SIGMA 5 THROUGH 9
SYSTEMS EXERCISER
PROGRAM NO. 705889 BO3

901737B-3
Price: $4.00

USERS MANUAL

SIGMA 5 THROUGH 9 SYSTEMS EXERCISER

PROGRAM NO. 705889B03

Specifications, equipment descriptions, and procedures contained herein are subject to change without notice.

January 1973

This publication supersedes 901737B-2,
dated October 1972

Prepared by
Field Engineering Publications

XEROX

701 So. Aviation Blvd., El Segundo, Calif. 90245, 213 679-4511

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

1.1 Objectives

The objective of the System Exerciser is to provide a tool which aids in detection and isolation of system failures and which verifies the proper operation of any standard Sigma 5/6/7/8/9 system.

The System Exerciser is sufficiently complex as to require special user training for the operator to achieve a high degree of fault isolating efficiency. However, relatively little training is required to use the program to verify proper operation of a system.

The program is required to start and run with a minumum amount of operator intervention. Program recovery is offered as an option to those systems which have a RAD.

The program includes an interactive user interface which permits local, remote terminal or both to control or observe the systems operation. The program includes a pre-exerciser configuration declaration which combines automatic configuration determination with operator declared configuration producing the test strategy by which the system is exercised. The program includes a supervisor which can exercise and test all of the systems hardware elements while approaching the maximum system activity. The program also maintains an error history of all the errors encountered during the exerciser's operation.

1.2 Configuration

The program can be executed on any standard Sigma 5/6/7/8/9 hardware configuration.

The minimum hardware configuration (which includes the program recovery option) is:

(1) 16K memory

(2) Console teletype model 7010, 7012, 7012-2, 7020, or 7020-2.

(3) Any card reader model 7120, 7122, 7140, or Magnetic Tape model 7320, 7361, 7374, 7330.

The minimum hardware configuration including the program recovery option requires at least one of the following added configurations:

(4) RAD Model 7201, 7202, 7203, 7204, 7211, 7212, 7231, 7232, 7235 and 7236.

1.2 Configuration (Cont'd.)

Any of the following elements can be tested:

(1)	Any Sigma 5 thru 9 CPU	8201, 8301, 8401, 8501, 8601
(2)	Memory Map	8415, 8615
(3)	Additional Register Blocks	8216, 8316, 8416, 8516, 8616
(4)	Floating Point	8218, 8318, 8418, 8518, 8618
(5)	Decimal	8319, 8419, 8519, 8619
(6)	All Memorys	
(7)	All IOPs & IOP Options (including RIOP)	
(8)	RAD Controllers and Devices	7211/12, 7231/32/35/36, 7201/02/03/04
(9)	Disc Packs	7240/41/42/43, 7260/65
(10)	9 Track Tape	7320/21&22, 7330/31/32/33
(11)	7 Track Tape	7361/62, 7371/74
(12)	Line Printers	7440, 7441, 7445, 7446, 7450
(13)	Card Readers	7120, 7121, 7122, 7140
(14)	Card Punches	7160, 7165

1.3 Loading

The System Exerciser is available as a stand-alone program as well as on MTL. As a stand-alone, the standard load procedure will load and automatically initialize the program.

When loading from MTL, the standard load procedure will load and initialize the MTL. The operator must respond by typing: SEX(CR).

This action will cause the exerciser to be loaded, MTL to be rewound and the exerciser to be initialized.

It is suggested that prior to loading SEX, both the HARDCORE, 706264 and AUTO, 706133 should be run (Sigma 8 and 9 only).

2.0 PROGRAM DESCRIPTION

2.1 General Overview

The System Exerciser is designed to isolate and detect system failures by approaching the system's maximum activity while exercising and testing all of the system's resources. The System Exerciser consists of a user interface, a configurator and a supervisor. It is loaded by an operator either from the Magnetic Tape Library (MTL) or directly from cards.

Configuration is established automatically by the configurator. The configurator performs a cursory operation test on each of the system elements. The operator is informed of any inconsistencies between the expected and the actual configurator and is warned of any disconnected or malfunctioning devices. After all the elements are tested, the theoretical bandwidth of the system is calculated and evaluated. Any violations are reported to the operator. After the configuration analysis is completed, a list of configuration is output.

During the configuration, one device is selected as a reboottable recovery device. A complete image of the exerciser and all of its test modules is kept on this device so that the program can be reinitialized for subsequent restarts. This device is referred to as the Base device. The system is ready to be exercised once the Base device is established.

The exerciser has handlers for every standard I/O controller and other system elements such as extra register pages, map, etc. Each handler is considered a test module to which system resources in the form of devices and memory buffers are allocated.

Once the exerciser is started, the supervisor attempts to provide service and resources to all the test modules in the fastest possible manner and thereby approach the system's maximum activity. While this is taking place, the CPU shares its time between checking buffers and fielding the test module's service request. The supervisor periodically changes system parameters which in turn cause expansion and contraction of the test module's parameters to create conflict in the system and aggravate system failures.

2.2 User Interface

2.2.1 General

The user interface is controlled by the Conversational Communications Package. Four types of communication devices are available and controlled by the communications package:

(1) Control Device: Console Teletype or Remote Terminal

(2) Observer Device: Remote Terminal or Console Teletype

(3) Message Device: Line Printer or Console Teletype

(4) Log Device: Line Printer or Console Teletype

The communication package provides the communications linkage between the controlling input device, the observing devices and the exerciser.

The communications package constantly monitors the control device so that no special action is necessary by the operator to get the program's attention.

There can be only one control device but the program can accommodate up to 64 observer devices which can be used for training purposes and remote consultation.

2.2.2 Statements

The system exerciser is always in one of two modes, the halt mode or the run mode. The halt mode, denoted by the halt prompt character H>, indicates that the system's exerciser is in an idle state. The run mode, denoted by the run prompt character R>, indicates that the system's exerciser is in a running state. Inputs are accepted by the system's exerciser in either mode. Such inputs are called statements.

All statements accepted by the system have the following formats:

DIRECTIVE, DESTINATION, QUALIFIER1, QUALIFIER2, QUALIFIER3(CR) or
DIRECTIVE, QUALIFIER1, QUALIFIER2, QUALIFIER3(CR)

Note that each field is terminated by a comma and each line terminated by a carriage return.

DIRECTIVES specify a task to be performed.

DESTINATIONS specify the object of the task (if there is a destination).

QUALIFIERS specify the boundaries of the task.

2.2.2 Statements (Cont'd.)

The Directive and Destination fields can be abbreviated to the minimum character set which makes the term unique. A directive can only be accepted after a prompt character (>) has been issued by the system.

The first and second qualifiers are boundary qualifications. Qualifier One is the lower limit. If Qualifier Two is equal or greater in magnitude than Qualifier One, it is interpreted as an upper limit. If Qualifier Two is less in magnitude than Qualifier One, it is interpreted as a count or displacement from Qualifier One. The third qualifier is a value which has discrete meaning depending on the directive. The default value of all qualifiers is zero.

2.2.2.1 Directives

A directive specifies a task to be performed. The following are descriptions and examples of each directive. All qualifiers are in hex unless otherwise stated. To simplify the description, input from the operator will be underlined. The term ''(CR)'' refers to carriage return.

2.2.2.1.1 ABSTRACT,Q1(CR)

Outputs a cursory description of the passes, the phases, the error file, the directives, and the destinations to the printer if Q1 = 0 or the teletype if Q1 ≠ 0.

H>ABS

H>

2.2.2.1.2 AIO(CR)

Outputs a list of controller addresses which are unable to interrupt because of a manual state, non-operational state, etc.; and then outputs the remaining controller addresses in descending order of their interrupt priority.

```
H>AIO

   0003 NO INTERRUPT PENDING
   0007 NO INTERRUPT PENDING
   0206 NO INTERRUPT PENDING
   0207 NO INTERRUPT PENDING
   020C NO INTERRUPT PENDING
   0280 NO INTERRUPT PENDING
   0288 NO INTERRUPT PENDING

PRIORITYS

   00E0
   0002
   0004
   0001
   02E0
   02F0

H>
```

2.2.2.1.3 BOOT,Q1(CR)

Simulates an auto fill from the device specified in Q1. MTL can be recalled without disconnecting the remote console. Through this directive, the operator (remote or local) can switch back and forth between SEX and any other MTL Diagnostic Program.

```
H>BOOT,180
```

2.2.2.1.4 BRANCH,Q1(CR)

Transfers control to the location specified by Q1 and establishes run mode. This directive should be used in conjunction with the SNAP directive.

```
H>BRA,3000

R>
```

2.2.2.1.5 COMPARE,DESTINATION,Q1,Q2,Q3(CR)

Compares the contents of the destination from Q1 to Q2 on the value Q3. Locations which have ones in bit positions which correspond to the ones in bit positions in Q3 will be output.

```
H>COM,M,3000,10,-1

003002 00000001
003006 00000001
00300C 68400FB9
H>
```

2.2.2.1.6 CONFIGURE(CR)

Recalls the configurator. The program responds as if it has just been loaded.

```
H>CON

SIGMA 5/6/7/8/9 SYSTEM EXERCISER(SEX)
PROGRAM NO. 705889-B03 (V238) MANUAL NO. 901737B-3
```

2.2.2.1.7 DESELECT,Q1,Q2(CR)

Deselects the controller specified by the CTX in Q1 and the units specified by bits 0 through 15 of Q2.

```
H>DES,1              (Deselect controller whose CTX = 1)

H>DI,C:3,1           (Display demonstrates controller was deselected
                      because bit 0 of C:3 is zero.)

001 20BC
H>
```

2.2.2.1.8 DISPLAY,DESTINATION,Q1,Q2(CR)

Outputs the contents of the specified destination from Q1 through Q2 on the TTY.

```
H>DI,M,300C,4

00300C 68400FB9 00000000 00000000 00000000
H>
```

2.2.2.1.9 EXPLAIN,DESTINATION,Q1(CR)
EXPLAIN,DIRECTIVE,Q1(CR)

Outputs explanation of the specified destination or directive to the printer if Q1 = 0 or the TTY if Q1 ≠ 0.

```
H>EXP,EXP              (An explanation of the EXPLAIN directive is output
                        to the printer.)
H>

H>EXP,M,1              (An explanation of the MEMORY table is output to
                        the TTY.)

MEMORY TABLE
MEMORY TREATED AS WORD TABLE.  Q1 & Q2 ARE ADD LIMITS
H>
```

2.2.2.1.10 ERRORS,Q1,Q2,Q3(CR)

Outputs all errors whose sequence numbers lie between Q1 and Q2 and whose IDENTS are Q3, to the LOG device (specified by ''LOG'' in the operator table). If Q1 and Q2 are zero, the entire range of sequence numbers are used. If Q3 is zero, all IDENTS are output. If Q1 is negative, the error file is reset.

```
H>ERR,0,5,E4040        ALL IDENTS of E4040 (nonexistent memory address).
                       Logs whose sequence lies between 0 and 5 are out-
H>                     put to the log device.
```

2.2.2.1.11 HALT(CR)

Imposes halt or idle.

```
R>HA

H>
```

2.2.2.1.12 HIO,Q1,Q2,Q3

Issues an HIO to all device addresses Q1 through Q2 and outputs the status response and resulting condition codes of those addresses which are recognized (condition code response other than X'C0'). The output is issued to the teletype if Q3=0 or the printer if Q3≠0.

```
H>HIO,1

DEV CC   CPDA     ST  PC     COM1     COM2
001 40 00000AE0 76000000 05005708 08000001
H>
```

CPDA = Command Pair Doubleword Address (even register status response)

ST BC = Status and Byte Count (odd register status response)

COM1 COM2 = Command Pair pointed to by the CPDA.

2.2.2.1.13 READ,Q1,Q2,Q3(CR)

Reads from the device specified by Q1, the seek value specified by Q2 (4 bytes) and byte address specified by Q3. The length of the transfer is defined by ''BUFSIZ'' of the system table. If the device specified is not found in ''DEV'' of the CONTROL table, the directive will abort. If Q2 is negative, the corresponding value found in ''SEEK'' of the CONTROL table will be used. If Q3 is zero, the corresponding address in ''BUFFER'' of the CONTROL table will be used.

This directive was primarily designed to aid the operator in inspecting the RAD surface when a ''DATA ERROR'' has occurred. (Refer to Paragraph 2.5.3.15)

```
H>READ,2E0,,8000     (Reads from Device 2E0, track 0, sector 0, to
                      memory location 2000 hex.)
H>
```

2.2.2.1.14 PRINT, DESTINATION,Q1,Q2,Q3(CR)

Prints the contents of the specified destination from Q1 to Q2 to the printer. The outputs will be preceded by Q3 + 1 line upspaces. If Q3 is 16 or greater, the paper is advanced to a new page.

```
H>P,M,3000,3007

H>
```

2.2.2.1.15 REDUMP,Q1(CR)

Rewrite the program, the CONTROL table and configurator or some combinations of the three to the base RAD. This directive cannot be executed until configuration has been completed and bit 2 (X'20') in ''SM'' of the SYSTEM table has been set. Q1 specifies whether the program or the CONTROL table is to be written:

Q1 = 0 write program and control table

1 write program

2 write CONTROL table

3 program restart

```
H>RED,1                          (Write program to base device)

!!SYSTEM RESTARTED.
!!TYPE RUN TO START EXERCISE.
H>
```

2.2.2.1.16 RELOAD,Q1(CR)

Reads the program, the CONTROL table, the configurator, or any combination of the three from the base RAD and then restarts the program. This directive cannot be executed until configuration has been completed.

Q1 specifies the option:

Q1 = 0 Read configurator, program and control table

1 Read configurator and program

2 Read configurator and control table

3 Read configurator

4 Read program and control table

5 Read program

6 Read control table

7 Restart program

```
H>REL,1                          (Reloads program from base device)

!!SYSTEM RESTARTED.
!!TYPE RUN TO START EXERCISE.
H>
```

2.2.2.1.17 REPLACE,DESTINATION,Q1,Q2(CR)

Displays and then allows the operator to replace the contents of the specified directive from location Q1 through Q2. The directive acts like the ''DISPLAY'' and ''STORE'' directives combined.

```
H>REP,M,3000,4

003000 00000000 00000000 00000001 00000000
003000 ,''''''''1234,''''4;''''''',
H>DI,M,3000,4

003000 00000000 00001234 00000005 00000000
H>
```

The second part of this example demonstrates how locations 3000 through 3003 were changed. Note that a comma (,) preceded by data indicates replace and a semicolon (;) indicates exclusive OR (Refer to Paragraph 2.2.3).

2.2.2.1.18 RUN(CR)

Restores RUN mode.

```
H>RU

R>
```

2.2.2.1.19 SELECT,Q1,Q2(CR)

Selects the controller specified by the CTX in Q1 and the units specified by bits 0 through 15 of Q2.

```
H>SEL,1                    (Select controller whose CTX = 1)
                            Display demonstrates controller was selected
H>DI,C:3,1                  because bit 0 of C:3 is zero.

001 A0BC
H>
```

2.2.2.1.20 SEARCH,DESTINATION,Q1,Q2,Q3(CR)

Searches the destination specified from Q1 to Q2 for the value. Specified in Q3 masking out those digit positions which are not present in Q3. This directive can be used to search destinations for matches in the least significant digit positions of each destination element. It can also be used to search for complete matches of each element.

```
H>SEAR,M,C77,8,003

000C79 D7D00003
000C7B D6D00003
H>
```

2.2.2.1.21 SIO,Q1,Q2,Q3(CR)

Loads register zero with the contents of Q2, issues an SIO to the devices specified by Q1 and outputs the condition codes and status responses to the teletype if Q3 = 0, or the printer if Q3 ≠ 0. The status output refers to the state of the device prior to the SIO.

```
H>SIO,2E0,2482/2      (The output is to the teletype because Q3 = 0).

DEV CC   CPDA     ST  BC     COM1     COM2
2E0 00 00001242 10800000 01000100 0A009AB8
H>
```

2.2.2.1.22 SNAP,Q1,Q2,Q3(CR)

Causes the PSW1 to be output to the teletype prior to every CPU execution of the instruction in the location specified by Q1. This output is followed by an output of the location and contents of the location specified in Q2. If Q2 is less than HEX 10, the address and contents of the corresponding register is output. If Q3 is negative, the program will halt after outputting the PSW1. If the BREAK switch is depressed during the output, the program will halt at the SNAP as if Q3 were negative.

```
H>SN,456,6          (Outputs PSW1, address and contents of Register
                     6 each time the CPU executes the instruction
H>                   in location HEX 456.)
```

2.2.2.1.23 SPREAD,DESTINATION,Q1,Q2,Q3(CR)

Stores the value specified by Q3 into locations Q1 through Q2 of the specified destination.

```
H>SPRE,M,3000,8,-1

H>
```

2.2.2.1.24 START,Q1,Q2(CR)

Starts the exerciser in the pass specified by Q1 and phase specified by Q2. Caution must be employed when switching phases to avoid not keying devices which must be keyed to pass the data checking test.

```
H>STA,1                    (The system always announces pass changes via
                            the teletype.)
R>
!!PASS 1
R>
```

2.2.2.1.25 STORE,DESTINATION,Q1,Q2(CR)

Allows the operator to store desired values into the specified destination from location Q1 through Q2. A comma (,) must follow all values to be stored. As in the case of the REPLACE directive, a semicolon (;) specifies the value input is to be exclusive OR'ed with the corresponding field.

```
H>STO,M,3000,2

003000 1,,,,,,,,1,
H>DI,M,3000,2              (The display demonstrates that the locations
                            were altered.)
003000 00000001 00000001
H>
```

2.2.2.1.26 SWITCH,Q1(CR)

This directive can be used only when a remote observer has logged on. It allows control of the system to be passed from the control device to the observer whose line number is specified in Q1. If Q1 = X'80', control is transferred to the local teletype.

```
H>SW,80

H>
```

2.2.2.1.27 TIO,Q1,Q2,Q3(CR)

Tests and outputs the condition codes and operational status to all device addresses Q1 to Q2 to the teletype if Q3 = 0 or the printer if Q3 ≠ 0. Those devices which respond with condition codes of 1100 are excluded from the output.

```
H>TIO,0,3

DEV CC   CPDA     ST   BC     COM1     COM2
001 40 00000AE0 76000000 05005708 08000001
002 00 00001247 10800000 000092CD 0A000037
003 00 000016E3 00000000 00000000 00000000
H>
```

2.2.2.1.28 TDV,Q1,Q2,Q3(CR)

Tests and outputs the condition codes and device status to all device addresses Q1 to Q2 to the teletype if Q3 = 0 or the printer if Q3 ≠ 0. Those devices which respond with condition codes of 1100 are excluded from the output.

```
H>TDV,2E0

DEV CC   CPDA     ST   BC     COM1     COM2
2E0 00 00001242 00800000 01000100 0A009AB8

H>
```

2.2.2.1.29 UNSNAP(CR)

Removes the SNAP and restores normal operation. If SNAP is already removed, the directive is ignored.

```
H>UNS

H>
```

2.2.2.1.30 WRITE,Q1,Q2,Q3(CR)

Writes to the device specified by Q1, the seek value specified by Q2 (4 bytes) and byte address specified by Q3. The length of the transfer is defined by ''BUFSIZ'' of the system table. If the device specified is not found in ''DEV'' of the CONTROL table, the directive will abort. If Q2 is negative, the corresponding value found in ''SEEK'' of the CONTROL table will be used. If Q3 is zero, the corresponding address in ''BUFFER'' of the CONTROL table will be used.

```
H>WRITE,2E0,,8000

H>
```

2.2.2.2 Destinations

A destination is the object of a directive's task. There are two types of destinations: tables which are a serial string of data elements each with the same number of digits; and, arrays which consist of many tables grouped together each with the same number of elements.

Tables may be displayed in vertical or horizontal manner while arrays are always displayed in a matrix like fashion. There are only eight directives that can act on a destination. They are:

(1) COMPARE

(2) DISPLAY

(3) EXPLAIN

(4) PRINT

(5) REPLACE

(6) SEARCH

(7) SPREAD

(8) STORE

The following are descriptions and display examples of each destination.

Throughout the description, the following rules prevail:

(1) All table descriptions have both a reference and a heading label. The reference label is used to directly reference the table as a destination. This label is used throughout these descriptions to reference a particular table of an array. This label is identifiable by a single letter followed by a colon (:) and a decimal digit which specifies the table in the array. The heading label is enclosed by parenthesis. This label appears only when the array is display.

(2) Those tables whose reference labels are followed by an asterisk cannot have their contents altered by the operator without setting the system protect inhibit bit in SM (X'20') of the system table.

(3) The contents of all tables are in hexadecimal unless otherwise stated.

2.2.2.2.1 BYTES

This destination treats memory as a string of bytes. A special way of addressing memory has been devised to aid the user when dealing with this destination. The byte address format uses a word address and a byte position separated by a period to specify a particular byte address. For instance, the byte address 4F831 would equal 13E0C.1 in byte address format. The byte destination is considered a table consisting of bytes. The table has a horizontal display format.

```
H>DI,B,3000.0,C

003000.0 00 00 00 01 00 00 00 01 00 00 00 00
H>
```

2.2.2.2.2 COCLINE

This is a byte table which contains information concerning the state of the COCLINES associated in the System Exerciser's Remote Interface. The table is 64 bytes in length and each table element is addressed by its corresponding line number (i.e., Element 9 is for line 9, etc.)

