
246 01 0009E 3200029D LW,0 MINS1 ALL 1'S
247 01 0009F 3500028E STW,0 PATCNT SET PATTERN COUNTER **1
248 01 000A0 3200028E NXTPAT LW,0 PATCNT GET PATTERN COUNTER
249 01 000A1 650000A2 BIR,0 **1 INCREMENT PATTERN COUNTER
250 01 000A2 3500028E STW,0 PATCNT LAST PATTERN
251 01 000A3 4800028B EBR,0 PATLIM
252 01 000A4 68300094 BCR,3 NEWPAG
253
254
255
* EXECUTE ' LW,0 PATTRN + PATCNT ' TO POINT TO PROPER PATTERN
*
256 01 000A5 00000000 A PATNBW DATA 0 SELECT TEST PATTERN
257 01 000A6 3500028F STW,0 PATSTB PUT CURRENT TEST PATTERN IN MEMORY
258 01 000A7 4800029D EBR,0 MINS1 FIND 1'S COMPLEMENT OF TEST PATTERN
259 01 000A8 35000290 STW,0 CMPATR STORE IT IN MEMORY
260 01 000A9 320000A5 LW,0 PATNBW
261 01 000AA 650000A8 BIR,0 **1
262 01 000AB 350000A5 STW,0 PATNBW

```

```

PAGE
*
* LOAD CURRENT REGISTER WITH PATTERN
*
267 01 000AC 320002C1 DUMPAT LW,0 ZERO
268 01 000AD 35000297 STW,0 R R=0
269 01 000AE 35000298 STW,0 RR R-COUNTER = 0
270 01 000AF 32000296 LW,0 LBDPAT LBDPAT = LW,0 PATS TO
271 01 000B0 350000B4 STW,0 SELECT GET REGISTER SELECTOR CODE
272 01 000B1 3200029E LW,0 STRPAT STRPAT = STW,0 TEMP
273 01 000B2 350000CD STW,0 STORE SET REGISTER STORAGE CODE
274 01 000B3 2F0002B6 NEXTRG LRP PAGE SET POINTER TO CURRENT PAGE
275 01 000B4 00000000 A SELECT DATA 0 LOAD PATTERN TO CURRENT REGISTER R
276
277
278
* LOAD ALL REGISTERS ( EXCEPT CURRENT REG.) OF THE CURRENT PAGE WITH
* COMPLEMENT OF THE CURRENT PATTERN
*
280 01 000B5 2F0002C1 DUMPAT LRP ZERO RETURN TO PAGE 0
281 01 000B6 32000298 LW,0 RR GET R-COUNTER
282 01 000B7 48000297 EBR,0 R IS R-COUNTER = TO CURRENT REG. R
283 01 000B8 683000BF BCR,3 PRESRG YES, SKIP LOADING IT WITH CMPATR
284 01 000B9 32000294 LW,0 LOADRG LOADRG = LW,0 CMPATR
285 01 000BA 48000298 EBR,0 RR MODIFY R FIELD
286 01 000BB 350000D0 STW,0 REGM8D SET REGISTER SELECTOR CODE
287 01 000BC 2F0002B6 LRP PAGE ADVANCE TO CURRENT PAGE
288 01 000BD 00000000 A REGM8D DATA 0 LOAD COMPL. OF PATTERN TO REG. R
289 01 000BE 2F0002C1 LRP ZERO RETURN TO PAGE 0
290 01 000BF 32000298 PRESRG LW,0 RR GET R-COUNTER
291 01 000C0 48000291 EBR,0 FIVEFS MODIFY LOWER BYTES FOR INCREMENTING
292 01 000C1 650000C2 RIR,0 **1 INCREMENT R-COUNTER
293 01 000C2 35000298 STW,0 RR STORE IT BACK
294 01 000C3 4800029A EBR,0 RMAX WAS THIS LAST REGISTER
295 01 000C4 683000C6 BCR,3 TSTREG YES, GO TEST THE CURRENT TEST-REG.
296 01 000C5 620000B6 B REGLOD REPEAT TO LOAD NEXT REGISTER

```

297
298
299
300
301 01 000C6 22000000 A TSTREG LI,0 0
302 01 000C7 35000298 STW,0 RR INITIALIZE R-COUNTER
303 01 000C8 32000286 LW,0 PAGE IS THIS PAGE 0
304 01 000C9 693000CC BCS,3 EXAMIN YES, MAKE TEST
305 01 000CA 32000297 LW,0 R IS THIS R0 OF PAGE 0
306 01 000CB 683000D2 BCR,3 REG00D YES, DO NOT TEST R0
307 01 000CC 2F000286 EXAMIN LRP PAGE RETURN TO TEST PAGE
308 01 000CD 00000000 A STORE DATA 0 STORE REG. R IN LOCATION TEMP
309 01 000CE 2F0002C1 LRP ZERO RETURN TO PAGE 0
310 01 000CF 3200029F LW,0 TEMP FETCH STORED DATA
311 01 000D0 4800028F EBR,0 PATST0 COMPARE WITH CURRENT PATTERN
312 01 000D1 693000DF BCS,3 BYTERR UNEQUAL GO TO FIND FAULTY BYTE
313 01 000D2 32000297 REG00D LW,0 R GET R
314 01 000D3 48000291 EBR,0 FIVEFS MODIFY TO INCREMENT BY OVERFLOW
315 01 000D4 650000D5 BIR,0 *+1 INCREMENT R
316 01 000D5 35000297 STW,0 R STORE NEW VALUE OF R
317 01 000D6 4800029A EBR,0 RMAX LAST REGISTER
318 01 000D7 683000A0 BCR,3 NXTPAT YES, REPEAT CURRENT PAGE TEST WITH
319 NEXT PATTERN
320 01 000D8 32000297 LW,0 R
321 01 000D9 48000296 EBR,0 L0DPAT L0DPAT = LW,0 PATST0
322 01 000DA 35000084 STW,0 SELECT MODIFY LOAD LOCATION FOR NEXT REG.
323 01 000DB 32000297 LW,0 R
324 01 000DC 4800029E EBR,0 SIRPAT SIRPAT = STW,0 TEMP
325 01 000DD 350000CD STW,0 STORE MODIFY STORE LOCATION FOR NEXT REG.
326 01 000DE 68000083 B NEXTTRG GO TEST NEXT REGISTER

327
328
329
330
331 01 000DF 320002C1 BYTERR LW,0 ZFR0 INITIALIZE THE BYTE COUNTER
332 01 000E0 35000288 STW,0 BYTCNT BYTMD = LW,0 BYT0
333 01 000E1 32000286 LW,0 BYTMD SET BYE MODIFYING CODE
334 01 000E2 350000F3 STW,0 BYTN0W GET PROPER BYT0'S MASK
335 01 000E3 00000000 A BYTN0W DATA 0 STORE IN MASK
336 01 000E4 3500029C STW,0 MASK GET STORED BYTE
337 01 000E5 3200029F LW,0 TEMP TEST THE BYTE WITH ORIGINAL BYTE
338 01 000E6 4800028F EBR,0 PATST0 DISCARD OTHER INTERFERING BYTES
339 01 000E7 4800029C AND,0 MASK ERROR, GO TO ERROR ROUTINE
340 01 000E8 69300188 BCS,3 REGERR NO ERROR
341 01 000E9 32000288 RFTBYT LW,0 BYTCNT INCREMENT BYTE COUNTER
342 01 000EA 6500008B BIR,0 *+1
343 01 000EB 35000288 STW,0 BYTCNT
344 01 000EC 48000284 EBR,0 BYTMAX LAST BYT0
345 01 000ED 683000F2 BCR,3 TSTSS1 YES, TEST SS1 FOR REPETITION
346 01 000EE 320000F3 LW,0 BYTN0W NO
347 01 000EF 650000F0 BIR,0 *+1 MODIFY LOCATION TO GET NEXT BYTE
348 01 000F0 350000E3 STW,0 BYTN0W
349 01 000F1 680000E3 B GO TEST NEXT BYT0
350
351
352
353 01 000F2 6C000010 A TSTSS1 RD,0 X'10' IS SS1 SET
354 01 000F3 698000DF BCS,8 BYTERR YES, REPEAT TESTING SAME REGISTER
355 01 000F4 680000D2 B REG00D NO, ADVANCE TO TEST NEXT REGISTER