Once the operator establishes the remote interface by putting the COC address into COC of the OPERATOR table, the state of each COC line can be querried via the COCLINE table. The following is a list of the states and their meaning:

00	line is hung and cannot be answered
03	line is answered and waiting for someone to dial in
10-1F	transmitting log on message
20	awaiting first password* character (escape)
22	awaiting second password* character (escape)
24-27	transmitting ''ON'' message
40	line has observer
80	line has remote control teletype

* Password - when the user dials into the system, the following message occurs:

SEX B03:

2.2.2.2.2 COCLINE (Cont'd.)

The user must respond by typing two consecutive escapes. The system will respond by typing the line number:

!!0001 ON

If the user types the wrong characters, the following message occurs:

????SEX B00:

H>DI,COC,0,4

000 00 00 00 00 00
H>

2.2.2.2.3 CONTROL

Control is an array of thirteen tables containing information corresponding to those controllers which have been configured into the system via the configurator. The length of the tables is determined by the number of configured controllers at configuration time. Each controller is assigned a Control Table Index (CTX) in ascending order based on transfer rate with the fastest Controller's CTX = 0001. CTX value of zero is reserved for the CPU so that the CPU can be controlled in the same manner as any I/O controller in the system.

The CONTROL array appears as a matrix whose dimensions are CTX by C:13 tables.

The following is a description of each table in the array:

CTX* = Control Table index. The ''Y'' coordinate of CONTROL matrix.

C:1* = Device Address of the currently running device (DEV).

C:2* = Specifies which units are to be exercised, i.e, C000 = units 0 and 1 (UNIT).

C:3 = Controllers Operation Selection. The user may alter as desired (OPS).

2.2.2.2.3 CONTROL (Cont'd.)

The following is a description of the Selection parameters.

Bit 0 (8000) = Controller is selected to be exercised.

Bit 1 (4000) = Do not automatically alter the C:5 (SEEK) or C:6 (BUFFER) of this controller.

Bit 2 (2000) = This controller may be selected as a source device.

Bit 3 (1000) = This controller does not need to be keyed.

Bit 4-6 (E00) = Unassigned.

Bit 7 (100) = This controller is to be turned on and off in accordance with the burst timer (BON & BOF) of the status table.

Bit 8 (80) = It is legal for this controller to have data overruns if it is on a legally overconfigured IOP.

Bit 9 (40) = Do not check the header or buffer data transferred by this controller.

Bit 10 (20) = This controller can perform read operations.

Bit 11 (10) = This controller can perform write operations.

Bit 12 (8) = Transfer each device's total surface before selecting another device. Otherwise, select a different device each transfer.

Bit 13 (4) = Select seek addresses serially rather than randomly.

Bit 14 (2) = Every buffer read from this controller's devices must be checked by the CPU prior to redistribution.

Bit 15 (1) = Every buffer to be written to this controller's devices must be first checked by the CPU.

2.2.2.2.3 CONTROL (Cont'd.)

C:4* = Controller's Current Operational Status (CS). This is a dynamic display of what the controller is doing.

Bit 0 (8000) = Controller is busy transferring data.

Bit 1 (4000) = Unassigned.

Bit 2 (2000) = The control is selected as SOURCE for this pass.

Bit 3 (1000) = The controller is currently being keyed.

Bit 4,5&6(E00) = Unassigned.

Bit 7 (100) = Controller is currently retired waiting for burst timer to turn on.

Bit 8 (80) = Controller is currently retired waiting for data overrun wait to expire.

Bit 9 (40) = Controller's surfaces have been keyed.

Bit 10 (20) = Controller currently retired waiting for a buffer.

Bit 11 (10) = Controller is sharing devices with another controller.

Bit 12 (8) = Controller has manual devices.

Bit 13 (4) = Controller is waiting on error recovery.

Bit 14 (2) = Controller is suspended from operation.

Bit 15 (1) = Controller is waiting to be initialized by handler.

C:5 = Current Seek Address (SEEK).

C:6 = Current buffer address in byte address format (BUFFER).

2.2.2.2.3 CONTROL (Cont'd.)

C:7* = Word address pointing to Command String being executed (COMS) (Refer to Paragraph 2.4.5.1.).

C:8* = Word address pointing to surface maps for the devices (MAPS) (Refer to Paragraph 2.4.5.1.).

C:9* = Current I/O command (OP).

C:10* = Controller Address (CA).

C:11* = Composite number of errors associated with controller's devices (EC).

C:12* = Controller's associated Element Table Index (ETX).

C:13* = CTX of next controller on IOP (REL).

```
H>DI,CON,1,2

CTX DEV  UNIT OPS  CS   SEEK     BUFFER   COMS   MAPS   OP  CA  EC ETX REL
001 02E0 8000 A0BC 0200 00000000 000000.0 002928 00293A 00 02E0 00 002 002
002 02F0 8000 A0BC 0000 00000000 000000.0 00292E 00294A 00 02F0 00 004 001
H>
```

2.2.2.2.4 ELEMENT

Element is an array of five tables containing information corresponding to the types of controllers which the system exerciser can accommodate. The tables are 40 elements in length with each controller assigned on Element Table Index (ETX) = 001. Index zero is unassigned. The array appears as a matrix whose dimensions are ETX by E:5 tables.

The following is a description of each table in the array.

ETX* = Element Table Index. The ''Y'' coordinate of the ELEMENT matrix.

E:1* = Controller Model number in four decimal digits (MOD).

E:2* = Two letter mnemonic for the controller (MN).

E:3* = Word address pointing to the controller's I/O handler (HANDLR).

E:4* = Relative PARAMETER table index (RPX). This index is used to access a set of tables which contain parameters concerning the controller (Refer to Paragraph 2.2.2.2.8, PARAMETER).

2.2.2.2.4 ELEMENT (Cont'd.)

E:5* = Percentage of the IOP bandwidth (PBW) associated with the use of this controller under maximum load conditions expressed in decimal.

```
H>DI,EL,2,4

ETX MOD  MN HANDLR RPX PBW
002 7231 EP 00081F 002 078
003 7235 EP 000B08 002 042
004 7201 MS 00081F 003 040
H>
```

2.2.2.2.5 IOP

IOP is an array of four tables containing information corresponding to those IOPs which have been configured into the system via the configurator. The length of the tables is determined by the number of configured IOPs. IOPs are indirectly configured through the controllers on the system. The IOP array appears as a matrix whose dimensions are IOP by I:4 where IOP indicates the IOP number.

The following is a description of each table in the array:

IOP* = IOP number.

I:1 = Data Overrun Control (DOC). This indication is used to specify that the IOP is overconfigured and that data overruns are allowable for controllers on this IOP only. The operator must input a value other than zero to set the indication. This action should be based on the configuration bandwidth analysis which automatically occurs at configuration time.

I:2* = This indicates the CTX of the last controller on this IOP to be retired for legal data overrun (LST).

I:3* = This table indicates the IOPs maximum transfer rate in thousands of words per second in decimal (KWPS). It can have one of three values:

	SIGMA 5-7	SIGMA 8-9
Integral IOP	100	-
Multiplex IOP	125	150
RAD IOP Selector IOP	750	750

2.2.2.2.5 IOP (Cont'd.)

I:4* = Percentage of the IOPs bandwidth (PCNT) used in the current configuration. The percentage is in decimal.

```
H>D,IOP,0,4

IOP  DOC  LST  KWPS  PCNT
000  000  008  0200  0070
001  000  005  0200  0080
002  000  001  0750  0090
003  000  000  0000  0000
004  000  000  0000  0000
H>
```

2.2.2.2.6 MEMORY

This destination treats memory as a string of words. Addressing is done in the normal fashion. The memory destination is considered a table whose elements are words and has a horizontal display format.

```
H>DI,M,3000,3

003000 00000001 00000001 00000000
H>
```

2.2.2.2.7 OPERATOR

Operator is an array of thirteen tables containing information corresponding to the operator's communications interfaces. The length of all tables is one so that the array is usually referred to as a table (i.e., OPERATOR table). The following is a description of each table in the array.

O:0 = Denotes the address of the Message Device (MSG). The message device is the object of the print directives output. If there is a line printer on the system, it will automatically become the message device, otherwise the control teletype will become the message device.

O:1 = Number of characters per line of control teletypes output (CH) in decimal. The local teletype can handle up to 84 characters per line while a remote teletype, Model 33, can handle only 73. This table allows the operator to establish his own limits. The default value is 80.

2.2.2.2.7 OPERATOR (Cont'd.)

O:2 = Denotes the address of the LOG device (LOG). The log device is the object of all ERROR FILE output. If there is a line printer on the system, it will automatically become the log device, otherwise, the control teletype will become the log device.

O:3 = ERROR LOG severity level (L). A hexadecimal value from 0 to F which establishes the lowest severity of error which is to be logged. Associated with all loggable errors is a value which denotes the severity of the error. Refer to Paragraph 2.5.2 for a list of the severity levels and their associated error types. The default value is 0.

O:4 = ERROR HALT severity level (H). A hexadecimal value from 0 to F which establishes the lowest severity of error which is to impose halt mode on the Exerciser. All errors whose severity is equal to or higher than H will cause the system to halt. Refer to Paragraph 2.5.2 for a list of the severity error levels. The default value is 4.

O:5 = Address of the local teletype (TTY).

O:6 = Address of the Remote Interfaces COC (COC). To initiate the remote interface the operator must set this address appropriately. A watchdog timer trap could occur if a wrong address is input.

O:7 = Allows the operator to establish the maximum number of observers he wishes to log on (MX). The default is X'40'.

O:8* = Current state of the Remote Interface (CS). This is used by internal SEX subroutines.

00 = No COC address established.

03 = COC address established but no one logged on yet.

43 = At least one observer logged on.

83 = One line logged on and is control teletype.

C3 = More than one line logged on and one is control teletype.

O:9 = Line number of the Control Unit (CL). If a remote unit is control, the appropriate line number appears here. If the local teletype is control, then CL = 80.

2.2.2.2.7 OPERATOR (Cont'd.)

O:10* = Indicates number of remote units currently logged on (CN).

O:11* = Indicates the address of the Base device (BASE).

O:12* = Indicates the address of the Current Source Device (SORS).

The next two tables are not included in the operator array but are available by querying the specified reference labels.

O:13* = Number of buffers accessed by the CPU during the current pass.

O:14* = Number of I/O interrupts, during the current pass.

```
H>DI,OP

MSG  CH LOG  L H TTY  COC  MX CS CL CN BASE SORS
0002 80 0002 0 4 0001 0000 40 00 80 00 02E0 00E0
H>
```

2.2.2.2.8 PARAMETER

Parameter is an array of ten tables containing information corresponding to the Multi-Unit Controllers in the ELEMENT table. All controllers which are in the CONTROL tables must have first been found in the ELEMENT table and if it is a Multi-Unit Controller it can also be found in the PARAMETER tables. The tables are ten in length with index zero unassigned. Each index is assigned to a particular model number. The following is a list of RPXs verses model numbers:

(1) 7211 - High Speed RAD

(2) 7231 - Extended Performance RAD

(3) 7201 - Medium Speed RAD

(4) 7240 - Disc Pack

(5) 7251 - New RAD

(6) 7252 - New Pack Low Density

(7) 7253 - New Pack Hi Density

(8) 7330 - New Tape

(9) 7320 - 9 Track

(10) 7361 - 7 Track

2.2.2.2.8 PARAMETER (Cont'd.)

The array appears as a matrix whose dimensions are RPX by 10 tables. The following is a description of each table in the array.

RPX* = Relative Parameter Index. The ''Y'' coordinate of the PARAMETER matrix.

BPX* = Number of bytes per sector.

SPH* = Number of sectors per head.

HPT* = Number of heads per track.

TPD* = Number of tracks per device.

DPC* = Number of allowable devices per controller.

INC* = Number of sectors devoted to current passes buffer size.

BPD* = Number of buffers per device based on current passes buffer size.

```
H>DI,PAR,1,4

ETX BPS  SPT TPD  DPC INC  BPD
001 0400 52  0040 04  02D  00074
002 0400 0C  0200 08  02C  0008B
003 0168 10  0200 08  000  00000
004 1800 14  00CB 08  009  001C3
H>
```

2.2.2.2.9 REGISTER

Register is a table which treats the register contents prior to each monitor entry or SNAP execution as a string of 16 consecutive words addressable by corresponding register number. The table has a horizontal display format.

```
H>DI,R,C,4

000 00000000 0000000D 70094690 6666580C
H>
```

Displays the contents of registers 12 through 15.

2.2.2.2.10 STATUS

Status is an array of thirteen tables containing information corresponding to the Exerciser's current pass and phase. The length of its tables is one so that the array is usually referred to as a table (i.e., the STATUS table). The status table is an extension of the system table (Refer to Paragraph 2.2.2.2.11) therefore, the reference tables start with S:8.

The following is a description of each table in the array:

S:8 = Allows the operator to specify a particular cycle (CS). Default is X'FF'.

S:9* = Current Cycle Indicator (CI).

S:10 = Pass Selector (PS). Allows operator to specify any of 8 passes (Pass 0 = X'80'). The default value is X'FE' which selects passes 0 through 6 automatically.

S:11* = Current Pass Indicator (PI).

S:12 = Phase Selector (FS). Allows the operator to select any of 4 phases (Phase 1 = X'40'). The default value is X'F0' which selects phases 0 through 3.

S:13* = Current Phase Indicator (FI).

S:14 = Amount of time the burst timer is to be on (BON).

S:15 = Amount of time the burst time is to be off (BOF).

S:16 = Maximum number of records to be accessed on tape (RECL). The default value is 256 (X'100').

S:17* = Amount of time remaining in the current pass in tenths of seconds (PASSTIME).

S:18 = First memory buffer location (MSTART).

S:19 = Last core location plus one (MEMSIZ).

S:20 = Register pages (REGOPS).

```
H>DI,STA

CS CI PS PI FS FI BON BOF RECL PASSTIME MSTART  MEMSIZ  REGOPS
FF 00 FE 01 F0 00 005 055 0100 00000000 00258A  018000  F0000000
H>
```

2.2.2.2.11 SYSTEM

System is an array of eight tables containing information corresponding to the control of the exerciser.

The following is a description of each table in the array:

S:0 = Number of bytes per buffer (BLKSIZ) for current pass.

S:1 = Number of bytes to be transferred per buffer (BUFSIZ) for current pass.

S:2 = Byte format address of first available buffer for current pass (FIRSTBUF).

S:3 = Byte format address of first location the operator selected to be exercised (STRTCORE).

S:4 = Byte format address of last location the operator selected to be exercised (ENDCORE).

S:5 = System Control Modes (SM). The following is a description of the available selections:

BIT0 (80) = Inhibit pass and phase change from occurring. Once this inhibit is set, the START, RELOAD, & REDUMP directive can override and reset the inhibit.

BIT1 (40) = Unassigned.

BIT2 (20) = Inhibit the system protect. When this bit is reset, all tables whose reference labels are followed by an asterisk are write protected. Also, the REDUMP directive is aborted if this bit is reset. This bit is also used to override automatic configuration (Refer to Paragraph 2.3.2).

BIT3 (10) = Inhibits error filing and logging. If this bit is set, all errors are ignored.

S:6 = Data Selection Byte (DS). The following is a list of the various selections available to the operator via this byte. The first four bits apply to pass 5 only. The last four apply to all other phases.

BIT0 (80) = Exercise SF, DW, MW, DH and MH instructions (Refer to Paragraph 2.4.5.2.2).

2.2.2.2.11 SYSTEM (Cont'd.)

BIT1 (40) = Exercise FSL, FAL, FDL, FML, SF, FSS, FAS, FDS, and FMS instructions.

BIT2 (20) = Exercise PACK, UNPK, DS, DA, DD, DM, DSA, DC, DL and DST instructions.

BIT3 (10) = Exercise TTBS, TBS, CBS, and MBS instructions; also EBS if BIT 2 also set.

BIT4 (8) = Fill all buffers with value specified in S:7.

BIT5 (4) = Random data: M = M_i, C = 1 (Refer to Paragraph 2.4.5.2.1).

BIT6 (2) = Sequential data: M = 1, C = 0.

BIT7 (1) = Fixed data: M = 0, C = 0.

S:7 = Specifies only one instruction to be tested (SO).

```
H>DI,SY

BLKSIZ BUFSIZ FIRSTBUF STRTCORE ENDCORE  SM DS SO
000010 00000C 00296A.0 00296A.0 018000.0 20 F7 00
H>
```

2.2.2.2.12 TIME

Time is an array of 6 tables which are concerned with the current date and time. The tables are only displayable as an array. The operator cannot individually display any of the tables. The tables are a length of one, therefore, the array is usually referred to as a table (i.e., the TIME table). The following is a description of each table in the array. All tables are in decimal.

MO = Month - two digits

DY = Day of the month - two digits

YR = Year - least significant two digits of the year (i.e., 1971 appears as 71)

HR = Hours - two digits

MN = Minutes - 2 digits

SC = Seconds - 2 digits

```
H>DI,T

MO DY YR HR MN SC
07 22 71 16 35 55
H>
```

This time indicates July 22, 1971 at 4:35 PM plus 55 seconds.

2.2.2.3 Qualifiers

The qualifiers determine the limitations of the directive's task. If a destination is involved, the destination determines if the qualifiers are to be expressed in decimal or hex. If there is no destination, the qualifiers are in hex.

For most of the directives, the qualifiers bear the following relationships:

(1) If Q1 > Q2, Q1 is a lower limit and Q2 is a displacement.

(2) If Q1 ≤ Q2, Q1 is a lower limit and Q2 is an upper limit.

(3) Q3 is always an object value.

The following is a list of directives which violate this relationship:

(1)	DESELECT	Q1 = CTX	Q2 = Units Selected
(2)	READ	Q1 = Device Address	Q2 = Seek Address
(3)	SELECT	Q1 = CTX	Q2 = Units Selected
(4)	SIO	Q1 = Device Address	Q2 = Doubleword Address of Command Pair
(5)	START	Q1 = Pass	Q2 = Phase
(6)	WRITE	Q1 = Device Address	Q2 = Seek Address

The BYTE destination violates the relationship by forcing Q2 to always be a displacement. Q1 is always in byte address format. It should be pointed out that all hex qualifiers can be expressed in decimal by use of the decimal operator (Refer to Paragraph 2.2.3).

2.2.3 Single Character Operators

There are a set of characters which cause the system exerciser interface to respond upon their receipt. The following is a list of these characters and a description of the system's response.

2.2.3.1 COMMA ,

The comma is the field delimiter for all exerciser input. When the exerciser interface sees a comma it terminates the current field input and initiates the next communications function. When inputting statements, the comma is used to separate the fields.

2.2.3.1 COMMA , (Cont'd.)

When an input statement is being executed (i.e., STORE) the comma is used to separate input data fields. When no data precedes a COMMA, the field is skipped.

2.2.3.2 SEMI-COLON ;

This is interpreted the same as a comma except when an input statement is being executed. Under these circumstances, the input field is exclusive OR'ed with the value it is to replace. A comma would have stored the input field.

2.2.3.3 CARRIAGE RETURN

The return is used to terminate a statement input and initiate its execution. It is also used to terminate an input statement's execution.

2.2.3.4 LESS THAN <

Resets the current input field. This allows the operator to reinitiate the current input field when an incorrect character has been typed.

2.2.3.5 GREATER THAN >

Resets the current statement. This allows the operator to reinitiate statement mode.

2.2.3.6 QUESTION MARK ?

This causes the system to output an explanation of the most recent destination accessed.

2.2.3.7 COLON :

This character is the decimal operator. Whenever the system sees a colon, the following string of characters are converted to decimal.

2.2.3.8 OPEN PARENTHESIS (

This causes the system to enter message mode. All input characters are subsequently ignored except for closed parenthesis which is explained below.

2.2.3.9 CLOSED PARENTHESIS)

This terminates message mode. All characters enclosed by parenthesis are completely ignored and the input resumes as if it had not been interrupted.

2.2.3.10 PERIOD .

This is used to separate word address and byte position when byte address format has been imposed.

2.2.3.11 PLUS +

This is an arithmetic operator. All subsequent input up to the next comma or arithmetic operator will be added to the preceding input. Only the least significant 32 bits are retained.

2.2.3.12 MINUS -

This is an arithmetic operator. All subsequent input up to the next comma or arithmetic operator will be subtracted from the preceding input. Only the least significant 32 bits are retained.

2.2.3.13 MULTIPLY *

This is an arithmetic operator. All subsequent input up to the next comma or arithmetic operator will be multiplied by the preceding input. Only the least significant 32 bits are retained.

2.2.3.14 DIVIDE /

This is an arithmetic operator. All subsequent input up to the next comma or arithmetic operator will be divided into the preceding input. The least significant 32 bits of the quotiant are retained. The remainder is lost unless the divisor is followed by an equal sign.