```

PAGE
* TEST FOR CORRECT ADDRESSING OF THESE IMPLEMENTED PAGES
*
*
360 01 000F5 2F0002C1 TESTPP LRP ZERO SET RP TO PAGE ZERO
361 01 000F6 320002B9 LW,0 PAGLIM IS ONLY PAGE 0 IMPLEMENTED *C
362 01 000F7 68300156 BCR,3 CORRIZ YES, GO TO CORE TEST *C
363 01 000F8 320002C1 LW,0 ZERO
364 01 000F9 350002B6 STW,0 PAGE INITIALIZE PAGE COUNTER
365 01 000FA 320002C1 FILLIS LW,0 ZERO
366 01 000FB 35000297 STW,0 R R=0
367 01 000FC 32000295 LW,0 LDMNS1 SET CODE TO LOAD REGISTERS *-1
368 01 000FD 350000FF STW,0 MODREG
369 01 000FE 2F0002B6 LADREG LRP PAGE SET CURRENT PAGE
*
* EXECUTE 'LW,R *-1' IN ORDER TO LOAD ALL REGS. OF CURRENT PAGE *-1
*
373 01 000FF 00000000 A MODREG DATA 0
374 01 00100 2F0002C1 LRP ZERO BACK TO PAGE 0
375 01 00101 32000297 LW,0 R FETCH R
376 01 00102 48000291 EBR,0 FIVEFS MODIFY LOWER BITS FOR INCREMENTATION
377 01 00103 65000104 BIR,0 *-1 INCREMENT R BY OVERFLOW
378 01 00104 35000297 STW,0 R STORE NEW R
379 01 00105 4800029A EBR,0 RMAX LAST REGISTER IN CURRENT PAGE
380 01 00106 6830010B BCR,3 FLPAGE YES, FLIP THE PAGE
381 01 00107 32000297 LW,0 R
382 01 00108 48000295 EBR,0 LDMNS1 LDMNS1= LW,0 MINS1
383 01 00109 350000FF STW,0 MODREG MODIFY LOCATION TO SET NEXT REG. *-1
384 01 0010A 680000FF R LADREG LOAD NEXT R *-1
    