2.2.3.15 EQUAL SIGN =

This is an arithmetic operator. The accumulated value of all preceding arithmetic operations is output. This operator should only be used at the statement level.

```
H>1234+9=123D                    (result is in hex)
H>:35=23                         (convert decimal to hex)
H>23:=35                         (convert hex to decimal)
H :125/7=11 6                    (decimal arithmetic converted to hex)
H>12+17-3*25/:11:=127 9          (operators can be strung out)
H>
```

2.2.3.16 BREAK (short)

The operator may break console messages a line at a time by depressing the break key for one second or less.

2.2.3.17 BREAK (long) or ESCAPE

The operator may terminate console messages by depressing the break key for two seconds or more on the local KSR or by depressing the escape key if it is a Remote Terminal. The message will terminate and halt mode will be imposed.

2.2.4 Syntax Error Messages

When the operator makes a mistake or attempts to initiate a request which cannot be honored, a syntax error message is issued. The following is a list of syntax error messages and the circumstances under which they occur.

2.2.4.1 ***INVALID QUALIFIER

This message is issued whenever the operator inputs a qualifier 1 which exceeds the upper boundary limit of the specified operation. For example, if the Q1 for a memory display was out of bounds. Once the message is issued the system retypes the input up to but excluding Q1. Input may proceed from there.

2.2.4.2 ***INVALID CHARACTER

This message is issued when an invalid character is detected. Upon completion of the message, the input line is retyped up to but excluding the errored field. The input may proceed from this point.

2.2.4.3 ***INVALID REQUEST

This message is issued when:

(1) The operator is attempting to store into a write protected table and the system inhibit is not set.

(2) The operator is attempting a REDUMP directive and the system inhibit is not set.

(3) Attempt to start a nonexistent pass (PASS>7).

(4) Attempt to start a nonexistent phase (PHASE>3).

(5) Attempt to SWITCH remote control to a nonexistent line.

(6) Attempt to SELECT a nonconfigured CTX.

(7) Attempt to DESELECT a nonconfigured CTX.

(8) Attempt to READ a device which cannot be found in C:1 of the control table.

(9) Attempt to WRITE a device which cannot be found in C:1.

(10) Attempt to BOOT from a nonexistent device.

(11) Attempt to RELOAD or REDUMP prior to completion of configuration.

Upon completion of this message, the statement is aborted and input reinitialized.

2.2.4.4 ***SELECTION ERROR

This message is issued if a nonexistent directive or destination is specified. Upon completion of the message, the statement is retyped up to but excluding the erroneous field.

2.3 Configurator

2.3.1 General

The configurator is automatically entered at load time or at the operator's request via the CONFIGURATOR directive. Its first task is to determine and output all the CPU and memory option. Then, it identifies all controllers and enters them into the configurator table. Next, it allows the operator to update any erroneously defined or missing controllers. When the declaration is completed, the IOP and memory bandwidths are computed and analyzed and discrepancies are reported. Once the analysis is completed, the IOP, Configuration, Control, System, Status and Operator tables are output. Lastly, the system declares a rotating memory as a rebootable recovery device. These tables determine the strategy by which the system is exercised. When the operator agrees with these tables, the exerciser's image is written to a rebootable recovery device (base device) and the exerciser may be started.

2.3.2 Pre-configuration

The configurator's first task is to determine the size of memory. When this is done, it determines and outputs the CPU type. Then it determines and outputs the floating point, decimal, map and write lock options. It also specifies how many register blocks are available. All interrupt groups and levels are output. The configurator then searches for the device controllers. All such controllers which are not definable because they are busy or are manual are specified as not operational and excluded from the configurator table. In addition, all controllers which share a common channel address are defined as duplicate channel addresses but are still included in the configurator table. The configurator can determine if there are any device sharing controllers and it can also determine if an EP RAD has a four byte interface. All such determinations are done during pre-configuration.

2.3.3 Declaring and Altering the Configurator

At completion of pre-configuration, a declaration of the current configuration is made via the TTY and the operator is allowed to add controllers not yet configured. He can also alter erroneously configured controllers (Sense Switch 3 must be set). In this way the operator can specify four byte interfaces which are not detected under program control.

The configuration input format is as follows:

MODEL NUMBER, CONTROLLER ADDRESS,

The model number must be four decimal digits followed by a comma. The controller address must be up to four hexadecimal digits, with leading zero suppression allowed followed by a comma. To terminate input, two commas must be input. The only acceptable model numbers are those listed in Paragraph 1.2. Once configuration input is completed, all controllers are entered into the control table and an IOP bandwidth analysis is performed. Any IOP whose bandwidth has been exceeded will be mentioned. The system will also state if the memories must be interleaved or not. Then the IOP table will be output. This is followed by the configuration table output. The last configurator output is the base device declaration to the teletype. If the operator wishes to elect another base device he may do so by altering the device address in register 13 (X'D'). Once the exerciser and operator agree on the system configuration, post-configuration can be started by typing RUN.

2.3.4 Post-configuration

Post-configuration verifies the base device selection. If it is not a rotating memory or if an incorrect address is given, an error message is issued and the system's choice will be reinstated and announced, again, subject to the operator's acceptance. Once the base device selection is accepted, the previously stated tables are output, and the system writes the exerciser's image (including the configurator) to the base device, sets aside another portion of the device's surface for an error history (Refer to Paragraph 2.5) and, then, initializes the exerciser and instructs the operator in how to start the exercise. From this point on, the base device becomes a re-bootable recovery device.

2.3.5 Configurator Messages

When the load has been completed the following message is output to the teletype:

```
SIGMA 5/6/7/8/9 SYSTEM EXERCISER (SEX)
PROGRAM NO. 705889-B03 (V238), MANUAL NO. 901737-B
CURRENT CONFIGURATION:
MODL ADDR
7445 0002
7140 0003
7160 0004
7201 00E0
7320 0280
7231 02E0
7201 02F0

R
```

The operator may input more information (described in section 2.3.3) or may terminate the input by typing two commas.

The following is a list of configurator error messages which are issued under the specified conditions:

(1) !!MODEL NUMBER ERROR. CHECK ELEMENT TABLE. The operator specified a model number which is not in the element table.

(2) !!LAST ENTRY HAS A DUPLICATE ADDRESS. The last entry contained an address which has already been specified either by the pre-configurator or a previous entry.

(3) !!XXXX NO RECOGNITION FOR CONTROLLER. CHANGE ENTRY. The system gets no address response for the address XXXX.

All three of these messages are followed by:

LAST ENTRY IGNORED.

2.3.5 Configurator Messages (Cont'd.)

The operator may continue without recalling or repeating the configuration. The following messages are output to the printer during configuration.

```
   SIGMA 7 CPU
   FLOATING POINT
   DECIMAL
   0002 REGISTER BLOCKS
   MAP
   WRITE LOCK
   0000 GROUP INT. LEVELS FBF0
   0002 GROUP INT. LEVELS FFF0
   0005 CANNOT CONFIGUR
   020C CANNOT CONFIGUR
   0200 IOP BANDWIDTH EXCEEDED
IOP TABLE
IOP DOC LST KWPS PCNT
000 000 008 0125 0044
001 000 000 0000 0000
002 000 004 0125 0148
SYSTEM CONFIGURATION TABLE
CTX DEV  UNIT MOD  MN PBW
001 02E0 8000 7231 EP 078
002 02F0 8000 7201 MS 040
003 00E0 8000 7201 MS 040
004 0280 8000 7320 9T 030
005 0002 8000 7445 PR 002
006 0003 8000 7140 CR 002
007 0004 8000 7160 CP 000
008 0005 8000 7165 CP 000
```

```
CONTROL TABLE
CTX DEV  UNIT OPS  CS   SEEK     BUFFER   COMS   MAPS   OP  CA   FC ETX REL
001 02E0 8000 A0BC 0000 00000000 000000.0 0028E4 002910 00 02E0 00 002 002
002 02F0 8000 A0BC 0000 00000000 000000.0 0028EA 002920 00 02F0 00 005 004
003 00E0 8000 A0BC 0000 00000000 000000.0 0028F0 002930 00 00E0 00 005 005
004 0280 8000 A0BC 0000 00000000 000000.0 0028F6 002940 00 0280 00 00C 001
005 0002 8000 8180 0000 00000000 000000.0 002900 000000 00 0002 00 010 006
006 0003 8000 8180 0000 00000000 000000.0 002902 000000 00 0003 00 013 007
007 0004 8000 8180 0000 00000000 000000.0 002904 000000 00 0004 00 017 008
008 0005 8000 8180 0000 00000000 000000.0 00290A 000000 00 0005 00 018 003

SYSTEM TABLE
BLKSIZ BUFSIZ FIRSTBUF STRTCORE ENDCORE  SM DS SO
000800 0007FC 000000.0 002950.0 018000.0 00 F7 00

STATUS TABLE
CS CI PS PI FS FI BON BOF RECL PASSTIME MSTART MEMSIZ  REGOPS
FF 00 FE 00 F0 00 005 055 0080 00000000 002950 018000 C0000000

OPERATOR TABLE
MSG  CH LOG  L H TTY  COC  MX CS CL CN BASE SORS
0002 80 0002 0 4 0001 0000 40 00 80 00 02F0 0000
```

2.4 Supervisor

2.4.1 General

The supervisor is the portion of the program which exercises and tests all of the system's hardware elements.

Each hardware element has a test module assigned to it which controls the element's activity. The test module is controlled by the supervisor. The test module provides the element with resources, in the form of data buffers, which can be transferred by the element to and from memory. Each individual test module can monitor its particular element for hardware detected faults while the CPUs test module is checking the data in the buffers for faults which the I/O test modules could not detect. The test modules perform the detection of and the recovery from errors. They also indicate the error reports which are filed and issued to the operator by the Error Auditor (Refer to Paragraph 2.5).

The operator may control an element test module by altering the corresponding contents of C:3 in the CONTROL TABLE. Use CTX of zero to control the CPUs test module.

2.4.2 Cycles

The supervisor's activity is partitioned into cycles. During each odd numbered cycle, the map, the external interrupts and the extra register blocks are exercised. During the even numbered cycles the map and the external interrupts are disabled and only register page zero is used. Testing the map is done by setting the map one to one and turning on the map bit for the data checker and the instruction tester (Refer to Paragraph 2.4.5.2) during the odd numbered cycles. Testing extra register pages is done by selecting a different existing register page address, each pass, during the odd numbered cycles. Testing the external interrupts is done by periodically toggling the arm and enable switch triggering the external interrupt levels during the odd numbered cycles. The first cycle is always even (zero).

2.4.3 Passes

Each cycle is partitioned into passes during which the amount of data transferred (the block size) by the devices remains constant. This simplifies the test modules problem of finding buffer space in memory and thereby increases the throughput of the systems data.

During each pass a different bi-directional controller and all of its devices is selected to restore the data in the unused buffers and provides a constant source of fresh data to be used by the system. This is a task usually delegated to the CPU in other exercisers. This device is referred to as the source device and frees the CPU so that it may perform other functions such as buffer checking. The operator may inhibit any controller from being used as a source device by resetting Bit 2 (X'2000') in its corresponding C:3 or OPS in the CONTROL TABLE (Refer to Paragraph 2.2.2.2.3).

There are seven automatic passes and one manual pass. The manual pass (pass 7) is the only pass in which the operator can select the block size (BLKSIZ) or S:0 of the SYSTEM TABLE. This value is fixed in the automatic passes.

Of the remaining passes, pass 0 through 6, each pass is devoted to the testing of the data transferred except for pass 5 which is discussed in Paragraph 2.4.5.2.2. Specific selection of passes can be done by modifying PS or S:10 of the STATUS (Refer to Paragraph 2.2.2.2.11). Any pass can be initiated by the operator via the START directives qualifier 1.

2.4.3.1 Pass 0

Pass 0 has a block size of 2048 bytes. All buffers are started on page boundaries and thus simulate in part much of the activity in our current operating systems. This pass provides the intermediate size block. Pass 0 is a data testing pass.

2.4.3.2 Pass 1

Pass 1 has a block size of 16 bytes. All buffers are started on a word boundary. This pass provides the small block size. Pass 1 is a data testing pass.

2.4.3.3 Pass 2

Pass 2 generates the maximum block size for the system. The block size is not to exceed 65536 bytes and is determined by dividing the exercisable memory space by the number of controllers selected at the time pass 2 is initiated. The first buffer will start on a byte boundary of one. The pass 2 block size will change from one pass 2 to another if the number of controllers selected is different. Pass 2 is a data testing pass.

2.4.3.4 Pass 3

Each time pass 3 is initiated, a different block size is randomly generated. The block size will lie between 16 and 2048 bytes. The first buffer will start on byte boundary 2. Pass 3 is a data testing pass.

2.4.3.5 Pass 4

Each time pass 4 is initiated a different block size is randomly generated. The block size will lie between 2048 and the maximum number of bytes. The first buffer will start on byte boundary 3. Pass 4 is a data testing pass.

2.4.3.6 Pass 5

Pass 5 has a block size of 48 bytes. The buffers all start on a doubleword boundary. Pass 5 is an instruction testing pass.

2.4.3.7 Pass 6

Pass 6 has a block size of 6144 bytes. All buffers start on a page boundary. All rotating memory surfaces are exercised during pass 6. All the other passes skip portions of the surface to minimize loss of latency. Pass 6 guarantees a loss of latency each transfer at the expense of fully testing the surfaces. Pass 6 is a data testing pass.

2.4.3.8 Pass 7

The block size is determined by the contents of BLKSIZ and BUFSIZ of the SYSTEM TABLE when Pass 7 is initialized; the operator may alter these values to his choosing and then start pass 7.

If the operator starts pass 7 without altering the SYSTEM TABLE, pass 7 will assume the characteristics of the most recently initiated pass except for passes 5 and 6. Pass 7 is a data checking pass.

Pass 5 is the only pass in which instruction testing can occur.

2.4.4 Phases

Each pass is subdivided into four phases. The phases control the initialization and the operating parameters of the test modules. The phases are numbered 0 through 3 and can be changed by operator request via the START directives Qualifier 2.

2.4.4.1 Phase 0

During Phase 0, the CPU initializes the data buffers and the newly elected source device is initialized and set to write only. This is done to establish known data patterns (Refer to Paragraph 2.4.5.2) on all the source controller's devices. There is no data testing during this phase.

2.4.4.2 Phase 1

During Phase 1, the source device is set to read only and all the bi-directional devices are initialized and set to write only. The CPU is now set to data checking. Since the source device is the only device being read it will be the only controller whose data transfer is being checked during this phase. The time duration of this phase is measured and is used to establish the time duration of the pass. The minimum is one minute.

2.4.4.3 Phase 2

When Phase 2 is started, the bi-directional devices are set to alternately read and write and their surfaces are accessed in a sequential manner. Loss of latency is kept to a minimum except for pass 6. This phase initializes the unidirectional devices. The duration of this phase is two times the duration of phase 1.

2.4.4.4 Phase 3

Phase 3 causes the bi-directional device surfaces to be accessed randomly with no latency considerations. Also, the rotating memories will periodically execute a CHECK WRITE operation in lieu of a read or a write. The duration of this phase is one half the duration of Phase 1.

2.4.5 Test Modules

The test modules initialize and sustain the operation of system elements. Error detection and recovery is also conducted by the test module. When the module detects an error, it reports the error to the Error Auditor (Refer to Paragraph 2.5), and the Auditor reports the error to the operator.

The test modules are controlled by the supervisor and the operator via the CONTROL table (Refer to Paragraph 2.2.2.2.3).

The operator may specify that a particular test module is to continuously repeat its current operation (BIT 1 of C:3 of the CONTROL TABLE). The operator may specify that all buffers read by a test module's hardware element are to be tested by the CPU prior to redistribution (Bit 14 of C:3).

All such control is available to the operator through the use of C:3 in the CONTROL table (Refer to Paragraph 2.2.2.2.3).

2.4.5.1 I/O Handler

Each controller configured into the system is assigned an I/O handler. The handler serves as the test module which controls the activity of all the devices on that controller. The handlers use the I/O interrupt to drive the devices so once a controller's activity is initiated, the handler will provide continuous activity for the controller and its devices. The handlers are designed to operate in parallel with one another and to drive all controllers simultaneously.

The activity for all handlers follows this basic format under normal operation with no error detected at step 1 (Refer to Paragraph 3.5, Flow Charts):

(1) Acknowledge interrupt, identify controller and check for error.

(2) Check position for bi-directional devices only.

(3) Give up data buffer.

(4) Generate next order (read, write, etc., for bi-directional devices only).

2.4.5.1 I/O Handler (Cont'd.)

(5) Generate next position (for bi-directional devices only).

(6) Get a new data buffer.

(7) Build new command string.

(8) Issue SIO, and set fault timer.

(9) Exit I/O interrupt.

If an error is detected, the following steps occur:

(1) Acknowledge interrupt, identify controller and acknowledge error.

(2) Identify and report error to the ERROR AUDITOR (Refer to Paragraph 2.5).

(3) Establish recovery by building recovery command string.

(4) Request supervisor retire controller for once clock tick and then attempt recovery.

(5) Exit I/O interrupt.

Two of the primary functions of the handlers are to generate position and to build command strings.

2.4.5.1.1 Positions and Maps

Positions apply only to the bi-directional devices. Tape positions refers to the number of the CURRENT record. This value appears in C:5 or SEEK of the corresponding CONTROL table entry. Positions for RADS and Packs refers to the SEEK address which also appears in C:5 of the CONTROL table (Refer to Paragraph 2.2.2.2.3).

The operator may control the selection of positions for the tapes by altering S:16 or RECL of the STATUS tables (Refer to Paragraph 2.2.2.2.10). This value specifies the maximum number of records which can be written to any tape.

The operator may control the selection of positions for the RADS and Packs by altering the unit map. The maps can be found by displaying the devices C:8 or MAPS entry in the CONTROL table. This is a word pointer to the map pairs for all the units on the controller. A particular unit's map pair is located at the contents of C:8 plus two times the unit number. The first word contains the first available sector to be exercised. This value is usually zero unless the unit is the base device. In that case, the value will be X'200'. The second word contains the last available sector + 1 to be exercised.

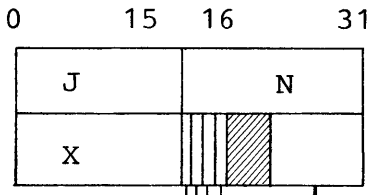
This value differs as follows:

HS RAD 7211	X'1480'	64 * 82 = 5248
EP RAD 7231	X'1800'	512 * 12 = 6144
MS RAD 7201	X'2000'	512 * 16 = 8192
DISC PACK 7240	X'FDC'	202 * 20 = 4040

The Pack is treated as having 202 tracks with 20 sectors of 6144 bytes each.

Therefore, the operator may specify the range of sectors he wishes to exercise on any RAD or Pack unit.

The tape maps are organized in pairs but contain different information. The following demonstrates the nomenclature of the tape maps:



J = Current record number

N = Maximum number of records

X = Used to time rewinds

SIOA = Unit is transferring data.

SIOP = Unit is pending a data transfer.

RWIA = Unit is rewinding.

RWIP = Unit is pending a rewind.

ORDER

1 = Write

2 = Read

C = Read backward

2.4.5.1.2 Command Strings and COMS

Each Controller has space allocated for its string of command pairs. C:7 or COMS is a word pointer to that space. All normal transfer operations are initiated by setting register 0 with the controller's corresponding C:7 contents divided by 2. Error recovery for mag tape and pack require the use of additional command pairs.

The following is a list of transfer command strings and error recovery strings for each of the various handlers. To simplify the explanation, the following conventions are followed:

BBBBB	Byte address of buffer + 4
SSSSS	Byte address of seek
NNNNN	Byte address of sense
0DDDD	Doubleword address of first command pair in string.
CCCC	Current buffer size (byte count)
1E	ICE, HTE, HUE, SIL flags are usually set

(1) Line Printers

010BBBBB+8	Print without format.
1E00CCCC-8	Does not output the header.

(2) Card Reader

0A0BBBBB	Read binary
1E00CCCC	

(3) Card Punch

090BBBBB	Punch binary and put error cards into alternate stacker.
2E00CCCC	CC-HTE-HUE-SIL
0800DDDD	Transfer in channel (TIC)
00000000	
00000000	Stop
1E000000	
090BBBBB 1E00CCCC	7165 only

2.4.5.1.2 Command Strings and COMS (Cont'd.)

(4) Mag Tape — In case of error SIO is issued from C:7+2. The error string is inserted only when an error occurs.

```
010BBBBB        Write
1E00CCCC

4B000000        Space record backward
24000000        CC-IUE
63000000        Set erase
24000000
0800DDDD        TIC
00000000

020BBBBB        Read forward
1E00CCCC

04000000        Sense - for recoverable read error
24000000
4B000000        Space record backward
24000000
03000000        Set correction
24000000
0800DDDD        TIC
00000000

0C0BBBBB+BUFSIZ-1    Read backward
1E00CCCC

43000000        Space record forward for non-
                correctable read error.
24000000
0800DDDD        TIC
00000000
```

(5) RADS & Packs — In case of error on the Packs, a restore carriage SIO is issued which does not appear in the command string.

```
030SSSSS        Seek
2E00000C
XX0BBBBB        XX = Write (01), read (12) or check
                write (05).
2E00CCCC
0400NNNN        Sense
1E00000C
```

2.4.5.2 Buffer Tester

The buffer tester initializes the data buffers during phase zero and checks the data in the buffers thereafter. The buffer tester activity is similar to that of the I/O handlers in that it must request, operate on and return data buffers to the data buffer pool.

Buffer testing is performed in two ways:

(1) Test the contents of the buffer for known data patterns. Such buffers are called Data Buffers.

(2) Use the contents of the buffer to test some of the instruction repertoire. Such buffers are called Instruction Buffers.

The buffer tester is considered the CPUs test module. The operator may control the buffer tester via the CONTROL table. The CPUs CTX is zero. Refer to Paragraph 2.2.2.2.3 for CONTROL table definitions.

2.4.5.2.1 Data Buffers

A data buffer is a portion of memory for use by a test module for all passes except pass 5. These buffers are byte oriented. The available memory is divided into sequential buffers of equal size whose boundaries may start on any byte. The SYSTEM table contains the information concerning the buffers to be generated during any pass. Refer to Paragraph 2.2.2.2.11.

Each buffer consists of three parts, a header (Refer to Figure 1), a data area and a residual area (Refer to Figure 2).

The header consists of 12 bytes which identifies the last user of the buffer and the source and destinations of the buffer's data.

The data area consists of two blocks of identical data. The number of bytes in a data block is divisible by four and can therefore be thought of in terms of words. This structure allows buffer checking by comparing the two identical blocks on a word basis even though the data was generated on a byte basis. The number of words in a data block can be expressed as the quotient of: $\frac{BLKSIZ-12}{8}$.

The remainder of the expression can be from 0 to 7 bytes and is the number of RESIDUAL buffer bytes.

2.4.5.2.1 Data Buffers (Cont'd.)

During phase 0 of each pass, memory is cleared and then subdivided into buffers. Patterns are generated in the buffers and written to the surfaces of the bi-directional devices. Each buffer will have a different pattern thus creating a random mix of surface data throughout the system.

Each byte in the data areas can be expressed as follows:

$$B_{i+1}=B_i+M_i$$

$$M_{i+1}=M_i+C$$

B_i is the ith byte with i equal zero for the first byte. M_i is the difference between ith byte (B_i) and the ith minus one byte (B_{i-1}). The modifiers are the first order difference of the bytes. C is a constant and remains unchanged for a given buffer. C is the difference between the ith modifier (M_i) and the ith minus one modifier (M_{i-1}). The constant is the second order difference of the bytes.

By examining the first three bytes of a data area the system and/or the operator can derive the constant and any modifier or byte. There are three types of data generated:

(1) Fixed, the modifier and constant equal zero, M=0, C=0.

(2) Sequential, the modifier equals one and constant equals zero M=1, C=0.

(3) Random, the constant equals one $M=M_i$, C=1.

During a pass, all three types are generated. The operator can control the selection of buffer types through S:6 or DS of the SYSTEM table (Refer to Paragraph 2.2.2.2.11).

Checking the buffers is done by summing up the two identical portions and comparing the sums to one another. Sigma 6, 7, 8 and 9, which use the CVA instruction, can sum-check buffers twice as fast as the high speed RAD can generate them. The Sigma 5 which has no CVA instruction, operates at about one third the speed of the Sigma 6, 7, 8, and 9. Checking the residual bytes is done by regenerating the bytes and comparing. The residual bytes are generated as an extension of the second data block.

2.4.5.2.1 Data Buffers (Cont'd.)

It should be noted that a BLKSIZ of less than 16 bytes (possible only in pass 7 by operator selection) reduces the header from 12 to 5 bytes. The A2 - A4 and P1 through P4 are replaced by data bytes. The buffer is treated as 5 header bytes and the rest of the buffer is all residual. The minimum BLKSIZ under these circumstances is 9 bytes, five for the header and 4 data. There must be at least 4 data bytes to check the data, three to determine C and M and 1 byte to check.

When a data error is found in a data buffer, the error is reported as a DATA ERROR (Refer to Paragraph 2.5.3.15).

2.4.5.2.2 Instruction Buffer

An instruction buffer is a portion of memory for use by the test module during pass 5 only. These buffers are word oriented and fall on doubleword boundaries.

Each buffer contains an instruction and randomly generated data which is used to test the instruction's execution. The floating point, decimal, byte string, and fixed point multiply and divide instructions are tested in this fashion.

These instructions are tested in two ways:

(1) Use complementary instruction pairs on data which is randomly generated. The data is generated during Phase 0. The test is performed by executing the primary and complementary instructions and comparing the results during the other phases.

(2) Simulate the operation of an instruction using randomly generated data and store the results in the buffer during Phase 0. The rest of the time the instruction is executed and the results compared against the predetermined results.

Each buffer consists of a header (Refer to Figure 1) an instruction descriptor and a data block (Refer to Figure 3).

The header consists of 3 words (12 bytes) which identifies the last user of the buffer and the source and destinations of the buffer's data. The instruction descriptor consists of two halfwords (4 bytes) which identify the instruction pair for which the buffer's contents were generated. The first halfword contains the primary instruction and register used. The second halfword contains the complementary instruction and register or all zeros if only one instruction is to be tested. The data block consists of eight words of psuedo randomly generated data to be used to test the instruction(s) specified by the instruction descriptors.

FIGURE 1. BUFFER HEADER

BS Buffer Status Byte

CTX Control table index of buffer's last user.

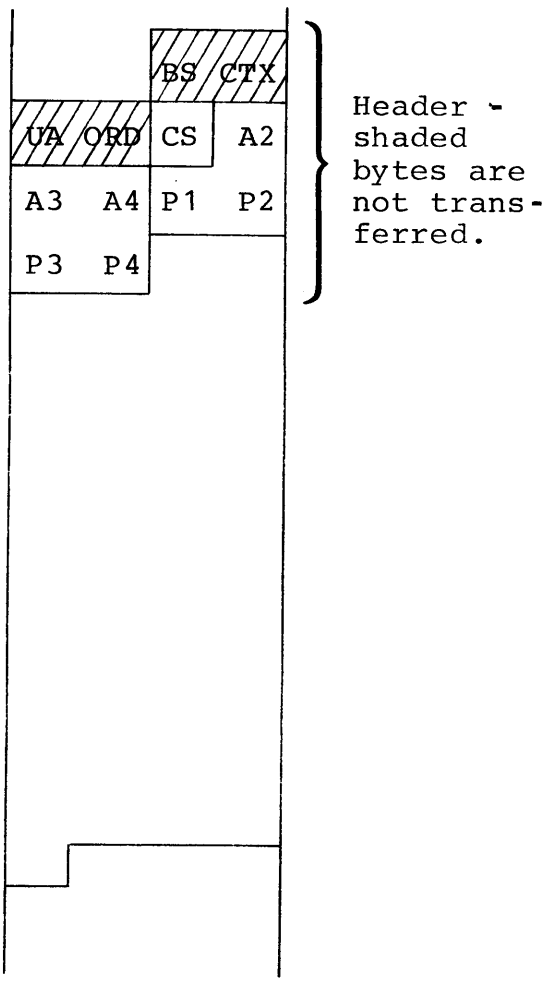
UA Unit address of buffer's last user.

ORD Order (read, write, etc.) of buffer's last user.

CS Card size - equals zero unless user is card equipment. Specifies number of bytes on card.

A2-A4 Byte address of buffer's last transfer from memory to device surface.

P1-P4 Seek address of device surface.



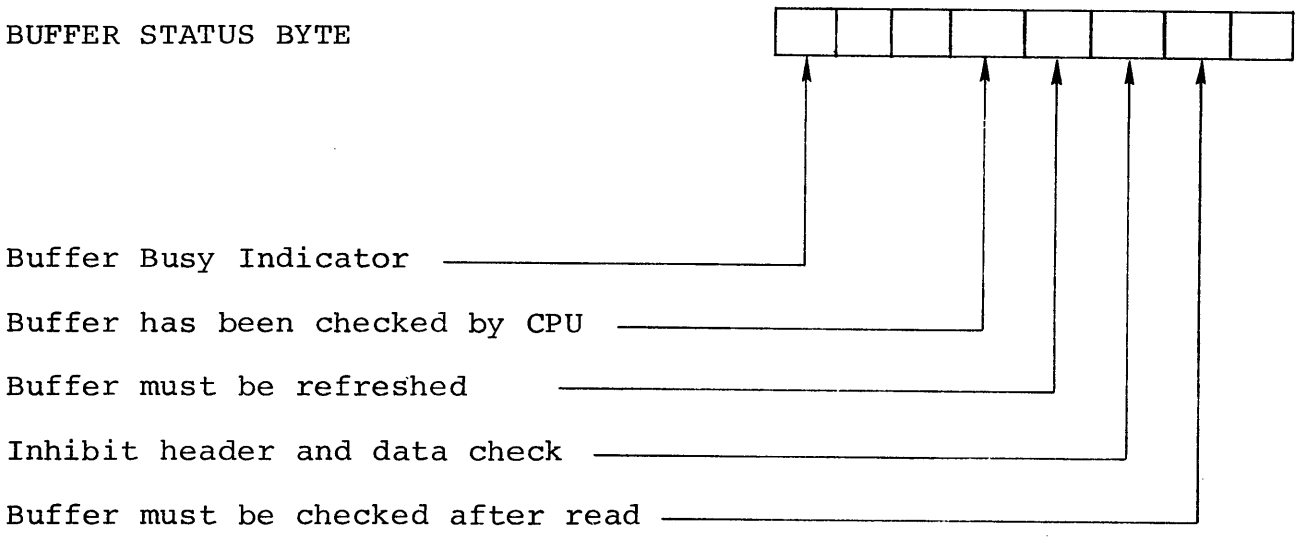
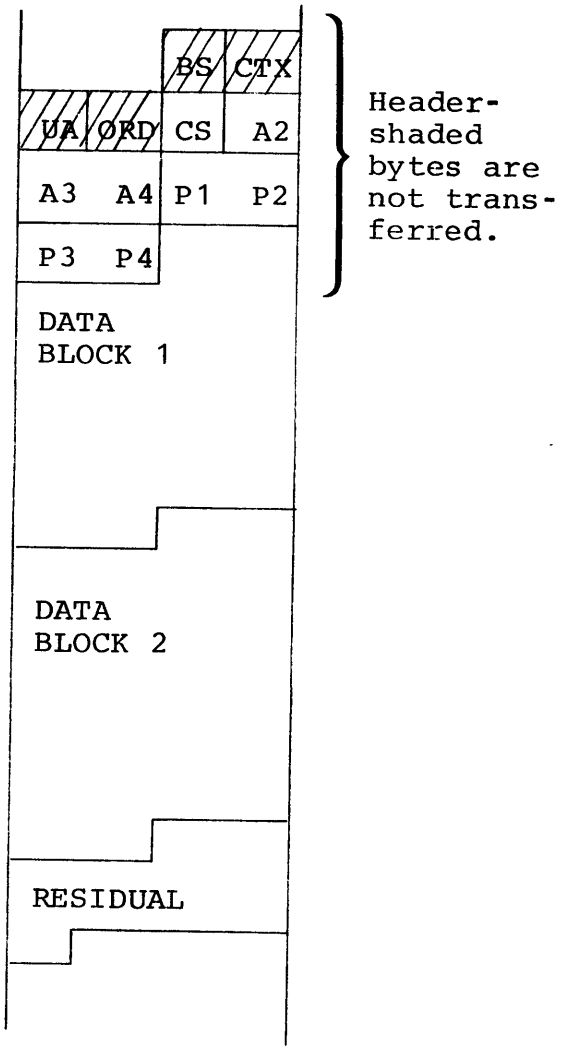


FIGURE 2. DATA BUFFER

DATA BLOCK 1 = First Data Block

DATA BLOCK 2 = Second Data Block. Identically equal to BLOCK 1.

RESIDUAL = Those data bytes which are not a part of BLOCK 1 or 2.



No. of words in a Data Block:

$$\left[\frac{\text{BLKSIZ} - 12}{8}\right]_{Q}$$

No. of bytes in a Data Block:

$$4^{*}\left[\frac{\text{BLKSIZ} - 12}{8}\right]_{Q}$$

No. of bytes of Residual:

$$\left[\frac{\text{BLKSIZ} - 12}{8}\right]_{R}$$

2.4.5.2.2 Instruction Buffer (Cont'd.)

During Phase 0 of pass 5, memory is cleared and subdivided into buffers of twelve words each. Each buffer is given a different instruction descriptor and a set of randomly generated data which corresponds to the instruction(s) specified by the instruction descriptor. The selection of instruction descriptors can be controlled by the operator through S:6 or DS of the SYSTEM TABLE (Refer to Paragraph 2.2.2.2.11). Furthermore, the operator may specify that a specific instruction is to be tested by setting S:7 or SO of the SYSTEM TABLE.

The following is a list of instruction descriptors and a description of how the buffer is generated during Phase 0 and tested during the remaining phases.

It should be noted that if an error is detected by the Buffer Tester during this pass, it is reported as an instruction error (Refer to Paragraph 2.5.3.16).

(1) MW,12/DW,12 (Refer to figure 3A)

Generate: Set R12 to zero with random data in R13 and M (memory).

Test: Load R2 with word address of M. Load R12,R13 from initial register portion of buffer. Execute (MW,12 0,2) and then (DW,12 0,2). Compare results in R12,R13 with initial R12,R13.

(2) MH,13/DH,13 (Refer to figure 3A)

Generate: Set R12 to zero. Put random data in second halfword of R13. Put random data in first halfword of M.

Test: Load R3 with word address of M. Load R12,R13 from initial register portion of buffer. Execute (MH,13 *3) and then (DH,13 *3). Compare results in second halfword of R13 with initial R13.

(3) SF,12 (Refer to figure 3A)

Generate: Set R13 to zero with random data in R12 with exponent less than 64. Generate random shift right value of 1 to 7 and put into M.

Test: Load R4 with M and execute (SF,12 *4). Load R4 with two's complement of M and execute (SF,12 X'100',4). Compare results in R12,R13 with initial R12,R13.

2.4.5.2.2 Instruction Buffer (Cont'd.)

(4) FAL,12/FSL,12 and FML,12/FDL,12 (Refer to figure 3A)

Generate: Put random normalized data in R12,R13, and M,M+1. Force the exponent to be equal and lie between X'21' and X'5E'.

Test: Load R3 with word address of M. Load R12,R13 from initial register portion of buffer. Execute (FAL,12 *3) or (FML,12 *3) then (FSL,12 *3) or (FDL,12 *3). Compare results in R12,R13 with initial R12,R13 for difference of 15 or less.

(5) FAS,12/FSS,12 and FMS,12/FDS,12 (Refer to figure 3A)

Generate: Put random normalized data in R12 and M and set R13 to zero. Force exponents to be equal and lie between X'21' and X'5E'.

Test: Load R3 with word address of M. Load R12,R13 from initial register portion of buffer. Execute (FAS,12 0,3) or (FMS,12 0,3) then (FSS,12 0,3) or (FDS,12 0,3). Compare results in R12 with initial R12 for difference of 15 or less.

(6) UNPK,8/PACK,8 (Refer to figure 3B)

Generate: Set R12,R13 to zero and put 15 random decimal digits plus sign in R14,R15 and set M through M+3 to zero.

Test: Clear contents of LOC through LOC+3 to zero. Load R12 through R15 with initial decimal registers in buffer. Execute (UNPK,8 LOC) then (PACK,8 LOC). Compare results in R14,R15 with initial R14,R15.

(7) DA,0/DS,0 (Refer to figure 3B)

Generate: Set R12 through R15 and M through M+3 each with 31 random decimal digits plus sign forcing most significant digits of R12 and M to have a value of less than 5 to avoid overflow.

Test: Load R3 with word address of M. Load R12 through R15 with initial decimal register in buffer. Execute (DA,0 *3) then (DS,0 *3). Compare results in R12 through R15 with initial R12 through R15.

2.4.5.2.2 Instruction Buffer (Cont'd.)

(8) DM,8/DD,8 (Refer to figure 3B)

Generate: Set R12,R13 to zero and put 15 random decimal digits plus sign into R14,R15 and another 15 random decimal digits plus sign into M+2, M+3. Set M,M+1 to zero.

Test: Load R3 with word addresses of M. Load R12 through R15 with initial decimal registers in buffer. Execute (DM,8 *3) then (DD,8 *3). Compare results in R12 through R15 with initial R12 through R15 excluding sign in R13.

(9) DSA,0 (Refer to figure 3B)

Generate: Set R12,R13 and M+1, M+2, M+3 to zero. Put random decimal digits plus sign into R14,R15. Set a random shift value not to exceed 16 in M.

Test: Load R12 through R15 with initial decimal registers in buffer. Load R4 with contents of M and execute (DSA,0 *4). Then load R4 with two's complement of M and execute (DSA,0 *4). Compare results in R12 through R15 with initial R12 through R15.

(10) DL,0/DC,0 (Refer to figure 3B)

Generate: Set R12 through R15 to zero. Put 31 random decimal digits plus sign into M,M+1, M+2, M+3.

Test: Load R3 with word address of M. Load R12 through R15 with initial decimal registers in buffer. Execute (DL,0 *3) then (DC,0 *3). The DC,0 instruction compares R12 through R15 with M through M+3.

(11) DST,0/DC,0 (Refer to figure 3B)

Generate: Set R12 through R15 with 31 random decimal digits plus sign. Set M through M+3 with zeros.

Test: Set LOC through LOC+3 with zeros. Load R12 through R15 with initial decimal registers in buffer. Execute (DST,0 LOC) then execute (DC,0 LOC). The DC,0 instruction compares R12 through R15 with the contents of LOC through LOC + 3.

2.4.5.2.2 Instruction Buffer (Cont'd.)

(12) MBS,12/CBS,10 (Refer to figure 3C)

Generate: Set R12 to zero. Set R13 with a random count from 1 to 7 in Byte 0 and the byte address of LOC. LOC is to be the destination of the test. Set R14,R15 with 9 randomly generated bytes. Set M,M+1 with 8 randomly generated bytes and set M+2, M+3 to zero.

Test: Set R12 with byte address of initial source portion of buffer. Set R12 with initial register from buffer. Set R10,R11 with contents of R12,R13. Execute (MBS,12 0) then execute (CBS,10 0). The CBS,10 compares the source with the destination. The contents of R10,R11 are compared with R12,R13.

(13) EBS,12 (Refer to figure 3C)

Generate: Set R12 and M to zero. Set M+2, M+3 to X'23232323', X'23232323'. Generate 8 random decimal digits and put into M+1. Simulate the EBS, generate a predetermined resultant and put into R14,R15. Put resulting register into R12,R13.

Test: Pick up initial destination from buffer and put into LOC, LOC+1. Load byte 0 of R12 and R15 from initial registers in buffer. Merge byte address of initial source plus one into R12. Execute (EBS,12 0). Compare LOC, LOC+1 with expected results in buffer.

(14) TTBS,12 (Refer to figure 3C)

Generate: Generate initial source, destination and initial registers contents. Simulate the TTBS and put the resulting register contents of R12,R13 into the expected results in the buffers.

Test: Pick up initial destination from buffer and move to LOC, LOC+1. Load register R12,R13 from initial registers in buffer. Merge byte address of buffers initial source location into R12. Execute (TTBS,12 0). Compare the results in R12,R13 with the expected results in the buffer.

2.4.5.2.2 Instruction Buffer (Cont'd.)

(15) TBS,12 (Refer to figure 3C)

Generate: Generate initial source, destination and initial register contents, simulate the TBS and put the resulting destination into the expected results in the buffer.

Test: Pick up initial destination from buffer and move to LOC, LOC+1. Load register R12,R13 from initial registers in buffer. Merge byte address of buffers initial source location into R12. Execute (TBS,12 0). Compare the results in LOC,LOC+1 with expected results in the buffer.

2.5 Auditor

2.5.1 General

The Auditor manages the error history file. It is responsible for accepting error reports from the test modules, entering the report onto the error history file and issuing the report to the operator. The Auditor files the error entry into a 256 word error buffer and then outputs the entry to the operator. When the error buffer is full, it is appended to the error history file on the base device.

All system restarts cause a reconstruction of the error history file. This minimizes the loss of error history regardless of the severity of the crash. Real time is also kept in the error history file and thereby minimizes the loss of relative time due to system restarts or rebooting.