```

```

PAGE
* FLIP TO LOAD REGISTERS OF THE PAGE *-1
*
*
389 01 0010H 2F0002C1 FLPAGE LRP ZERO RETURN TO PAGE 0
390 01 0010C 320002B6 LW,0 PAGE GET PAGE COUNTER
391 01 0010D 480002B9 EBR,0 PAGLIM LAST PAGE
392 01 0010E 68300114 BCR,3 FULPGS YES
393 01 0010F 320002B6 LW,0 PAGE GET PAGE COUNTER
394 01 00110 480002BC EBR,0 REGLIM MODIFY TO INCREMENT BY OVERFLOW
395 01 00111 65000112 BIR,0 *-1 INCREMENT PAGE COUNTER
396 01 00112 350002B6 STW,0 PAGE STORE NEW VALUE OF THE PAGE COUNTER
397 01 00113 680000FA B FILLIS NO, LOAD NEXT PAGE ALL REGS. *-1
398 01 00114 320002C1 FULPGS LW,0 ZERO
399 01 00115 350002B6 STW,0 PAGE INITIALIZE PAGE COUNTER
400 01 00116 35000293 STW,0 FLPAGE INITIALIZE PAGE FLIP
401 01 00117 2F0002B6 BEGPAGE LRP PAGE SET RP TO CURRENT PAGE
402 01 00118 320002C1 LW,0 ZERO CLEAR REGISTER 0
403 01 00119 32A002C1 LW,8 ZERO CLEAR REGISTER 8
    
```



```

PAGE
*
* CALCULATE MAXIMUM CORE ADDRESS
*
CORISZ  LRP      ZERO      RESET PAGE POINTER
         LW,0     PASTAG    IS THIS THE FIRST PASS
         BCR,3    *+3      YES, CALCULATE CORE SIZE
         RD,0     X'10'    IS SS2 SET
         BCR,4    CORFIL   NO, USE CURRENT CORE SIZE
         LW,1     N300     YES, RECALCULATE CORE SIZE
         LW,0     MEMFIL
SETFIL  BIR,0    *+1      INCREMENT STW ADDRESS
         STW,0    *+1
         DATA   0        EXECUTE STW,1 (R0)+1
         BIR,1    SETFIL   INCREMENT FILL DATA
         LW,1     MAXCOR   128K CORE IMPLEMENTED
MAXMEM  AND,1    MAXCOR   TRAP RETURN = MASK BUT NON-ADDRESS
         STW,1    MEMAX    SAVE MAX ADDRESS
         LW,1     P300
         STW,1    MEMIN    SAVE MIN TEST ADDRESS
         LW,0     FIVES
         STW,0    PASTAG   STORE NON-ZERO NO. IN PASTAG
*
* SET EACH CORE LOCATION EQUAL TO ITS OWN ADDRESS
*
CORFIL  LW,8     MEMST0
         EBR,8    MEMIN   ADD MIN TEST ADDRESS
         LW,6     MEMIN   INITIAL ADDRESS AND TEST VALUE
CORLBD  STW,8    *+1
         DATA   0        EXECUTE STW,6 TEST ADDRESS
         LW,9     MEMAX
         EBR,9    6        TEST VALUE = MAX ADDRESS
         BCR,3    MACTST  YES
         BIR,6    *+1     NO, INCREMENT TEST VALUE
         BIR,8    *+1     INCREMENT TEST ADDRESS
         BCR,0    CORLBD

```

```

PAGE
*
* TEST CORE MEMORY BETWEEN DEFINED LIMITS FOR CONTENT EQUAL TO ADDRESS
*
MACTST  LW,A     STORCR
         EBR,A    MEMIN   ADD MIN TEST ADDRESS
         LW,B     BTGERR  ERROR BRANCH FOR MEM EXECUTE TEST
         LW,F     MEMLSD
         EBR,F    MEMIN   ADD TEST ADDRESS
         LW,6     MEMIN   INITIAL TEST VALUE
MACLUP  STW,8    *+1
         DATA   0        EXECUTE LW,5 TEST ADDRESS
         LW,7     6        PREDETERMINED VALUE
         EBR,7    5        PREDETERMINED = ACTUAL
         BCR,3    *+2     YES
         BCR,0    MACERR  NO, MEMORY ACCESS ERROR
         RD,0     X'10'   SS1 SET
         BCS,A    MACLUP+1 YES, REPEAT TEST FOR THIS ADDRESS
         STW,A    *+1     NO,
         DATA   0        STORE (BCR,0 MACERR) FOR EXECUTE TEST
         LW,9     MEMAX
         EBR,9    6        TEST VALUE = MAX ADDRESS
         BCR,3    MEXTST  YES, PROCEED TO NEXT TEST
         BIR,6    *+1     INCREMENT TEST VALUE
         BIR,8    *+1     INCREMENT TEST ADDRESS
         BIR,A    *+1     INCREMENT REFILL ADDRESS
         BCR,0    MACLUP

```

PAGE

* TEST INSTRUCTION EXECUTION FROM EACH LOCATION WITHIN MEMORY LIMITS

535								
536								
537								
538								
539	01	0018A	329002A3	MEXTST	LW,9	BTERR		LOAD BRANCH TO ERROR
540	01	0018B	328002C0		LW,8	STORE9		LOAD STORE R9
541	01	0018C	488002AD		EOR,8	MEMIN		ADD MIN TEST ADDRESS
542	01	0018D	32A002A1		LW,4	BIRL8D		LOAD TEST RIR
543	01	0018E	32B002BF		LW,8	ST08IR		LOAD STORE RA
544	01	0018F	488002AD		EOR,8	MEMIN		ADD MIN TEST ADDRESS
545	01	00190	32D002A2		LW,D	BT08IR		LOAD BRANCH TO TEST ADDRESS
546	01	00191	48D002AD		EOR,D	MEMIN		ADD MIN TEST ADDRESS
547	01	00192	35B00193	MEXLUP	STW,8	**1		
548	01	00193	00000000		DATA	0		EXECUTE STW,4 TEST ADDRESS
549	01	00194	32C002AA	TESTEX	LW,C	MAXP9S		RC = LARGEST POSITIVE VALUE
550	01	00195	35D00196		STW,D	**1		
551	01	00196	00000000		DATA	0		EXECUTE BCR,0 TEST ADDRESS
552	01	00197	6C000010	EXRET	RD,0	X'10'		SUCCESSFUL EXECUTION RETURN
553	01	00198	69800194		BCR,8	TESTEX		SS1 SET - REPEAT TEST FOR THIS ADDR
554	01	00199	3580019A		STW,8	**1		
555	01	0019A	00000000		DATA	0		REPLACE BIR WITH BCR TO ERROR
556	01	0019B	32E00008		LW,E	8		CURRENT TEST LOCATION
557	01	0019C	48E002A8		AND,E	MAXC0R		ADDRESS MASK
558	01	0019D	48E002A8		EOR,E	MEMAX		CURRENT ADDRESS = MEMORY LIMIT
559	01	0019E	683001A3		BCR,3	PASINC		YES, EXECUTION TEST COMPLETED
560	01	0019F	658001A0		BIR,8	**1		NO, TEST NEXT LOCATION
561	01	001A0	658001A1		BIR,8	**1		
562	01	001A1	65D001A2		BIR,D	**1		
563	01	001A2	68000192		RCR,0	MEXLUP		

PAGE

* END OF PROGRAM PASS. INCREMENT PASS COUNTER AND DISPLAY MEMORY
* PARAMETERS FOR ALTERATION IF REQUESTED.

564								
565								
566								
567								
568								
569	01	001A3	3240028A	PASINC	LW,4	PASCTR		
570	01	001A4	654001A5		BIR,4	**1		
571	01	001A5	3540028A		STW,4	PASCTR		
572	01	001A6	32000288		LW,0	PAGREG		
573	01	001A7	6C000010		RD,0	X'10'		SS3 SET
574	01	001A8	68200181		BCR,P	REITER		NR
575	01	001A9	32100289		LW,1	PAGLIM		
576	01	001AA	322002AD		LW,2	MEMIN		
577	01	001AB	323002A8		LW,3	MEMAX		
578	01	001AC	325002A4		LW,5	ERRCTR		
579	01	001AD	2E000000		*AIT			YES, HALT FOR CHANGE OF REGISTER- PAGE LIMITS AND/OR MEMORY LIMITS
580								
581	01	001AE	35100289		STW,1	PAGLIM		
582	01	001AF	352002AD		STW,2	MEMIN		
583	01	001B0	353002A8		STW,3	MEMAX		
584	01	001B1	35000286	REITER	STW,0	PAGE		ENTER DESIRED STARTING PAGE
585	01	001B2	35000288		STW,0	PAGBEG		
586	01	001B3	6C000010		RD,0	X'10'		SS2 SET
587	01	001B4	68400098		BCR,4	GETPAT		REPEAT TEST WITH CURRENT PAGE LIMITS AND/OR CURRENT CHRE PARAMETERS
588								
589	01	001B5	320002C1		LW,0	ZERO		
590	01	001B6	35000288		STW,0	PAGBEG		
591	01	001B7	6800007D		B	RESET		RETURN TO RECALCULATE PARAMETERS