2.5.2 Severity Levels of L and H

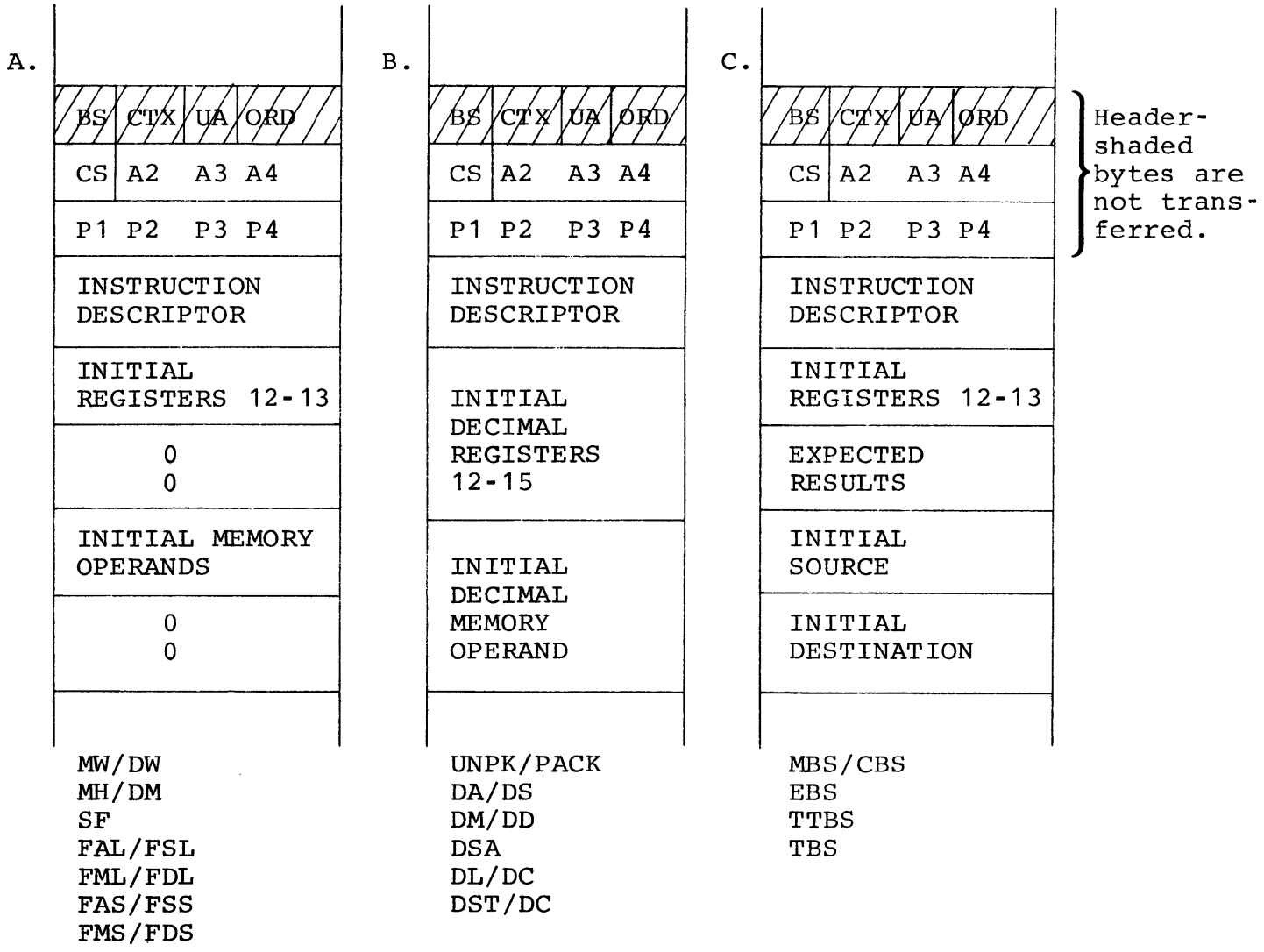
There is a severity level associated to each error type. The operator may select which levels of severity to be reported and which levels should cause the system to halt by altering L or H in the OPERATOR table (Refer to Paragraph 2.2.2.2.7.).

The system default is zero for L and 4 for H. This causes all errors with a severity level of zero or greater to be reported to the operator on the LOG device and all severity levels of 4 or higher to cause the system to halt after the error is reported. L must always be less than or equal to H or the system would halt without an observable reason.

The following is a list of the severity levels, corresponding error types, and the subsequent error report.

FIGURE 3. INSTRUCTION BUFFERS

INSTRUCTION BUFFERS with associated instructions listed below.



2.5.2.1 Severity 0

A. Recoverable I/O errors

(1) Disc Pack flaw mark

(2) Mag tape reads or writes

2.5.2.2 Severity 1

A. New Pass

B. New Phase

C. Software timeout

(1) Mag tape rewind

D. Unidentified trap

(1) Fixed point arithmetic trap (43)

(2) Floating point fault (44)

(3) Decimal fault (45)

(4) CAL 1 (48)

(5) CAL 2 (49)

(6) CAL 3 (4A)

(7) Counter 1 interrupt (58)

(8) Counter 2 interrupt (59)

(9) Counter 3 interrupt (5A)

E. Unidentified I/O interrupt

(1) Device address not in C:1 of CONTROL table.

2.5.2.3 Severity 2

A. Unidentified I/O interrupt

(1) No address recognition at the AIO

2.5.2.4 Severity 3

A. Recoverable I/O error

(1) Data overrun

(2) Transmission data error

(3) Unusual end with no other error flag

2.5.2.5 Severity 4

A. Stack limit fault (42)

2.5.2.6 Severity 5

A. Software timeout

B. SIO failure

2.5.2.7 Severity 8

A. CPU memory parity (56 Sigma 5,6, and 7)

B. I/O memory parity

(1) Transmission memory error

C. I/O unrecoverable error

(1) Disc Pack header verification or header parity error.

(2) Memory address error

(3) IOP memory error

(4) IOP control error

D. Processor fault (56 Sigma 8 and 9)

E. Data error

F. Instruction error

2.5.2.8 Severity A

A. Position error

2.5.2.9 Severity E

A. Memory protect violation (40-1)

B. Privileged instruction access (40-2)

C. Nonexistent memory access (40-3)

D. Nonexistent instruction access (40-4)

E. Unidentified trap (40)

(1) Trap 40 with none or multiple condition code settings.

2.5.2.9 Severity E (Cont'd.)

F. Unimplemented instruction trap (41)

G. Watchdog timer runout

(1) DIO bus hang up (46-1)

(2) Memory bus hang up (46-2)

(3) Processor bus hang up (46-4)

(4) Instruction hang up (46-0 or 8)

H. Instruction exception trap (4D Sigma 8 and 9)

(1) Invalid register destination - master mode (4D-0)

(2) Invalid register destination - slave mode (4D-1)

(3) Illegal MMC configuration (4D-2)

(4) Nonexistent register block (4D-8)

(5) Intrap instruction violation (4D-C)

(6) PDF violation (4D-F)

I. CPU memory parity trap (4C - Sigma 8 and 9)

J. CPU memory parity interrupt (57 Sigma 8 and 9)

K. Double PDF fault

L. Power failure (50, 51)

2.5.2.10 Severity F

A. System return stack fault

2.5.3 Error Report

All error reports consist of three lines of output. The first line is an identification message which specifies the type of error. The second line contains the error heading and the third line the error data. These three lines are referred to as a basic error report.

In addition, some error reports include a listing of the busy devices at the time of the error. This listing consists of a heading line and one data line for each busy device.

2.5.3 Error Report (Cont'd.)

Some error reports may also include an extended status report This report consists of a heading line followed by two lines of eight words each. The following is an example of an error report which includes the basic error report, the busy device listing and the extended status report. For a complete explanation of this report refer to Paragraph 2.5.3.23.

```
G
--WATCHDOG TIMER RUN OUT
 SEQ   IDENT HR MN SC   PSW1     PSW2    STATUS 0 STATUS 1 STATUS 2 STATUS 3
00034 E4640 01 54 33 0000018D 07000008 4F040000 00000004 00140501 80000000
CTX DEV   TIOR    TIORUl  OP  BUF    SEEK    SENSE
01 02F0 00001491 90800000 0103176E 08380000 083A3A00
02 01F0 00001494 10800000 1203117A 1E600000 1E620200
03 00F2 00001497 10800000 1203411A 00380E00 00380E02
04 0080 0000149A 10800000 01031D62 00180200 00180202
05 0180 0000149B 10000000 01030B86 000000CC 00000000
EXTENDED STATUS.
00140501 80E12C00 80000000 A8000000 00009000 00008000 00008000 0000C058
0840C45E 41A01480 19818860 00000000 00000000 00000000 00000000 00000000
```

Each error report is given a sequence number which appears under ''SEQ'' in hex.

The ''IDENT'' is a five digit hex number which is in the form STTCC. Where ''S'' is the severity, ''TT'' is the trap or interrupt which detected the error and ''CC'' is a sub-error code or the condition code. The busy device list gives the CTX of the specific device. The information under the heading ''OP BUF'' is the transfer command. It may not be the command pointed at by the ''TIOR''. The ''SEEK'' is the most recent seek value. The ''SENSE'' is the result of the last successful sense so it may not coincide with the ''SEEK''. Only in the case of the errored device is the ''SENSE'' guaranteed to be in step with the ''SEEK''.

2.5.3.1 New Pass

Each time a new pass is started.

```
--NEW PASS
 SEQ   IDENT HR MN SC CI PI SORS BLKSIZ BUFSIZ FIRSTBUF
00005 11000 00 05 19 00 05 02F0 000030 000020 002A00.0
```

CI = Current cycle indicator 00 through FF.

PI = Current pass indicator 00 through FF.

SORS = Current source device.

2.5.3.2 New Phase

Each time a new phase is started.

```
G
--NEW PHASE
 SEQ  IDENT HR MN SC
00003 11010 00 01 03
```

2.5.3.3 Unidentified Trap

Location X'40' through X'13F' contains XPSDs which are entries to the trap and interrupt handlers. Most of these handlers are inactive because the system does not have corresponding hardware. When a trap or interrupt occurs which enters an inactive handler, an unidentified trap report is issued.

The handler reports the occurrence to the Auditor and then returns control to the location specified by the PSW1.

```
G
--UNIDENTIFIED TRAP
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST
00004 E0000 00 05 12 9D000D5F 00000000 A9862433
```

2.5.3.4 Memory Protect Violation

The CPU attempted to access a page whose write locks or access protection under the map was violated.

After the report is issued, the program resumes execution at PSW1.

```
G
--MEMORY PROTECT VIOLATION
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST
00003 E4010 00 30 42 10000C4A 20001400 F5AC26DC
```

2.5.3.5 Privileged Instruction Access

The exerciser has accessed a privileged instruction. After the report, the program resumes execution at PSW1.

```
G
--PRIVILIDGED INSTRUCTION ACCESS
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST
0004C E4020 00 29 24 10800C53 00000000 0F000BDA
```

2.5.3.6 Nonexistent Memory Access

The CPU attempted to access out of bounds memory. After the report, the program resumes execution at PSW1.

```
G
--NON-EXISTANT MEMORY ACCESS
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST
00015 E4040 00 14 59 90000B8F 07000110 E8002313
```

2.5.3.7 Nonexistent Instruction Access

The CPU attempted to access an undefined instruction. After the report, the program resumes execution at PSW1.

```
G
--NON-EXISTANT INSTRUCTION ACCESS
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST
00032 E408C 00 19 21 00000026 00000000 00000000
```

2.5.3.8 Unimplemented Instruction Access

The CPU attempted to access an unimplemented instruction. For example, accessing a decimal instruction when there is no decimal unit. After the report, the program resumes execution at PSW1.

```
G
--UNIMPLEMENTED INSTRUCTION ACCESS
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST
00005 E4100 00 31 02 4000DED5 00000000 7740000A
```

2.5.3.9 Stack Limit Fault

A stack limit trap has occurred. After the report, the program resumes execution at PSW1.

```
G
--STACK LIMIT FAULT
 SEQ  IDENT HR MN SC   PSW1     PSW2     INST     STACK1   STACK2
00048 44200 00 25 56 00000101 00000000 0B000018 00000000 00000000
```

2.5.3.10 Recoverable I/O Error

The device interrupted and unusual end is detected at the AIO and TIO odd register status bits 10 through 13 are reset. The handler recovers by repositioning and then retrying the operation. The following example is a data overrun.

```
--RECOVERABLE I/O ERROR
 SEQ  IDENT HR MN SC   AIOR     TDVRU1    TIOR     TIORU1    COM1      COM2
00013 05C40 00 05 38 88D80181 C8C20008 00001402 18C20008 02037804 1C0007FC
CTX DEV   TIOR     TIORU1   OP  BUF     SEEK     SENSE
01 02F0 00000000 00000000 01029000 0FBA0000 0FB83800
02 01F0 000013FA 760007F0 01028800 1D260000 10250500
03 00F3 000013F0 760007F0 1202E000 000F1200 000E1002
04 0080 00001400 760007F0 01029800 00150000 00141202
05 0181 00001402 18C20008 02037800 00000078 00000000
06 0102 00001408 760002F4 0102A000 00000000 00000000
07 0002 00001409 76000000 0103D800 00000000 00000000
09 0004 0000140B 76000000 09049800 00000000 00000000
```

The following is a transmission data error.

```
--RECOVERABLE I/O ERROR
 SEQ  IDENT HR MN SC   AIOR     TDVRU1    TIOR     TIORU1    COM1      COM2
00028 05C40 00 25 40 00580080 40420000 0000149D 18420000 01039804 1C0007FC
CTX DEV   TIOR     TIORU1   OP  BUF     SEEK     SENSE
01 01F0 00001495 760107FC 1202F800 08AA0000 08A82800
02 02F2 00001498 760002B4 12030800 11630000 11620202
03 0380 0000149B 760007EC 01031800 00651000 00650E02
04 0080 0000149D 18420000 01039800 0000000A 00000000
05 00E0 000014A3 760004AC 01038800 00000006 00000000
```

2.5.3.11 Unexpected I/O Interrupt

This occurs when there is no recognition to an AIO response after an I/O interrupt has occurred (Refer to Paragraph 2.5.2.2) or when the device which responds to the AIO cannot be found in the CONTROL table (Refer to Paragraph 2.5.2.3.).

```
--UNEXPECTED I/O INTERRUPT
 SEQ  IDENT HR MN SC   AIOR     TDVRU1    TIOR     TIORU1    COM1      COM2
00022 25C10 00 34 29 00001FFF 00000000 00001028 000001F0 404040E2 C5C5D240
CTX DEV   TIOR     TIORU1   OP  BUF     SEEK     SENSE
01 03F0 0000145F 10000000 1203FF90 03B80000 03B93900
02 01F0 20001461 1000000C 0103FFC0 00000000 05C30300
```

AIOR contains X'1FFF' when there was no address recognition to the AIO (severity = 2) otherwise the address was not found in C:1 of the CONTROL table (severity = 1).

2.5.3.12 Software Timeout

All I/O transfers are allowed from 2 to 20 seconds to complete depending on the device. If a device does not interrupt in that period of time, a software timeout occurs and the device is halted with an HIO.

The device status will appear in two places in the error output. The status reported on the first line is two seconds later in time than the status which appears in the busy device list. The handler recovers by retrying the operation without repositioning. This may result in position errors (Refer to Paragraph 2.5.3.17).

```
--SOFT-WARE TIME OUT
 SEQ  IDENT HR MN SC   AIOR     TDVRU1    TIOR     TIORU1    COM1     COM2
00049 45B00 00 36 16 000002F1 00800000 00000009 10800000 01166740 0A00002C
CTX DEV   TIOR     TIORU1   OP  BUF    SEEK     SENSE
01 01F0 00001495 7601002C 01040F50 109B0000 10991900
02 02F1 00000009 10800000 1204BC30 0F240000 13390902
03 0381 0000149C 90800000 01040F20 00BF1200 00BF1201
04 0081 0000149D 7600002C 0C040E60 0000001C 00000000
05 00E0 000014A3 7600002C 02040E30 00000096 00000000
```

AIOR contains the device address and is not the result of an AIO.

2.5.3.13 SIO Failure

When the SIO to transfer data is not accepted because the device is busy, the handler recovers by retrying the operation without repositioning.

```
G
--SIO FAILURE
 SEQ  IDENT HR MN SC   AIOR     TDVRU1    TIOR     TIORU1    COM1     COM2
0003E 45C20 00 14 42 00000080 04800000 00001499 90800000 0400A20C 1E000004
CTX DEV   TIOR     TIORU1   OP  BUF    SEEK     SENSE
01 02F0 00001490 90800000 12060A50 07C20000 07C34300
02 01F0 00001493 90800000 01060A20 17E60000 17E70700
03 00F2 00001496 90800000 01060A70 00331200 00331201
04 0080 00001499 90800000 12060A10 00330C00 00330C01
05 0181 0000149A 90000000 02060A30 000000FF 00000000
```

2.5.3.14 Unrecoverable I/O Error

This report is issued as the result of:

(1) Memory address error bit 11 in TIORU1.

(2) IOP memory error bit 12 in TIORU1.

(3) IOP control error bit 13 in TIORU1.

(4) Disc Pack header verification bit 1 in TDVRU1.

The handler recovers in the following manner:

(1) If the controller has not had all of its surface keyed, the command string is rebuilt but the same position and buffer are used.

(2) If the controller has been keyed, the command string is rebuilt and a new position and buffer are used.

```
G
--UNRECOVERABLE I/O ERROR
 SEQ   IDENT HR MN SC   AIOR     TDVRU1     TIOR     TIORU1     COM1      COM2
00006 85CA0 00 06 52 00080380 6EC40000 0000149A 18000000 0300A068 2C000004
CTX DEV   TIOR     TIORU1  OP  BUF    SEEK    SENSE
01 01F0 00000000 00000000 0106E800 094C0000 094E4E00
02 02F0 00000009 10800000 01021800 01F60000 02520200
03 0380 0000149A 18000000 0106F000 00050C00 80000000
04 0080 0000149D 76000494 01072800 00000021 00000000
05 00E0 000014A3 760001EC 02079000 00000001 00000000
```

2.5.3.15 Data Error

During all passes except pass 5, the CPU is used to check the buffer data. Paragraph 2.4.5.2.1 explains how the buffers are organized and how they are tested.

This error report is issued when the sum of the two data blocks in the buffer are not equal or when the residual is incorrect. When either of the above happens, a routine is entered that systematically tests each byte in the buffer and identifies the first failure and the number of subsequent failures. After the report, the buffer status byte is set to ''buffer must be refreshed'' (bit 4).

```
G
--DATA ERROR
 SEQ   IDENT HR MN SC  CTX DEV  BUFADD   ERRADD  ISSBMIC0 COUNT POSITION
00005 88000 00 02 57 000202F0 012800.0 012891.0 01E90000 00003 01F60000
```

2.5.3.15 Data Error (Cont'd.)

CTX = Control table index of buffer's last user.

DEV = Device address of buffer's last user.

ERRADD = Byte address of first erroneous data byte.

IS = What the error byte is.

SB = What the error byte should be.

MI = First modifier or buffer.

C0 = Buffers constant.

COUNT = Number of errors found in buffer.

POSITION = Last surface position the buffer was transferred to or from.

2.5.3.16 Instruction Error

An instruction error report can only occur during pass 5 under the conditions specified in Paragraph 2.4.5.2.2.

The error report uses the extended status in the following manner as a function of the instruction descriptor in the report.

(1) The first line which is the first eight words contains the buffer's data block regardless of the instruction descriptor.

(2) The second line first four words are the register results of the complementary instruction. The second four words are the results of the primary instruction. This applies to all instruction descriptors except for the byte instruction.

(3) The second line for the EBS (63) and the TBS(41): the first 2 words are the register results. The second 2 words are the destination results. The third 2 words are the initial source and the last 2 words are the initial destination.

(4) The second line for the TTBS(40) is the same as the other byte string instructions except the second 2 words are the register results (same as first 2 words).

2.5.3.16 Instruction Error (Cont'd.)

(5) The second line for the MBS/CBS(60/61) is the same as the other byte string instructions except the last two words are the original contents of the registers.

```
G
--INSTRUCTION ERROR
 SEQ  IDENT HR MN SC IN1 IN2   BUFADD
00012 88010 00 29 35 79007800 004B00.0
EXTENDED STATUS.
02938841 60940385 38182901 3535310D 05522318 50145408 35871230 39393200
82938841 60940385 38182901 3535310D 08461160 11085793 74054131 7474630D
```

IN1 = Primary instruction

IN2 = Complementary instruction

2.5.3.17 Position Error

All the multi-unit controller's test modules check the buffer header after a read is completed. P1 through P4 is compared with the appropriate controller's C:5 (SEEK). When they are not equal, a position error report is issued.

When the operator types RUN, the same operation is retried. If it fails again, the position is incremented by the proper amount, according to the pass, and the next position is tried.

The operator may override the position error output by setting Bit 1 (X'40') in SM of the SYSTEM table.

```
G
--POSITION ERROR
 SEQ  IDENT HR MN SC  CTX DFV   SEEK     SENSE    POSITION  BUFFER
00047 A5C00 00 14 46 00050180 000000D9 00000000 000000D5 009200.0
CTX DEV   TIOR     TIORU1   OP  BUF     SEEK     SENSE
01 02F0 00000000 00000000 0101C800 16C60000 16C44400
02 01F0 000013FA 76000550 1201E000 08B30000 08B50500
03 00F3 000013FD 760007FC 01020800 00780000 00771202
04 0080 00001400 76000548 12021800 007A1200 007A1202
05 0180 00001402 10000000 0C024800 000000D9 00000000
06 0102 00001408 76000000 01042800 00000000 00000000
07 0002 00001409 76000000 01025800 00000000 00000000
09 0004 0000140B 76000078 0904B800 00000000 00000000
```

2.5.3.17 Position Error (Cont'd.)

This is an example of a Mag Tape position error.