```

592                                     PAGE
593
594 * ERROR ROUTINES FOR REGISTER PAGE TESTS
595 *
596 01 001B8 320002C1 REGERR LW,0 ZERO CLEAR REGISTER ZERO
597 01 001B9 480002B6 EBR,0 PAGE RECORD PAGE
598 01 001BA 48000297 EBR,0 R RECORD R
599 01 001BB 48000288 EBR,0 BYCNT RECORD BYTE
600 01 001BC 4800029R EBR,0 REGIND REGIND=X'10000000'
601 01 001BD 321002B7 LW,1 BYTRET RETURN ENTRY
602 01 001BE 351001EA STW,1 RETURN
603 01 001BF 321002B9 LW,1 PAGLIM LOAD REG.1 WITH MAX. PAGE IMPLEMENTF
604 01 001C0 322002BF LW,2 PATST0 LOAD REG.2 WITH CURRENT PATTERN
605 01 001C1 3230029F LW,3 TEMP LOAD REG. 3 WITH PATTERN RECEIVED
606 01 001C2 32400002 A LW,4 2
607 01 001C3 48400003 A EBR,4 3 LOAD REG.4 WITH DIFFERING BITS
608 01 001C4 680001E2 B ERROR
609 01 001C5 320002C1 PAGERR LW,0 ZERO CLEAR R0
610 01 001C6 48000293 EBR,0 ELPCNT RECORD HIGHER PAGE NUMBER BITS 23-27
611 01 001C7 48000297 EBR,0 R RECORD R IN BITS R=11
612 01 001C8 48000288 EBR,0 BYCNT RECORD BYTE NO. BITS 29-31
613 01 001C9 4800028D EBR,0 PGAIND PGAIND=X'20000000'
614 01 001CA 325002B6 LW,5 PAGE RECORD DESIRED PAGE IN R5-BITS23-27
615 01 001CB 321002BA LW,1 PAGRET RETURN ENTRY
616 01 001CC 351001EA STW,1 RETURN
617 01 001CD 321002B9 LW,1 PAGLIM RECORD PAGES IMPLEMENTED IN R1
618 01 001CE 680001E2 B ERROR

```

```

619                                     PAGE
620
621 * ERROR ROUTINES FOR MEMORY TESTS
622 *
623 01 001CF 32000006 A MACERR LW,0 6 FAILING ADDRESS
624 01 001D0 480002AA EBR,0 MACERI ERROR ID
625 01 001D1 32F002A7 LW,F MACRET RETURN ENTRY
626 01 001D2 35F001FA STW,F RETURN
627 01 001D3 321002B9 FRINFP LW,1 PAGLIM MAXIMUM PAGES IMPLEMENTED
628 01 001D4 322002B5 LW,2 MEMAX MAXIMUM CORE ADDRESS
629 01 001D5 323002A4 LW,3 ERRCTR ERROR COUNT
630 01 001D6 324002BA LW,4 PASCTR PASS COUNT
631 01 001D7 680001E2 BCR,0 ERROR
632 01 001D8 3200000A A MEXERR LW,0 R FAILING ADDRESS
633 01 001D9 480002AR AND,0 MAXCHR ADDRESS MASK
634 01 001DA 480002B1 EBR,0 MEXERI ERROR ID
635 01 001DB 32F002B0 LW,F MEXRET RETURN ENTRY
636 01 001DC 35F001FA STW,F RETURN
637 01 001DD 3250000C A LW,5 C RESULT IN RC AFTER BIR,C
638 01 001DE 326002B2 LW,6 MINUS1 CORRECT RESULT IN RC
639 01 001DF 32700005 A LW,7 5
640 01 001E0 48700006 A EBR,7 6 DIFFERENCE
641 01 001E1 680001D3 BCR,0 ERINFB
642
643
644 01 001E2 60000041 A ERRPR WD,0 X'41' TURN ON ALARM
645 01 001E3 60000010 A RD,0 X'10' OS4 SET
646 01 001E4 691001E6 BCS,1 *+2 YES, NO HALT ON ERROR
647 01 001E5 2F000000 A FRHALT WAIT
648 * REFER TO PAGE 4 OF LISTING FOR AMPLIFYING ERROR INFORMATION
649 01 001E6 60000040 A WD,0 X'40' TURN OFF ALARM
650 01 001E7 320002A4 LW,0 ERRCTR INCREMENT ERROR COUNT
651 01 001E8 650001E3 BIR,0 *+1
652 01 001E9 350002A4 STW,0 ERRCTR
653 01 001EA 00000000 A RETURN DATA 0 RETURN BRANCH

```

PAGE

```

654
655
656
657
658
659 01 001EC 00000000 A  NONBP  BBUND  8
660 01 001ED 00000000 A  DATA  0
661 01 001EE 000001F0  DATA  NONBP+4
662 01 001EF 00000000 A  DATA  0
663 01 001F0 68000252  BCR,0  TRAP#0  INTERRUPT SYSTEM FAULT
664 01 001F1 68000252  BCR,0  TRAP#0  MEMORY PROTECT VIOLATION
665 01 001F2 68000252  BCR,0  TRAP#0  MODE VIOLATION
666 01 001F3 2E000000 A  WAIT
667 01 001F4 68000162  BCR,0  MAXMEM  NONEXISTENT MEMORY
668 01 001F5 2E000000 A  WAIT
669 01 001F6 2E000000 A  WAIT
670 01 001F7 2E000000 A  WAIT
671 01 001F8 68000252  BCR,0  TRAP#0  NONEXISTENT INSTRUCTION
672
673 01 001FA 00000000 A  UNIMP  BBUND  8
674 01 001FB 00000000 A  DATA  0
675 01 001FC 00000254  DATA  TRAP#1
676 01 001FD 00000000 A  DATA  0
677 01 001FE 00000000 A  STACK  DATA  0
678 01 001FF 00000000 A  DATA  0
679 01 00200 00000256  DATA  TRAP#2
680 01 00201 00000000 A  DATA  0
681 01 00202 00000000 A  GFLB  DATA  0
682 01 00203 00000000 A  DATA  0
683 01 00204 00000258  DATA  TRAP#3
684 01 00205 00000000 A  DATA  0
685 01 00206 00000000 A  FLBAT  DATA  0
686 01 00207 00000000 A  DATA  0
687 01 00208 0000025A  DATA  TRAP#4
688 01 00209 00000000 A  DATA  0

```

PAGE