SEEK — The Seek address in C:5.

SENSE — The result of the most recent SENSE (for RADs and Packs only).

POSITIONS — The contents of the buffer's header P1 through P4.

2.5.3.18 I/O Memory Parity

When an I/O device reports a transmission memory error (bit 10 of TIORU1), the IOP has detected a memory parity error while accessing memory for the device.

```
G
--I/O-MEMORY PARITY
 SEQ  IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
00067 85C80 01 41 20 10400C27 00000010 000201F0 000225CB 00000C76 2100003D
CTX DEV   TIOR    TIORU1  OP  BUF    SEEK    SENSE
01 03F0 00001467 90800000 1201DD7D 1E820000 1E890900
02 01F0 00001469 18220C0C 010225CB 13880000 138B0500
03 0081 0000146B 76001816 0201C563 00000069 00000000
```

STATUS 0 — Device address reporting the parity.

STATUS 1 — Byte address of buffer in which the parity is detected.

STATUS 2 — Word address of parity location.

STATUS 3 — Contents of parity location.

2.5.3.19 CPU Memory Parity Interrupt

This is a Sigma 8,9 error report. This is the result of a X'57' interrupt - (MFI) which is caused by memory overheat or bus check fault detected at the memory. MFI is also issued when a memory parity trap occurs, but the trap handler clears MFI to avoid a duplicate error report. This report is similar to the CPU memory parity trap report (Refer to Paragraph 2.5.3.21).

The MFI handler polls the busy device and then polls the memories via the LMS and clears the memory status registers.

2.5.3.19 CPU Memory Parity Interrupt (Cont'd.)

After entering the Auditor to report the error, the program resumes execution from the point of the interruption (PSW1 and PSW2).

```
G
--CPU-MEMORY PARITY INTERRUPT
 SEQ  IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
00066 E5700 01 40 35 20000B5E 07000010 44050100 94C00000 00000000 40C5D9D9
CTX DEV   TIOR    TIORU1  OP  BUF    SEEK    SENSE
01 03F0 00000000 00000000 1201DD7D 1E820000 1E890900
02 01F0 00001469 98220C0C 010225CB 13880000 13870700
03 0081 0000146B 7600075F 0200EC79 00000068 00000000
```

STATUS 0 through STATUS 2 are the contents of the memory status word 0 through 2. STATUS 2 is altered for interleaving thus is the physical address that failed. STATUS 3 is the contents of the failing location.

2.5.3.20 CPU Memory Parity

This report is issued only for a Sigma 5, 6, or 7 and is a result of an interrupt to location 56.

The interrupt 56 handler searches memory for a parity error. It clears the first error found by restoring the contents of that location. Then it counts the number of remaining errors in core.

```
G
--CPU-MEMORY PARITY
 SEQ  IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
00009 85600 00 04 33 60000B56 00000000 00000001 00000000 04016061 FAFBFAFA
CTX DEV   TIOR    TIORU1  OP  BUF    SEEK    SENSE
01 02E0 00001490 90800000 0105E000 05930000 05950500
02 02F0 00001493 90800000 0105F800 00460000 004C0C00
03 00E0 00001496 90800000 1205E800 00540000 005A0A00
04 0002 00001244 10800000 01016000 00000000 00000000
06 0004 00001498 90000000 09017000 00000000 00000000
```

STATUS 0 = Number of paritys found (in hex).

STATUS 1 = Will always contain a zero.

STATUS 2 = Byte 0 contains the fault status. The rest of the word contains the word address of the first location with a parity. If no parity can be found, the last location +1 will appear.

STATUS 3 = The contents of the address in STATUS 2 when an error is found.

2.5.3.21 CPU - Memory Parity Trap

This report is used for a Sigma 8 or 9 only and is the result of a trap to location X'4C'.

When the 4C handler is entered, a diversion is set which will count subsequent 4C errors and report them under the double PDF report (Refer to Paragraph 2.5.3.25) while retaining the original environment. The handler polls the busy devices and then the memory via the LMS instruction and clears the memory status register of the errored memory. MFI is also reset to avoid a redundant error report. After the error is reported, the program resumes execution from the location specified by the PSWs.

```
G
--CPU-MEMORY PARITY TRAP
 SEQ   IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
000C1 E4C00 00 35 10 2000072F 07000008 08080400 94C00000 00000020 00007FFC
CTX DEV   TIOR    TIORU1  OP  BUF    SEEK    SENSE
01 02F0 00001490 90800000 120252E9 16110000 162C2C00
02 01F0 00001493 10800000 12032971 05400000 05630300
03 0081 00001496 10800000 0100A5D9 006E0E00 006E1203
05 0102 0000149D 10800000 0101E7A5 00000000 00000000
06 0002 00001244 90800000 010394B5 00000000 00000000
```

STATUS 0 through STATUS 2 are the contents of the memory status word 0 through 2. STATUS 2 is altered to reflect physical address (interleaving). STATUS 3 is the contents of the address in STATUS 2.

2.5.3.22 Processor Fault

This is a Sigma 8 or 9 report and is the result of an interrupt to location 56 which points to the PFI handler.

When the handler is entered, the devices are polled and then the processors are polled via the POLR instructions.

After the report is issued, the program resumes execution from the location specified by the PSWs.

```
G
--PROCESSOR FAULT
 SEQ   IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
00008 A5680 00 06 32 E0000882 07000000 00000000 40000100 00000040 00000000
CTX DEV   TIOR    TIORU1  OP  BUF    SEEK    SENSE
01 02F0 00001490 90800000 01011800 1AAE0000 1AB03000
02 01F0 00001493 90800000 12011000 0B930000 0B950500
03 00F3 00001496 90800000 12014000 00571200 00571202
04 0080 00001499 10800000 01010800 00BA0000 00420802
05 0181 0000149A 90000000 0C013000 0000000A 00000000
```

2.5.3.22 Processor Fault (Cont'd.)

STATUS 0 and STATUS 3 are always set to zero. STATUS 1 byte 0 contains the condition codes after the POLR is issued and rest of the word contains the IOP address. STATUS 2 is the status received as a result of POLR.

2.5.3.23 Watchdog Timer Runout

When a watchdog timer trap occurs, two types of error reports and subsequent recovery occurs.

(1) If the CPU is a Sigma 8 or 9 and the instruction that initiates the trap is a TIO, SIO, HIO or TDV instruction, the WDT handler will retrieve the IOP maintenance interface display groups via the appropriate read directs and then issue a reset IO to that IOPs CCU. The groups are displayed in the extended status portion of the error report in the following manner for MIOPs.

The first four words are group 0 through group 3. The next four words are the subchannel information (lower FAM and upper FAM) for the device which cause the trap (effective add). The last eight words are groups 8 through F if the device is on Channel A or groups 10 through 17 if on Channel B.

For RIOPs both lines are group 0 through 6.

```
G
--WATCHDOG TIMER RUN OUT
 SEQ   IDENT HR MN SC   PSW1     PSW2    STATUS 0 STATUS 1 STATUS 2 STATUS 3
00014 E4640 00 13 31 0000018D 07000008 4F040000 000001F0 80000010 88B50F0C
CTX DEV   TIOR    TIORU1  OP  BUF    SEEK    SENSE
01 03F0 0000145F 90800000 12036DE0 0E900000 0E911100
02 01F0 00001460 10000002 12036EA0 05C40000 05C30300
EXTENDED STATUS.
00000020 C0A2F41F 19818000 00000000 0700A308 88002C00 07009EEC 88200002
80000010 88B50E0C F000A308 D4202C00 00000000 00000000 00008000 00008000
```

STATUS 0 contains the instruction that trapped.

STATUS 1 contains the instruction effective address.

STATUS 2 contains group 0, group 8 or group 10 as per the above explanation.

STATUS 3 contains group 1, group 9 or group 11 as per the above explanation.

2.5.3.23 Watchdog Timer Runout (Cont'd.)

This report will always have an IDENT of E4640. The reset I/O will halt all the devices on Channel A and Channel B of the affected IOP.

(2) If the CPU is a Sigma 5, 6 or 7 or if the instruction that caused the trap is not a TIO, SIO, HIO or TDV, the WDT handler will not issue a reset I/O and will not retrieve and report the maintenance interface display groups.

```
G
--WATCHDOG TIMER RUN OUT
 SEQ  IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
00021 E4640 00 33 57 00000749 07000008 6E400000 00000000 00000000 00000000
CTX DEV   TIOR     TIORU1  OP  BUF    SEEK     SENSE
01 03F0 00000000 00000000 1203FF90 03B80000 03B93900
02 01F0 20001461 1000000C 0103FFC0 00000000 05C30300
```

STATUS 0 and STATUS 1 are the same as case 1.

STATUS 2 and STATUS 3 will always be zero.

This report can have the following IDENTS:

(1) E4600 Sigma 5,6 or 7 or Sigma 8,9 instruction aborted.

(2) E4610 Sigma 8,9 instruction completed successfully.

(3) E4620 Sigma 8, 9 memory bus hang up.

(4) E4640 Sigma 8,9 AIO instruction caused the trap.

(5) E4680 Sigma 8,9 DIO bus hang up.

If the IOP of the error reporting device (the LOG device) becomes hung and cannot be reset under program control, the system will appear to hang in a loop. The operator should set Sense Switch 1. If the loop tightens near location X'4B0' he may take it out of RUN, depress the IO reset, reset Sense Switch 1 and put the CPU back in RUN. This should result in a double PDF error report being issued (Refer to Paragraph 2.5.3.25). If this does not work, depress the system reset.

2.5.3.24 Instruction Exception Trap

This is a Sigma 8 and 9 report and is the result of a trap to location X'4D'. The 4D handler clears PDF and then reports the occurrence. After the error is reported, the program resumes execution at the location specified by the PSWs.

```
G
--INSTRUCTION EXCEPTION TRAP
 SEQ   IDENT HR MN SC   PSW1     PSW2     INST
00003 E4D00 00 01 01 20000E4A 00000000 66D0242E
```

2.5.3.25 Double PDF Fault

If a X'46' trap occurs while the CPU is still accessing the X'46' (watchdog timer) trap handler or if a X'4C' trap occurs while the CPU is still accessing the X'4C' (memory parity) trap handler, a double PDF report occurs.

Upon entering either the 46 or 4C trap, a diversion is set up. When the diversion handler is entered, this report is issued, and the program resumes execution from the location specified by the PSWs.

```
G
--DOUBLE PDF FAULT
 SEQ   IDENT HR MN SC   PSW1     PSW2   STATUS 0 STATUS 1 STATUS 2 STATUS 3
00029 E2000 01 54 32 00000576 07000008 4D8E0000 000001F0 000E4640 000E4640
```

STATUS 0 contains the instruction that cuased the second trap.

STATUS 1 contains the effective address of the instruction.

STATUS 2 contains the IDENT of the first trap.

STATUS 3 contains the IDENT of the second or current trap.

2.5.3.26 Power Failure

When a power OFF interrupt occurs, location X'51' is accessed. The 51 handler halts and records all the busy devices and saves the environment then enters a wait state.

When a power ON interrupt occurs, location X'50' is accessed. The 50 handler sets up a time wait of approximately 30 seconds. Then it reinitializes the interrupts, issues the power failure report and resumes program execution at the location specified by the PSWs.

2.5.3.26 Power Failure (Cont'd.)

The devices will all ''software time out''. The mag tapes will be in manual.

```
G
--POWER FAILURE
 SEQ   IDENT HR MN SC   PSW1      PSW2
0003F E5100 00 15 10 20000BE7 07000000
CTX DEV   TIOR     TIORU1   OP   BUF     SEEK     SENSE
01 02F0 0000148F 76010000 0102C4B0 13490000 13474700
02 01F0 00001492 76800000 0102C4C0 006A0000 00690900
03 00F2 00001495 76000000 0102C510 008B0400 008B0201
04 0080 00001498 7600000C 1202C530 008A0A00 008A0801
05 0180 00000000 00000000 0102C640 00000060 00000000
```

2.5.3.27 System Register Stack Fault

All trap and interrupt handlers must save the environment upon entry and restore it upon exit. The environment consists of:

(1) The current registers.

(2) The current program status doubleword.

There is a register stack which is used for this purpose. When the stack becomes out of balance, this report is issued.

After the report is issued, the program is reinitialized and the current pass restarted.

```
G
--SYSTEM REGISTER STACK FAULT
 SEQ   IDENT HR MN SC   PSW1      PSW2    STATUS 0 STATUS 1 STATUS 2 STATUS 3
00029 F42F0 00 19 18 00000026 00000000 000E4080 000E4080 000E4080 00004B00
```

STATUS 0 through STATUS 3 contains the IDENT of the last four traps or interrupts in inverse chronological order with STATUS 0 containing the most recent.

The register stacks are located at X'2520'. The return stacks are located at X'2560'.

3.0 APPENDIX

3.1 System Exerciser Program Layout

Location	Contents
2A - 3F	Base device boot loader.
40 - 13F	XPSDs.
140 - 1B1	Initialization program.
1B2 - 1FF	Patch area.
200 - 240	Environmental save and restore - PUSH & PULLS.
241 - 379	AUDITOR-LOGGER & BASE DRIVER (Base device handler).
37A - 3F1	Trap 40, 41, 42, 43, 44, 45, 48, 49, 4A and interrupt 88, 59, 5A and 5D handlers.
3F2 - 451	CAL4 handlers (4B).
452 - 4B9	Watchdog timer trap handler (46).
4BA - 4F1	Memory parity trap handler (4C).
4F2 - 519	Instruction exception trap (4D) and power ON (50) and OFF (51) interrupt handlers.
51A - 563	56 and 57 interrupt handlers.
564 - 71F	Real time clock supervisor (5B) handler.
720 - AC1	AIO or I/O interrupt (5C) handler plus the I/O test modules.
AC2 - B47	I/O test module Auditor interface routines.
B48 - BE9	Buffer return & distribute routine (PUTBUF & GETBUF).
BEA - F0F	Buffer tester.
F10 - 26CB	Communication control package and the abstract.
26CC - 27CF	Error file.
27D0 - 2A5C	Control, IOP, Interrupt and Configuration tables, COMMAP.
2B00 - 2FFF	Configurator.

3.2 List of Directives

1. ABSTRACT,Q1
2. AIO
3. BOOT,Q1
4. BRANCH,Q1
5. COMPARE,DESTINATION, Q1, Q2, Q3
6. CONFIGURE
7. DESELECT, Q1, Q2
8. DISPLAY, DESTINATION, Q1, Q2
9. ERRORS, Q1, Q2, Q3
10. EXPLAIN, DESTINATION, Q1
11. HALT
12. HIO, Q1, Q2, Q3
13. READ, Q1, Q2, Q3
14. PRINT, DESTINATION, Q1, Q2, Q3
15. REDUMP, Q1
16. RELOAD, Q1
17. REPLACE, DESTINATION, Q1, Q2
18. RUN
19. SELECT, Q1, Q2
20. SEARCH, DESTINATION, Q1, Q2, Q3
21. SIO, Q1, Q2, Q3
22. SNAP, Q1, Q2, Q3
23. SPREAD, DESTINATION, Q1, Q2, Q3
24. START, Q1, Q2
25. STORE, DESTINATION, Q1, Q2

3.2 List of Directives (Cont'd.)

26. SWITCH, Q1
27. TIO, Q1, Q2, Q3
28. TDV, Q1, Q2, Q3
29. UNSNAP
30. WRITE, Q1, Q2, Q3

3.3 List of Destinations

1. BYTES
2. COCLINE
3. CONTROL
4. ELEMENT
5. IOP
6. MEMORY
7. OPERATOR
8. PARAMETER
9. REGISTER
10. STATUS
11. SYSTEM
12. TIME

3.4 I/O Status Responses

3.4.1 AIO/TDV Status Response

BIT	TDV	AIO
	DISK STORAGE MODEL NUMBER 7242-43-44-46-47	
0	Data overrun	Data overrun
1	Flaw mark	Unassigned (set to zero)
2	Sector unavailable	
3	Unassigned (set to zero)	
4	Header verification error	Device interrupt
5	On cylinder	On cylinder
6	Seek timeout error	Seek timeout error
7	Header parity error	Unassigned
	RAD STORAGE MODEL NUMBER 7201-02-03-04-31-32-11-12	
0	Data overrun	Data overrun
1	Unassigned (set to zero)	Unassigned (set to zero)
2	Sector unavailable	Sector unavailable
3	Write protect violation	Write protect violation
4	Unassigned (set to zero)	Unassigned (set to zero)
5		
6		
7		
	MAG TAPE MODEL NUMBER 7320-21-22-23-61-62-65-71-72-74	
0	Data overrun	Data overrun
1	Write permitted	Device end
2	Write protect violation error	Write protect violation error
3	End of file	End of file
4	Unassigned (set to zero) Ⓐ	Ⓐ
5	Load point	Unassigned (set to zero)
6	End of tape	
7	Rewind on-line	

Ⓐ For 7361-62-65-71-72-74 only; for remaining models, Bit 4 is ''noncorrectable read error'' for both TDV and AIO

BIT	TDV	AIO
	LINE PRINTER MODEL NUMBER 7440-45-50	
0	Unassigned (set to zero)	Unassigned (set to zero)
1	Print fault Ⓑ	Data transmission complete
2	Paper low	
3	Top of page	
4	Paper moving	
5	Paper runaway	Unassigned (set to zero)
6	Unassigned Ⓒ	
7	(set to zero) Ⓓ	
	KEYBOARD/PRINTER MODEL NUMBER 7012-14-20-21-8091-92	
0	Unassigned (set to zero)	
1		
2		
3	Reader, manual mode (SR)	
4	Off-line (ASR)	Unassigned (set to zero)
5		
6	Unassigned (set to zero)	
7		
	PAPER TAPE SYSTEM MODEL NUMBER 7060	
0	Rate error	
1	Punch, tape low	
2	Punch, manual mode	
3	Reader, manual mode	(No status for AIO)
4		(set to zero)
5		
6	Unassigned (set to zero)	
7		

Ⓑ For 7440 and 7445 only: ''type line in odd sector'' on 7450. At present time not used or connected.

Ⓒ ''Print order expected'' on 7450.

Ⓓ ''Maintenance panel used'' on 7450.

BIT	TDV	AIO
	CARD READER MODEL NUMBER 7120-21-22-40	
0	Data overrun	Data overrun
1	Unassigned (set to zero)	Unassigned (set to zero)
2		
3		
4		
5		
6		
7		
	CARD PUNCH MODEL NUMBER 7160-65	
0	Rate error	Rate error
1	Unassigned (set to zero)	Ⓔ
2	Read check	Ⓕ
3	Parity error	
4	Row 15 (RC15)	Unassigned (set to zero)
5	Test switch on	
6	Unassigned (set to zero)	
7		Ⓖ

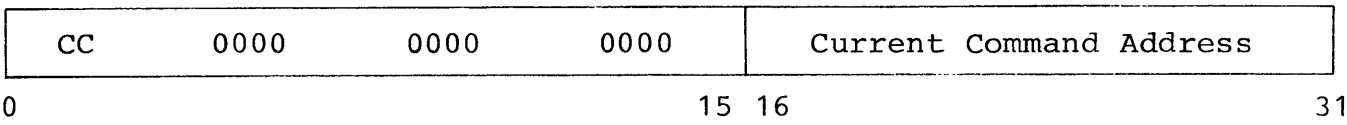
Ⓔ ''Data transmission complete'' for 7165.

Ⓕ ''Punch error'' or ''read check'' for 7165.

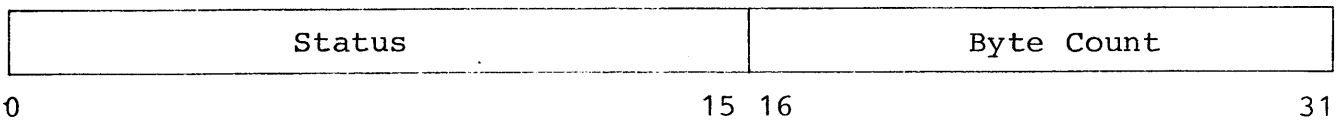
Ⓖ ''Interrupt call'' for 7165.

3.4.2 TIO Status Response

Word into register R



Word into register Ru1



CC = Condition codes

Condition code settings:

1 2 3 4	Result of TIO
0 0 - -	I/O address recognized and acceptable SIO is currently possible.
0 1 - -	I/O address recognized but acceptable SIO is not currently possible.
1 0 - -	Busy SIOP or controller busy with another device.
1 1 - -	I/O address not recognized.