```

689
690
691 01 0020A 00000000 A  DEC  DATA  0
692 01 0020B 00000000 A  DATA  0
693 01 0020C 0000025C  DATA  TRAP#5
694 01 0020D 00000000 A  DATA  0
695 01 0020E 00000000 A  TIMER  DATA  0
696 01 0020F 00000000 A  DATA  0
697 01 00210 0000025E  DATA  TRAP#6
698 01 00211 00000000 A  DATA  0
699 01 00212 00000000 A  CALL1  DATA  0
700 01 00213 00000000 A  DATA  0
701 01 00214 00000260  DATA  TRAP#8
702 01 00215 00000000 A  DATA  0
703 01 00216 00000000 A  CALL2  DATA  0
704 01 00217 00000000 A  DATA  0
705 01 00218 00000262  DATA  TRAP#9
706 01 00219 00000000 A  DATA  0
707 01 0021A 00000000 A  CALL3  DATA  0
708 01 0021B 00000000 A  DATA  0
709 01 0021C 00000264  DATA  TRAP#A
710 01 0021D 00000000 A  DATA  0
711 01 0021E 00000000 A  CALL4  DATA  0
712 01 0021F 00000000 A  DATA  0
713 01 00220 00000266  DATA  TRAP#B
714 01 00221 00000000 A  DATA  0
715 01 00222 00000000 A  TRAPUN  DATA  0
716 01 00223 00000000 A  DATA  0
717 01 00224 00000268  DATA  TUNASS
718 01 00225 00000000 A  DATA  0

```

				PAGE	
719					
720					
721					* INTERRUPT PROCESSING SWITCH
722					
723				BBUND	8
724	01 00226	00000000	A	PBWN	DATA 0
725	01 00227	00000000	A		DATA 0
726	01 00228	0000026A			DATA INT50
727	01 00229	00000000	A		DATA 0
728	01 0022A	00000000	A	PBWBFF	DATA 0
729	01 0022B	00000000	A		DATA 0
730	01 0022C	0000026C			DATA INT51
731	01 0022D	00000000	A		DATA 0
732	01 0022E	00000000	A	PULSE1	DATA 0
733	01 0022F	00000000	A	PULSE2	DATA 0
734	01 00230	00000000	A	PULSE3	DATA 0
735	01 00231	00000000	A	PULSE4	DATA 0
736	01 00232	00000000	A	MEMPAR	DATA 0
737	01 00233	00000000	A		DATA 0
738	01 00234	0000026E			DATA INT56
739	01 00235	00000000	A		DATA 0
740	01 00236	00000000	A	INTUN	DATA 0
741	01 00237	00000000	A		DATA 0
742	01 00238	00000270			DATA INT58
743	01 00239	00000000	A		DATA 0
744	01 0023A	00000000	A	CBUNT1	DATA 0
745	01 0023B	00000000	A		DATA 0
746	01 0023C	00000272			DATA INT58
747	01 0023D	00000000	A		DATA 0
748	01 0023E	00000000	A	CBUNT2	DATA 0
749	01 0023F	00000000	A		DATA 0
750	01 00240	00000274			DATA INT59
751	01 00241	00000000	A		DATA 0

				PAGE	
752					
753					
754	01 00242	00000000	A	CBUNT3	DATA 0
755	01 00243	00000000	A		DATA 0
756	01 00244	00000276			DATA INT5A
757	01 00245	00000000	A		DATA 0
758	01 00246	00000000	A	CBUNT4	DATA 0
759	01 00247	00000000	A		DATA 0
760	01 00248	00000278			DATA INT5B
761	01 00249	00000000	A		DATA 0
762	01 0024A	00000000	A	INBUT	DATA 0
763	01 0024B	00000000	A		DATA 0
764	01 0024C	0000027A			DATA INT5C
765	01 0024D	00000000	A		DATA 0
766	01 0024E	00000000	A	PCPINT	DATA 0
767	01 0024F	00000000	A		DATA 0
768	01 00250	00000060		RESTRY	DATA START
769	01 00251	00000000	A		DATA 0

770 PAGE
 771 * TRAP HALTS FOR IDENTIFICATION OF SPURIOUS TRAPS
 772 *
 773 *
 774 BOUND 8
 775 01 00252 2E000000 A TRAP40 WAIT BCR,0 START NONALLOWED OPERATION
 776 01 00253 68000060 TRAP40 BCR,0 START UNIMPLEMENTED INSTRUCTION
 777 01 00254 2F000000 A TRAP41 WAIT BCR,0 START PUSH-DOWN STACK LIMIT REACHED
 778 01 00255 68000060 TRAP42 WAIT BCR,0 START FIXED-POINT ARITHMETIC OVERFLOW
 779 01 00256 2E000000 A TRAP42 WAIT BCR,0 START FLOATING-POINT FAULT
 780 01 00257 68000060 TRAP43 WAIT BCR,0 START DECIMAL ARITHMETIC FAULT
 781 01 00258 2E000000 A TRAP43 WAIT BCR,0 START WATCHDOG TIMER RUN-OUT
 782 01 00259 68000060 TRAP44 WAIT BCR,0 START CALL1
 783 01 0025A 2E000000 A TRAP44 WAIT BCR,0 START CALL2
 784 01 0025B 68000060 TRAP45 WAIT BCR,0 START CALL3
 785 01 0025C 2E000000 A TRAP45 WAIT BCR,0 START CALL4
 786 01 0025D 68000060 TRAP46 WAIT BCR,0 START UNASSIGNED TRAP 45,47,4C-4F
 787 01 0025E 2E000000 A TRAP46 WAIT BCR,0 START
 788 01 0025F 68000060 TRAP4R WAIT BCR,0 START
 789 01 00260 2F000000 A TRAP4R WAIT BCR,0 START
 790 01 00261 68000060 TRAP4R WAIT BCR,0 START
 791 01 00262 2E000000 A TRAP4R WAIT BCR,0 START
 792 01 00263 68000060 TRAP4A WAIT BCR,0 START
 793 01 00264 2E000000 A TRAP4A WAIT BCR,0 START
 794 01 00265 68000060 TRAP4R WAIT BCR,0 START
 795 01 00266 2E000000 A TRAP4R WAIT BCR,0 START
 796 01 00267 6F000060 TUNASS WAIT BCR,0 START
 797 01 00268 2E000000 A TUNASS WAIT BCR,0 START
 798 01 00269 68000060 TUNASS WAIT BCR,0 START

799 PAGE
 800 * INTERRUPT HALTS FOR IDENTIFICATION OF SPURIOUS INTERRUPTS
 801 *
 802 *
 803 BOUND 8
 804 01 0026A 2F000000 A INT50 WAIT LPSD,0 RESTRT POWER ON
 805 01 0026B 0E000250 INT50 LPSD,0 RESTRT POWER OFF
 806 01 0026C 2F000000 A INT51 WAIT LPSD,0 RESTRT MEMORY PARITY
 807 01 0026D 0F000250 INT51 LPSD,0 RESTRT UNASSIGNED INTERRUPT 57,5E,5F
 808 01 0026E 2E000000 A INT56 WAIT LPSD,0 RESTRT COUNTER 1 ZERO
 809 01 0026F 0E000250 INT56 LPSD,0 RESTRT COUNTER 2 ZERO
 810 01 00270 2E000000 A TUNASS WAIT LPSD,0 RESTRT COUNTER 3 ZERO
 811 01 00271 0E000250 INT5R WAIT LPSD,0 RESTRT COUNTER 4 ZERO
 812 01 00272 2E000000 A INT5R LPSD,0 RESTRT INPUT/OUTPUT
 813 01 00273 0F000250 INT59 WAIT LPSD,0 RESTRT
 814 01 00274 2E000000 A INT5A WAIT LPSD,0 RESTRT
 815 01 00275 0F000250 INT5A LPSD,0 RESTRT
 816 01 00276 2E000000 A INT5A WAIT LPSD,0 RESTRT
 817 01 00277 0E000250 INT5B WAIT LPSD,0 RESTRT
 818 01 00278 2E000000 A INT5B LPSD,0 RESTRT
 819 01 00279 0F000250 INT5C WAIT LPSD,0 RESTRT
 820 01 0027A 2E000000 A INT5C LPSD,0 RESTRT
 821 01 0027B 0E000250 INT5C LPSD,0 RESTRT

822					PAGE	
823	01	0027C	3C3C3C3C	A	PATRN	DATA X'3C3C3C3C'
824	01	0027D	C3C3C3C3	A		DATA X'C3C3C3C3'
825	01	0027E	5A5A5A5A	A		DATA X'5A5A5A5A'
826	01	0027F	A5A5A5A5	A		DATA X'A5A5A5A5'
827	01	00280	FF000000	A	BYTE0	DATA X'FF000000'
828	01	00281	00FF0000	A	BYTE1	DATA X'00FF0000'
829	01	00282	0000FF00	A	BYTE2	DATA X'0000FF00'
830	01	00283	000000FF	A	BYTE3	DATA X'000000FF'
831	01	00284	00000004	A	BYTMAX	DATA X'4'
832	01	00285	4800029C	A	BYTAND	AND,0 MASK
833	01	00286	32000280	A	BYTH0D	LW,0 BYTE0
834	01	00287	480000E9	A	BYTRET	B RETBYT
835	01	00288	00000000	A	BYTCNT	DATA 0
836	01	00289	68000060	A	BRST	B START
837	01	0028A	6800014A	A	PAGRET	B RETPAG
838	01	0028B	00000008	A	PATLIM	DATA X'8'
839	01	0028C	3200027C	A	PATSCT	LW,0 PATRN
840	01	0028D	20000000	A	PGAIN0	DATA X'20000000'
841	01	0028E	00000000	A	PATCNT	DATA 0
842	01	0028F	00000000	A	PATST0	DATA 0
843	01	00290	00000000	A	CMPATR	DATA 0
844	01	00291	000FFFFFF	A	FIVFES	DATA X'FFFFFF'
845	01	00292	00555555	A	FIVES	DATA X'555555'
846	01	00293	00000000	A	FLPCNT	DATA 0
847	01	00294	32000290	A	LRA0RG	LW,0 CMPATR
848	01	00295	32000290	A	LDMNS1	LW,0 MINS1
849	01	00296	3200028F	A	L0DPAT	LW,0 PATST0
850	01	00297	00000000	A	R	DATA 0
851	01	00298	00000000	A	RR	DATA 0
852	01	00299	00800000	A	RR	DATA X'800000'
853	01	0029A	01000000	A	RMAX	DATA X'1000000'
854	01	0029B	10000000	A	REGIND	DATA X'10000000'
855	01	0029C	00000000	A	MASK	DATA 0
856	01	0029D	FFFFFFFF	A	MINS1	DATA X'FFFFFFFF'
857	01	0029E	3500029F	A	STRPAT	STW,0 TEMP
858	01	0029F	00000000	A	TEMP	DATA 0

859					PAGE	
860	01	002A0	00000R10	A	ARMINT	DATA X'R10'
861	01	002A1	65C00197	A	BIRL0D	BIR,C EXRET
862	01	002A2	68000000	A	BT0BIR	RCR,0 0
863	01	002A3	680001D8	A	BYBERR	RCR,0 MEXERR
864	01	002A4	00000000	A	FRRCTP	DATA 0
865	01	002A5	2E000000	A	HALT	WAIT
866	01	002A6	30000000	A	MACERI	DATA X'30000000'
867	01	002A7	6800017F	A	MACRFT	RCR,0 MACRET
868	01	002A8	0001FFFF	A	MAXC0R	DATA X'0001FFFF'
869	01	002A9	0000020F	A	MAXPAG	DATA X'20F'
870	01	002AA	7FFFFFFF	A	MAXP8S	DATA X'7FFFFFFF'
871	01	002AB	00000000	A	MEMAX	DATA 0
872	01	002AC	35100300	A	MEMFIL	STW,1 X'300'
873	01	002AD	00000000	A	MEMIN	DATA 0
874	01	002AE	32500000	A	MEMLBD	LW,5 0
875	01	002AF	35600000	A	MEMST0	STW,6 0
876	01	002B0	68000197	A	MEXRET	RCR,0 EXRET
877	01	002B1	40000000	A	MEXERI	DATA X'40000000'
878	01	002B2	80000000	A	MINUS1	DATA X'80000000'
879	01	002B3	FFFE0300	A	N300	DATA X'FFFE0300'
880	01	002B4	00000001	A	0NE	DATA 1
881	01	002B5	00000300	A	P300	DATA X'300'
882	01	002B6	00000000	A	PAGE	DATA 0
883	01	002B7	00000000	A	PAGEP	DATA 0
884	01	002B8	00000000	A	PAGBEG	DATA 0
885	01	002B9	00000000	A	PAGLIM	DATA 0
886	01	002BA	00000000	A	PASCTR	DATA 0
887	01	002BB	00000000	A	PASTAG	DATA 0
888	01	002BC	0000000F	A	REGLIM	DATA F
889	01	002BD	00000000	A	SAVERR	DATA 0
890	01	002BE	35800000	A	ST0RCP	STW,B 0
891	01	002BF	35A00000	A	ST0BIR	STW,A 0
892	01	002C0	35900000	A	ST0RE9	STW,9 0
893	01	002C1	00000000	A	ZFR0	DATA 0
894	01	002C2	FFFFFFFF	A	ZFSZRR	DATA X'FFFFFFFF'
895		01 00060			END	START