3.4.2 TIO Status Response (Cont'd.)

Status:

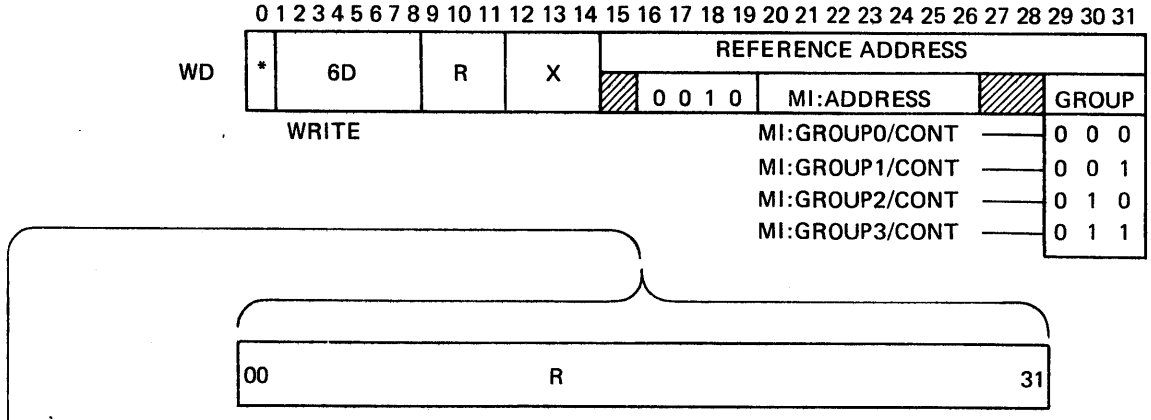
Position and State in Register Ru1

Device Status Byte								Operational Status Byte								Significance for SIO, HIO and TIO
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	Interrupt pending
–	0	0	–	–	–	–	–	–	–	–	–	–	–	–	–	Device ready
–	0	1	–	–	–	–	–	–	–	–	–	–	–	–	–	Device not operational
–	1	0	–	–	–	–	–	–	–	–	–	–	–	–	–	Device unavailable
–	1	1	–	–	–	–	–	–	–	–	–	–	–	–	–	Device busy
–	–	–	0	–	–	–	–	–	–	–	–	–	–	–	–	Device manual
–	–	–	1	–	–	–	–	–	–	–	–	–	–	–	–	Device automatic
–	–	–	–	1	–	–	–	–	–	–	–	–	–	–	–	Device unusual end
–	–	–	–	–	0	0	–	–	–	–	–	–	–	–	–	Device controller ready
–	–	–	–	–	0	1	–	–	–	–	–	–	–	–	–	Device controller not operational
–	–	–	–	–	1	0	–	–	–	–	–	–	–	–	–	Device controller unavailable
–	–	–	–	–	1	1	–	–	–	–	–	–	–	–	–	Device controller busy
–	–	–	–	–	–	–	0	–	–	–	–	–	–	–	–	Unassigned
–	–	–	–	–	–	–	–	1	–	–	–	–	–	–	–	Incorrect length
–	–	–	–	–	–	–	–	–	1	–	–	–	–	–	–	Transmission data error
–	–	–	–	–	–	–	–	–	–	1	–	–	–	–	–	Transmission memory error
–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	–	Memory address error
–	–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	IOP memory error
–	–	–	–	–	–	–	–	–	–	–	–	–	1	–	–	IOP control error
–	–	–	–	–	–	–	–	–	–	–	–	–	–	1	–	IOP halt
–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	1	Selector IOP busy

Device Status Byte								Operational Status Byte								Significance for AIO
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
–	–	–	–	–	–	–	–	1	–	–	–	–	–	–	–	Incorrect length
–	–	–	–	–	–	–	–	–	1	–	–	–	–	–	–	Transmission data error
–	–	–	–	–	–	–	–	–	–	1	–	–	–	–	–	Zero byte count interrupt
–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	–	Channel end interrupt
–	–	–	–	–	–	–	–	–	–	–	–	1	–	–	–	Unusual end interrupt
–	–	–	–	–	–	–	–	–	–	–	–	–	0	–	–	Unassigned
–	–	–	–	–	–	–	–	–	–	–	–	–	–	0	–	Unassigned
–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0	Unassigned

3.4.3 RIOP Display Status

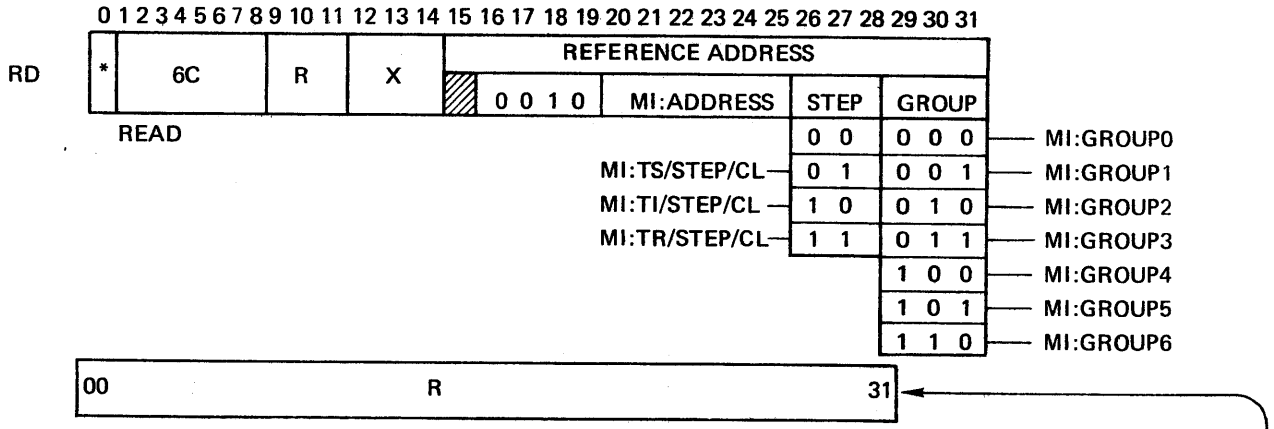
MI:CONTROL



	MI:GROUP0/CONT	MI:GROUP1/CONT	MI:GROUP2/CONT	MI:GROUP3/CONT
00	TESTMODE1			
01	TESTMODE2			
02	TESTMODE3			
03	TESTMODE4			
04	MI:POWERLO		-------	
05	MI:POWERHI		-------	
06	MI:DVT		SET ST:ZBCI	
07	MI:DVO		SET ST:MAE	
08	MI:CLOCKLO		SEND NL32	
09	MI:CLOCKHI		SEND NM32	
10			SET MPE	
11			SET PE:D	
12			SET ST:RATER	
13			SET ST:SUN	
14			SET ST:WPV	
15			SET ST:SPM	
16		-------	SET ST:INL	
17		-------	SET ST:IOPH	
18		MI:PH00/STOP	SET ST:BTE	
19		MI:PH01/STOP	SET ST:HCE	
20		MI:PH02/STOP		
21		MI:PH03/STOP		
22		MI:PH04/STOP		
23		MI:PH05/STOP		
24		MI:PH10/STOP1		
25		MI:PH10/STOP2		
26		MI:PH11/STOP		
27		MI:PH12/STOP		
28		MI:DATA/STOP		
29		MI:TS/STEP		MI:SYNC
30		MI:TI/STEP		MI:PHASE/STOP/CL
31		MI:TR/STEP		MI:RESET

RIOP MAINTENANCE INSTRUCTION FORMATS

MI:DISPLAY



	MI:GROUP0	MI:GROUP1	MI:GROUP2	MI:GROUP3	MI:GROUP4	MI:GROUP5	MI:GROUP6
00	PH00	TESTMODE1	MAR	BPA	R:L0	KA0	M00
01	PH01	TESTMODE2	MDR	BPB	R:L1	KA1	M01
02	PH02	TESTMODE3	MAE	BFC	R:L2	KA2	M02
03	PH03	TESTMODE4	MQ:D	BPD	R:L3	KA3	M03
04	PH04	MI:POWERLO	MPE:R	MBPA	R:LW0	KA4	M04
05	PH05	MI:POWERHI	MPE	MBPB	R:LW1	KA5	M05
06	PH10	MI:DVT	MSTART	MBPC	R:LW2	KA6	M06
07	PH11	MI:DVO	MBUSY	MBPD	R:LW3	KA7	M07
08	PH12	MI:CLOCKLO	M32:D	HBPA	R:LR0	KB0	M08
09	PH20	MI:CLOCKHI	NL32	HBPB	R:LR1	KB1	M09
10	PH30	MPE	NL10	HBPC	R:LR2	KB2	M10
11	M:DATA	MPE:R	NL11	HBPD	R:LR3	KB3	M11
12	M:IN	ST:RATER	NL12	IBPA	R:K0	KB4	M12
13	PBFNC0	ST:SUN	NL13	IBPB	R:K1	KB5	M13
14	PBFNC1	ST:WPV	NL14	IBPC	R:K2	KB6	M14
15	PBFNC2	ST:SPM	NL15	IBPD	R:K3	KB7	M15
16	BUSY	ST:INL	NL16		R:K4	KC0	M16
17	-------	ST:IOPH	NL17		RE8	KC1	M17
18	-------	MI:PH00/STOP	NL18		RF8	KC2	M18
19	-------	MI:PH01/STOP	NL19		RG8	KC3	M19
20		MI:PH02/STOP	NL20		RA8	KC4	M20
21		MI:PH03/STOP	NL21		RB8	KC5	M21
22		MI:PH04/STOP	NL22		RC8	KC6	M22
23		MI:PH05/STOP	NL23		RD8	KC7	M23
24	ORDREAD	MI:PH10/STOP/1	NL24	MFULL	JA8	KD0	M24
25	ORDWRITE	MI:PH10/STOP2	NL25	HFULL	JB8	KD1	M25
26	ORDSEEK	MI:PH11/STOP	NL26	IFULL	JC8	KD2	M26
27	ORDSENSE	MI:PH12/STOP	NL27	JFULL/1	JD8	KD3	M27
28	ORDCHWRT	MI:DATA/STOP	NL28	KFULL/2	KA8	KD4	M28
29	ORDRELEASE	MI:TS/STEP	NL29	-------	KB8	KD5	M29
30	ORD3	MI:TI/STEP	NL30	BCZ	KC8	KD6	M30
31	ORD7	MI:TR/STEP	NL31	(BC=>4)	KD8	KD7	M31

RIOP MAINTENANCE INSTRUCTION FORMATS

3.4.4 MIOP Display Status

READ DIRECT INSTRUCTION FORMAT

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
*	6C							R				X			0	MODE FIELD				UNIT ADDRESS							DISPLAY GROUP ADDRESS [9]					
																0	0	1	0												

READ DIRECT DATA FORMAT

CCU	CHANNEL	M.S.	GROUP ADDRESS	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
			0 0 0 0 0	DWAT1	LX21	LXFAM-A	LXFAM-B	LXFNC	P:MW0:R	P:MW1:R	P:MW2:R	P:MW3:R	P:L32:D	P:L10:D	P:L11:D	P:L12:D	P:L13:D	P:L14:D	P:L15:D	P:L16:D	P:L17:D	P:L18:D	P:L19:D	P:L20:D	P:L21:D	P:L22:D	P:L23:D	P:L24:D	P:L25:D	P:L26:D	P:L27:D	P:L28:D	P:L29:D	P:L30:D	P:L31:D
			0 0 0 0 1	NRGA-1	NRGB-1	MXJA	MXJB	MXD	CONNECT	CONNECT-B	DLSTOP/1	MAR	MAE/CCU	MDR	P:MQ:R	P:MUWS:R	DWAT2	TSPW/1	TMPW	SNAPTP1	SNAPTP2	SNAPTP3	MI/GNDM31	MI/GNDCFAM37	MI/CNT/EN	SNAPMODE	MIC0	MIC1	MIC2	MIC3	SNAPGPA0	SNAPGPA1	SNAPGPA2	SNAPGPA3	SNAPGPA4
			0 0 0 1 0	CCUN1	CCUN2	CCUN3	NCPUN0	NCPUN1	---	CCUN0		NCLKMGHI	---	PB:ADRS0	PB:ADRS1	PB:ADRS2	PB:ADRS3	PB:ADRS4	MATCH/IOP	MATCH/CPU	LSM	TORD	EH	SELECT	CONNECT	CCURQ	JEVEN	DWA	SWREQOUT	PRL1	PRL2	FNDASC:D	INH	AP	NCOND1	NCOND2	COND3
	A		0 1 0 0 0	PHI	PH00	PH01-3	PH02-1	PH04	PH05	PH06	PH07-1	PH08-1	PH09	PH10	PH11	PH12	FNC-1	WCF	LSE	LSM	ORD	OUT	NIER	IO8	IOEVEN-08	CONNECT	CCURQ	JEVEN	DX2:R	---	PFI/1	IRA	IRB	CNST	PR/OUT	NCOND1	NCOND2	COND3
	B		1 – 0 0 0	ED:R	BM:R/1	FSL:R	DX4:R/1-1	RS:R	SC:R/1	FS:R	RSA:R	ED:R	ES:R	NPR1	NPR2	OVRLAP	FIN	---	---	ORD	OUT	NIER	IO8	IOEVEN-08	CONNECT	CCURQ	JEVEN	DX2:R	---	IR0	IR1	IR2	IR3	IR0	IR1	IR2	IR3
	A		0 1 0 0 1	A0	A1	A2	A3	A4	A5	A6	A7	BA00	BA01	BA02	BA03	BA04	BA05	BA06	BA07	BA08	BA09	BA10	BA11	BA12	BA13	BA14	BA15	BA16	BA17	BA18	BA19	BA20	BA21	BA22	BA23
	B		1 – 0 1 0	CA00	CA01	CA02	CA03	CA04	CA05	CA06	CA07	CA08	CA09	CA10	CA11	CA12	CA13	CA14	CA15	CA16	CA17	CA18	CA19	CA20	---	CCF	---									
	A		0 1 0 1 1	FAMP1	FAMP2	FAMP3	FAMP4	FAMP5	FAMP6	---	---	DC	SIL	HTE	SK	BK	---	CM	---	BC00	BC01	BC02	BC03	BC04	BC05	BC06	BC07	BC08	BC09	BC10	BC11	BC12	BC13	BC14	BC15
	B		1 – 0 1 1									IL	TDE	TME	MAE	IOPME	IOPCNE	IOPHLT	BCF	DC	IZBC	CC	ICE	HTE	IUE	SIL	SK	A4	A5	A6	A7	ZBCI		CEI	UEI
	A		0 1 1 0 0																	MS/DB0:R	MS/DB1:R	MS/DB2:R	MS/DB3:R	MS/DB4:R	MS/DB5:R	MS/DB6:R	MS/DB7:R	MS/DA0:R	MS/DA1:R	MS/DA2:R	MS/DA3:R	MS/DA4:R	MS/DA5:R	MS/DA6:R	MS/DA7:R
	B		1 – 1 0 0																																
	A		0 1 1 0 1																	MS/DD0:R	MS/DD1:R	MS/DD2:R	MS/DD3:R	MS/DD4:R	MS/DD5:R	MS/DD6:R	MS/DD7:R	MS/DC0:R	MS/DC1:R	MS/DC2:R	MS/DC3:R	MS/DC4:R	MS/DC5:R	MS/DC6:R	MS/DC7:R
	B		1 – 1 0 1																																
	A		0 1 1 1 0																	MS/PHFS	MS/PHRS	MS/PHFSL	MS/PHSRV	MS/PHIO	AVO:R	MS/AVO:D	MS/SERVICON	MS/CDN	MS/UNE	MS/DATA	MS/IN	MS/DCBUSY	MS/CCH	MS/DX4:D	MS/ORD7
	B		1 – 1 1 0																																
	A		0 1 1 1 1																	MS/ED:R	MS/ES:R	MS/CSL	MS/DAP:D	MS/PC:R	MS/ONLINE	MS/IER	MS/CIL	MS/SIOU	MS/TIO:R	MS/TDV:R	MS/HIOU	MS/AIO:R	MS/ASC:R	MS/RSA:R	MS/FS:R
	B		1 – 1 1 1																																

NOTES:

READ DIRECT

[9] WHEN THE MIOP IS NOT IN THE SNAPSHOT MODE, THE DISPLAY-GROUP-ADDRESS SELECTS ONE OF 19 DATAWORD FORMATS TO BE GATED ONTO THE DIO DATA LINES (DIODB00-31) DURING FSA. IF THE MIOP IS IN THE SNAPSHOT MODE, THE DISPLAY–GROUP-ADDRESS IS IGNORED. THE FORMAT SELECTION IS MADE BY THE SNAPSHOT-GROUP-ADDRESS SPECIFIED BY THE LAST WRITE-DIRECT, WHICH SETS THE MIOP INTO SNAPSHOT MODE.

[10] BITS 08-31 ARE READ FROM C00-23 RESPECTIVELY.

[11] BITS 08-15 ARE READ FROM C24-31, WHICH ARE CONTROLLED BY LSM; BITS 16-31 ARE READ FROM C32-47, WHICH ARE CONTROLLED BY LSE.

LSE. } [10]

NLSE. }

LSE,LSM } [11]

NLSE,NLSM }

3.4.5 Memory Status

Data Bits	READ STATUS 0
32	PARITY BIT
31	PORT 12
30	PORT 11
29	PORT 10
28	PORT 9
27	PORT 8
26	PORT 7
25	PORT 6
24	PORT 5
23	PORT 4
22	PORT 3
21	PORT 2
20	PORT 1
19	FAST PORT OR 0
18	
17	
16	
15	Bank Number L.S.B.
14	Bank Number M.S.B.
13	Last Parity Bit Written
12	Subsequent Faults
11	
10	
9	
8	
7	BMU Temp. or Voltage Failure
6	Port Selection Error
5	Loop Check Error
4	Data in Error
3	Address Parity
2	Uncorrectable Parity Error in core detected on Write Partial
1	Uncorrectable Parity Error in core on Read
0	Always 0 for Core Memory

READ STATUS 1
PARITY BIT
CM4 Clock Margin 4 Early DR POK
CM3 Clock Margin 3 Late Strobe
CM2 Clock Margin 2 Early Sense Strobe
CM1 Clock Margin 1 Late Write Half Cycle
CM0 Clock Margin 0 Early Write Half Cycle
Power Normal
Memory Type***
Memory Type***
Unit Size**
Unit Size**
Memory Unit Number L.S.B.
Memory Unit Number
Memory Unit Number
Memory Unit Number M.S.B.
Bank Size*
Bank Size*
2 Way Interleave
4 Way Interleave

READ STATUS 2
PARITY BIT
EFFECTIVE MEMORY ADDRESS (MAPPED OR INTERLEAVED)
DATA ERROR CORRECTION CODE

OM	MW0	MW1	MW2	MW3	Function	Instruction
1	0	0	0	0	Read & Set	LAS
1	0	0	0	1	Read & Inhibit Parity	
1	0	0	1	0	Read & Change Parity	
1	0	0	1	1		
1	0	1	0	0		
1	0	1	0	1		
1	0	1	1	0		
1	0	1	1	1	Set Clck. Margin	
1	1	0	0	0	Read Memory Status 0	LMS
1	1	0	0	1	Read Memory Status 1	LMS
1	1	0	1	0	Read Memory Status 2	LMS
1	1	0	1	1		
1	1	1	0	0	Read Status 0 and Clear	
1	1	1	0	1		
1	1	1	1	0	Read Status 2 and Clear Snapshot	
1	1	1	1	1	Clear Memory	

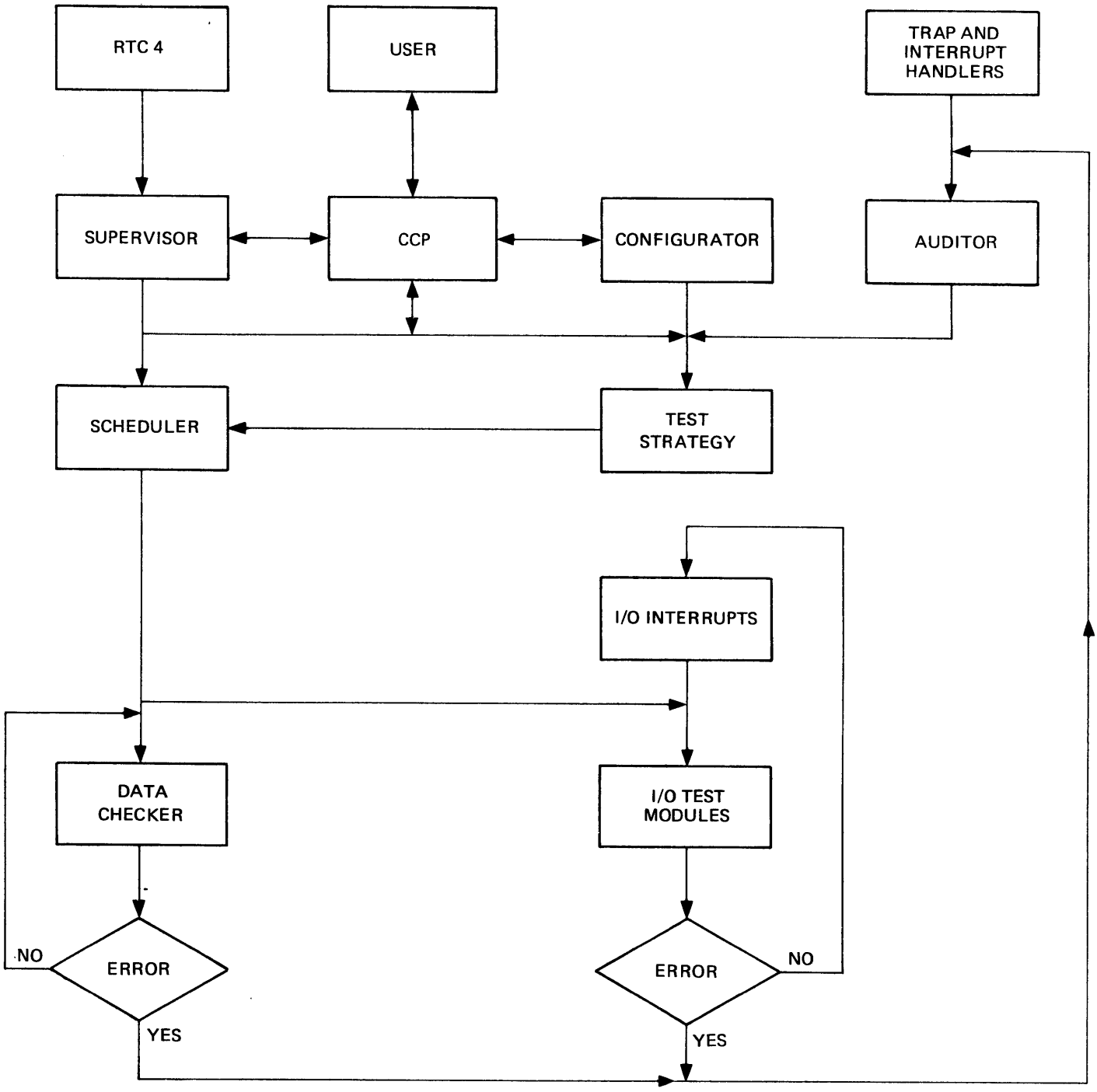
**Unit Size	
Data Bit	Size
8 9	
0 0	8K'
0 1	16K
1 0	32K
1 1	

*Bank Size	
Data Bit	Bank Size
2 3	
0 0	8K
0 1	16K
1 0	
1 1	

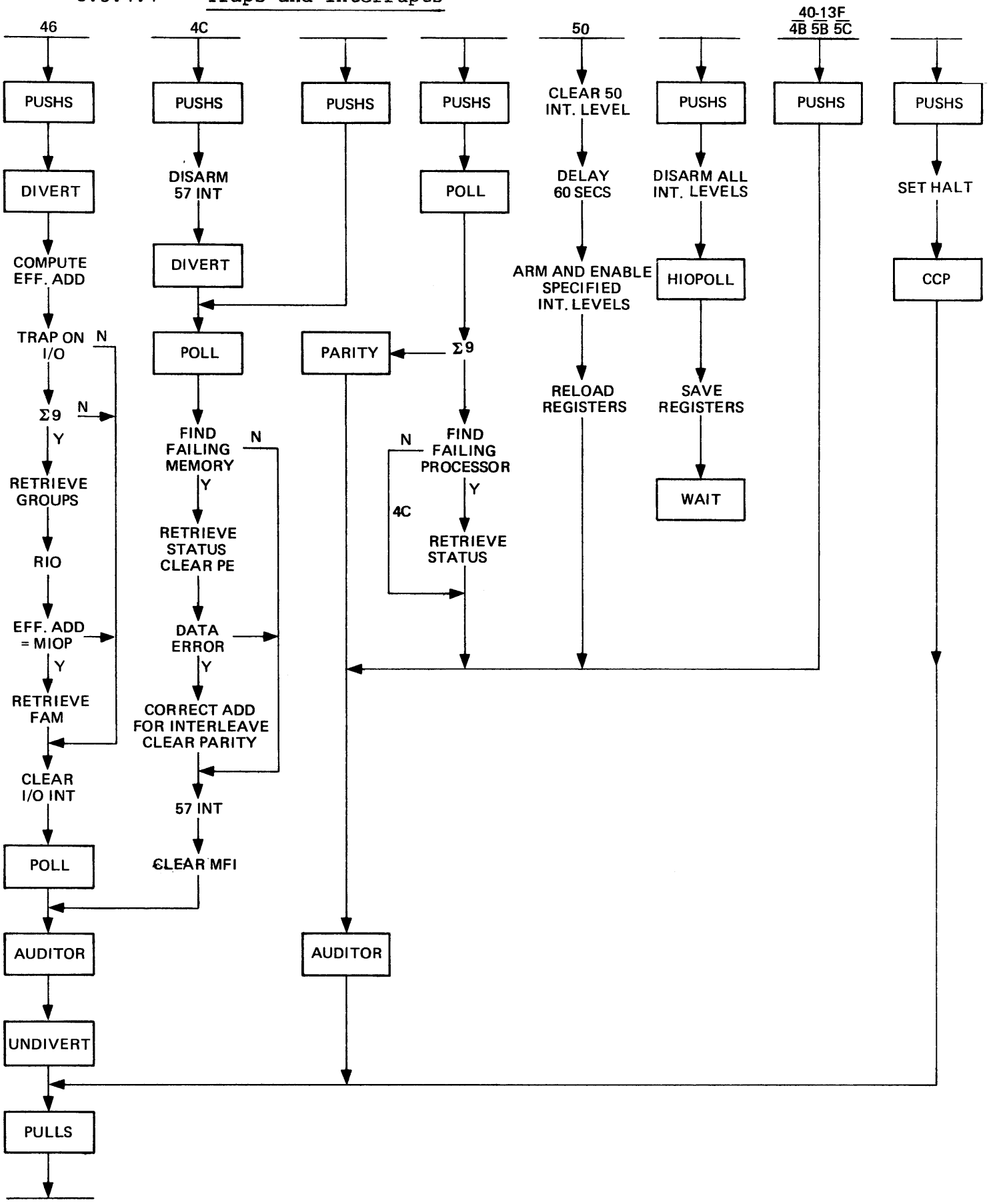
***Memory Type	
Data Bit	Type
0 11	
0 0	Core
0 1	LCS
1 0	Plated Wire
1 1	

3.5 Flow Charts

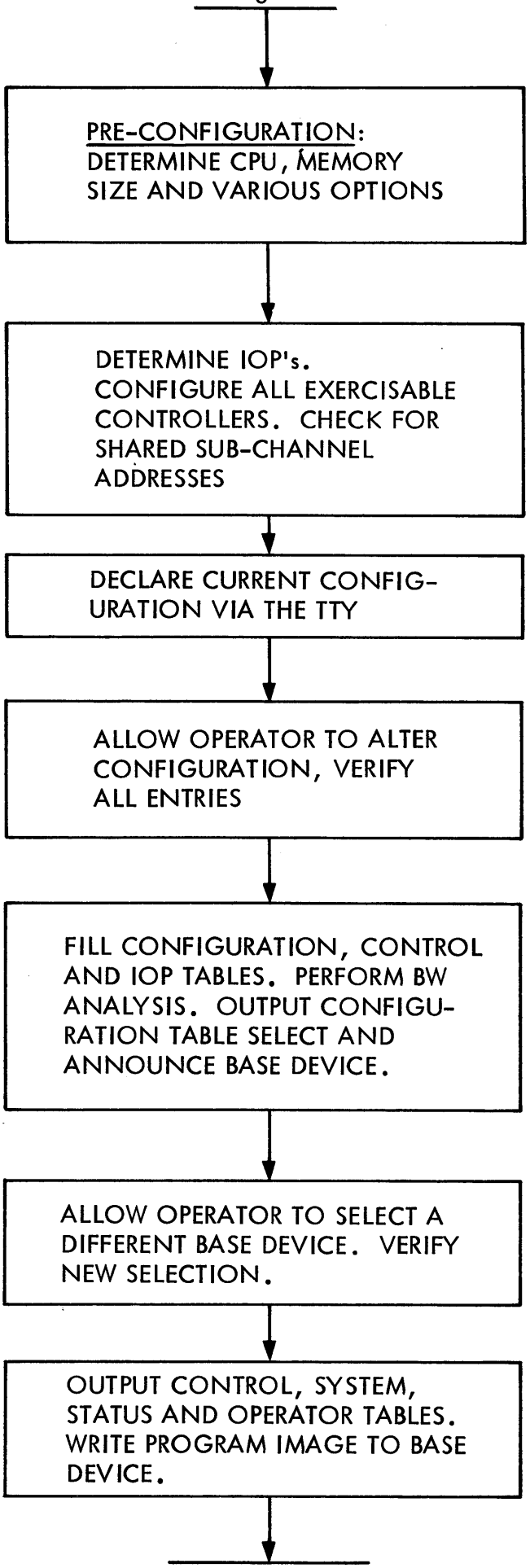
3.5.1 System Exerciser Overview



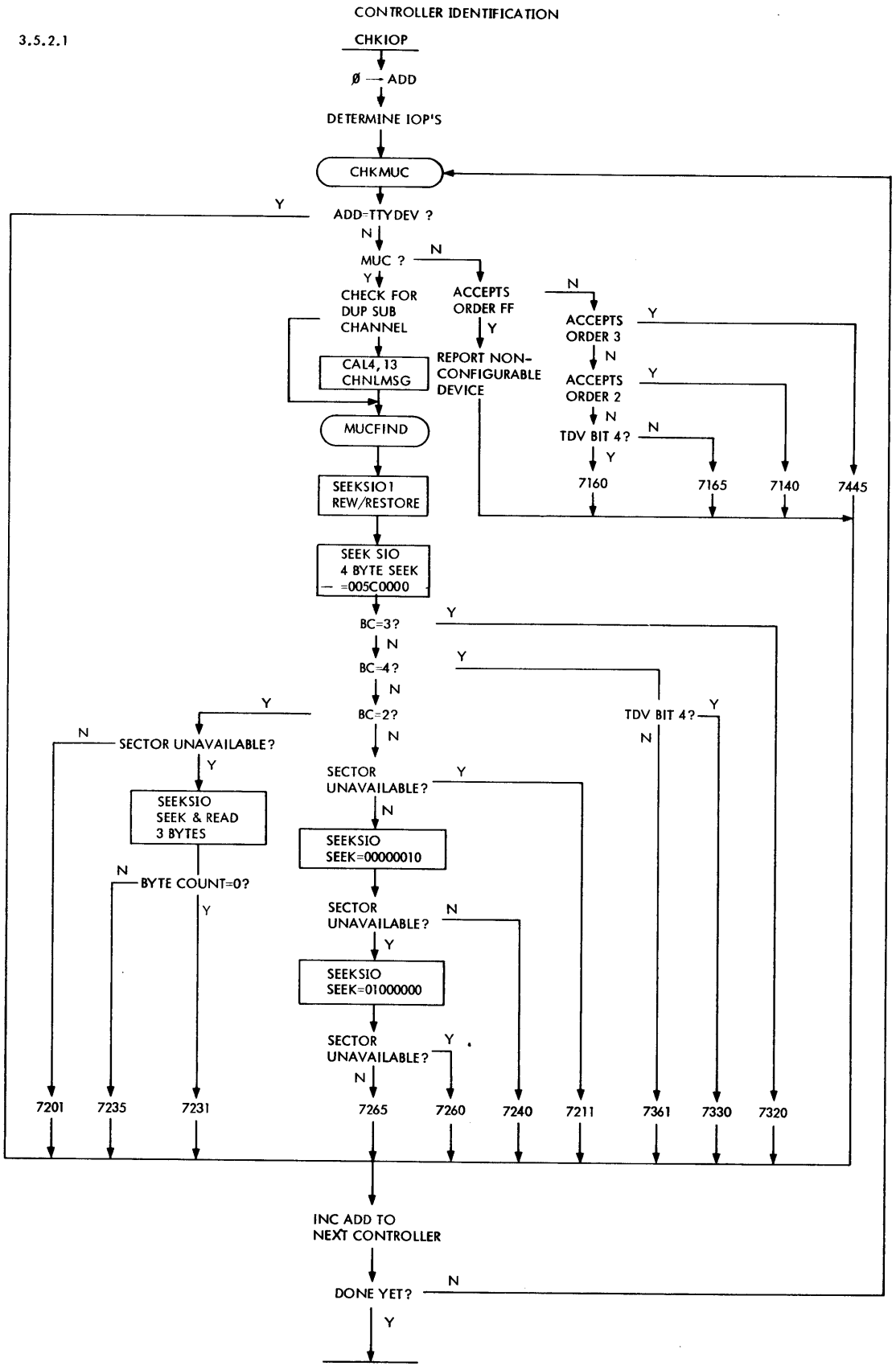
3.5.1.1 Traps and Interrupts

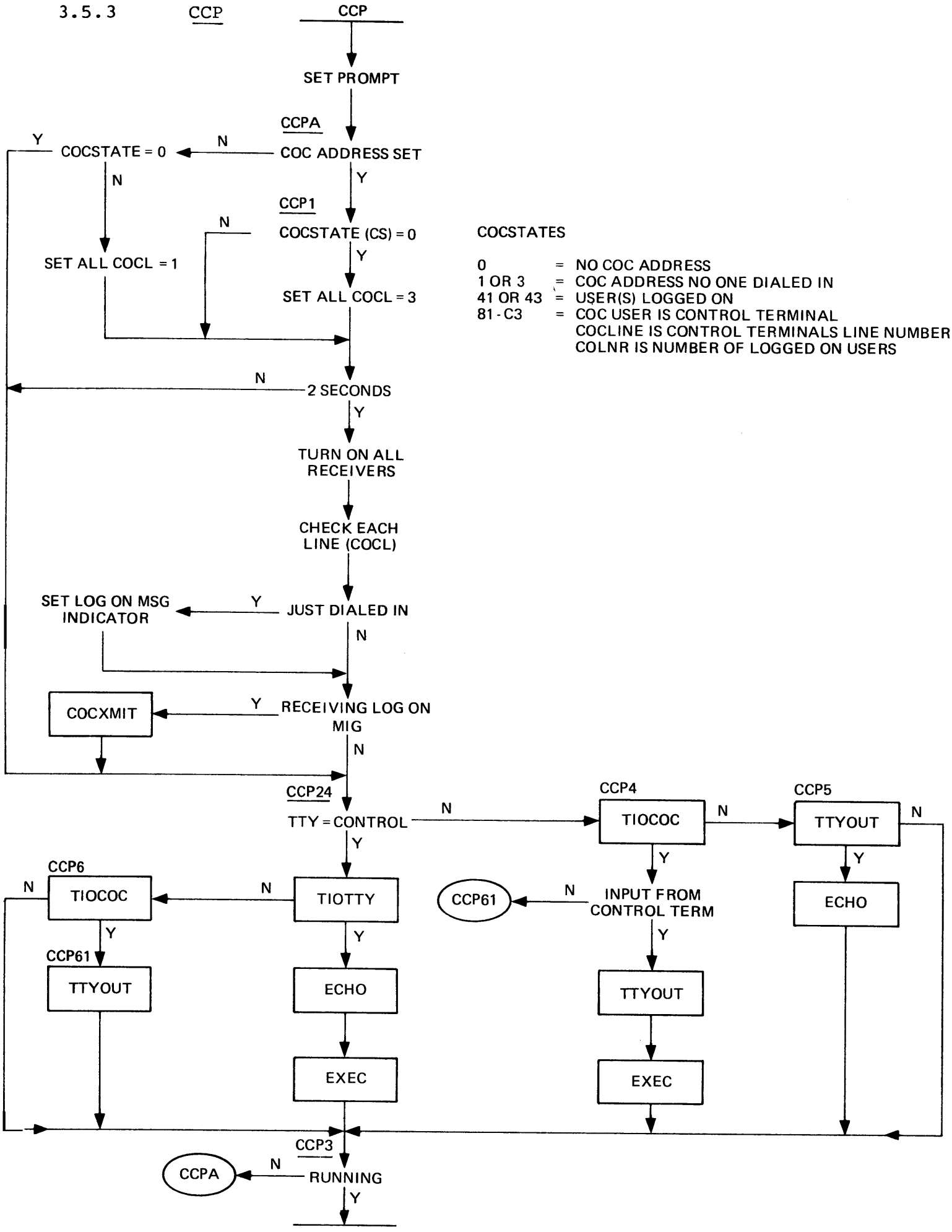


Configurator



3.5.2.1

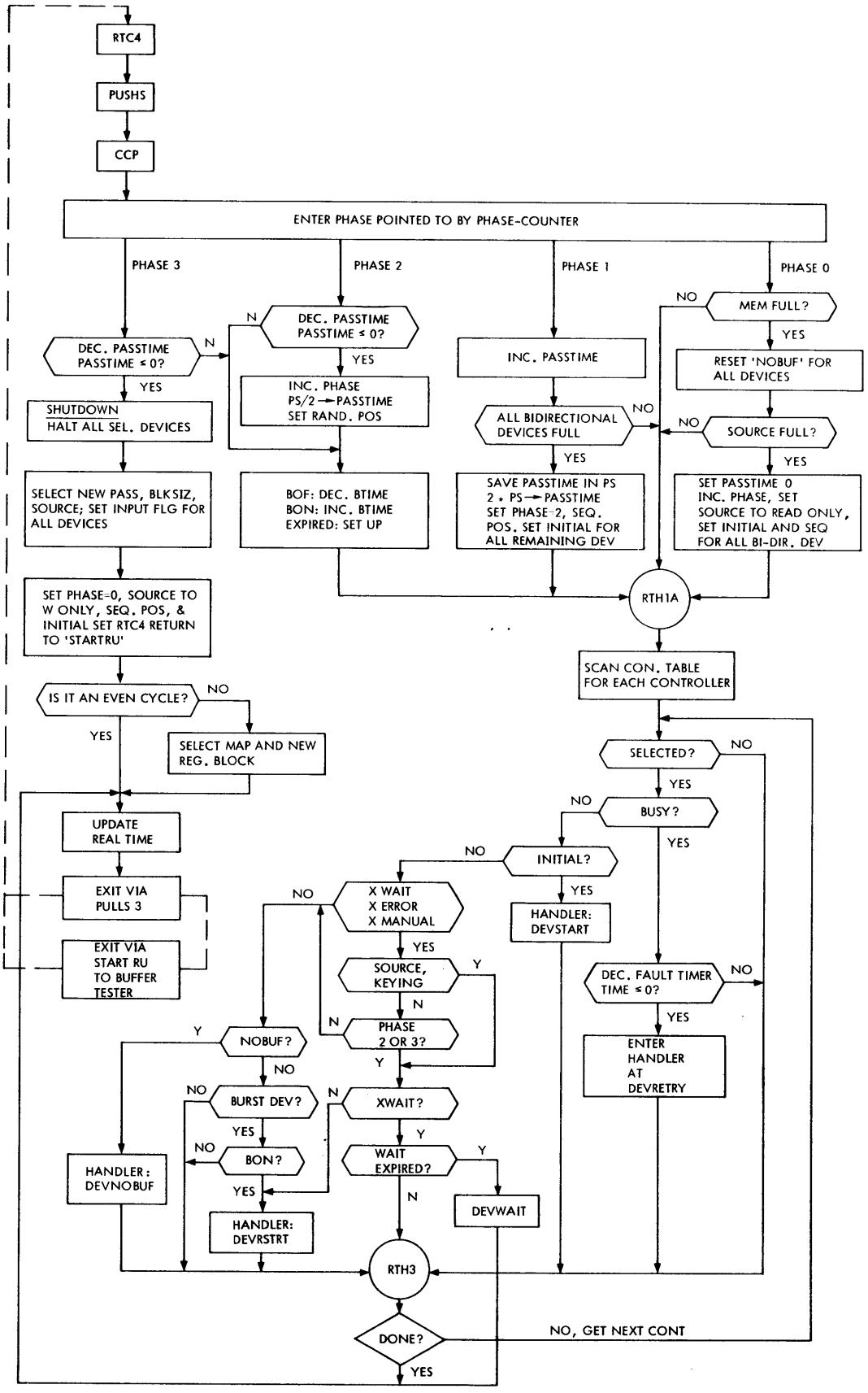

CONTROLLER IDENTIFICATION
CHKIOP
Ø → ADD
DETERMINE IOP'S
CHKMUC
ADD=TTYDEV ?
Y
N
MUC ?
N
Y
CHECK FOR DUP SUB CHANNEL
CAL4, 13 CHNLMSG
MUCFIND
SEEKSIO1 REW/RESTORE
SEEK SIO 4 BYTE SEEK — =005C0000
ACCEPTS ORDER FF
Y
REPORT NON-CONFIGURABLE DEVICE
N
ACCEPTS ORDER 3
Y
N
ACCEPTS ORDER 2
Y
N
TDV BIT 4?
N
Y
7160
7165
7140
7445
BC=3?
Y
N
BC=4?
Y
N
BC=2?
Y
N
SECTOR UNAVAILABLE?
N
Y
SEEKSIO SEEK & READ 3 BYTES
BYTE COUNT=0?
N
Y
SECTOR UNAVAILABLE?
Y
N
SEEKSIO SEEK=00000010
SECTOR UNAVAILABLE?
N
Y
SEEKSIO SEEK=01000000
SECTOR UNAVAILABLE?
Y
N
TDV BIT 4?
Y
N
7201
7235
7231
7265
7260
7240
7211
7361
7330
7320
INC ADD TO NEXT CONTROLLER
DONE YET?
N
Y



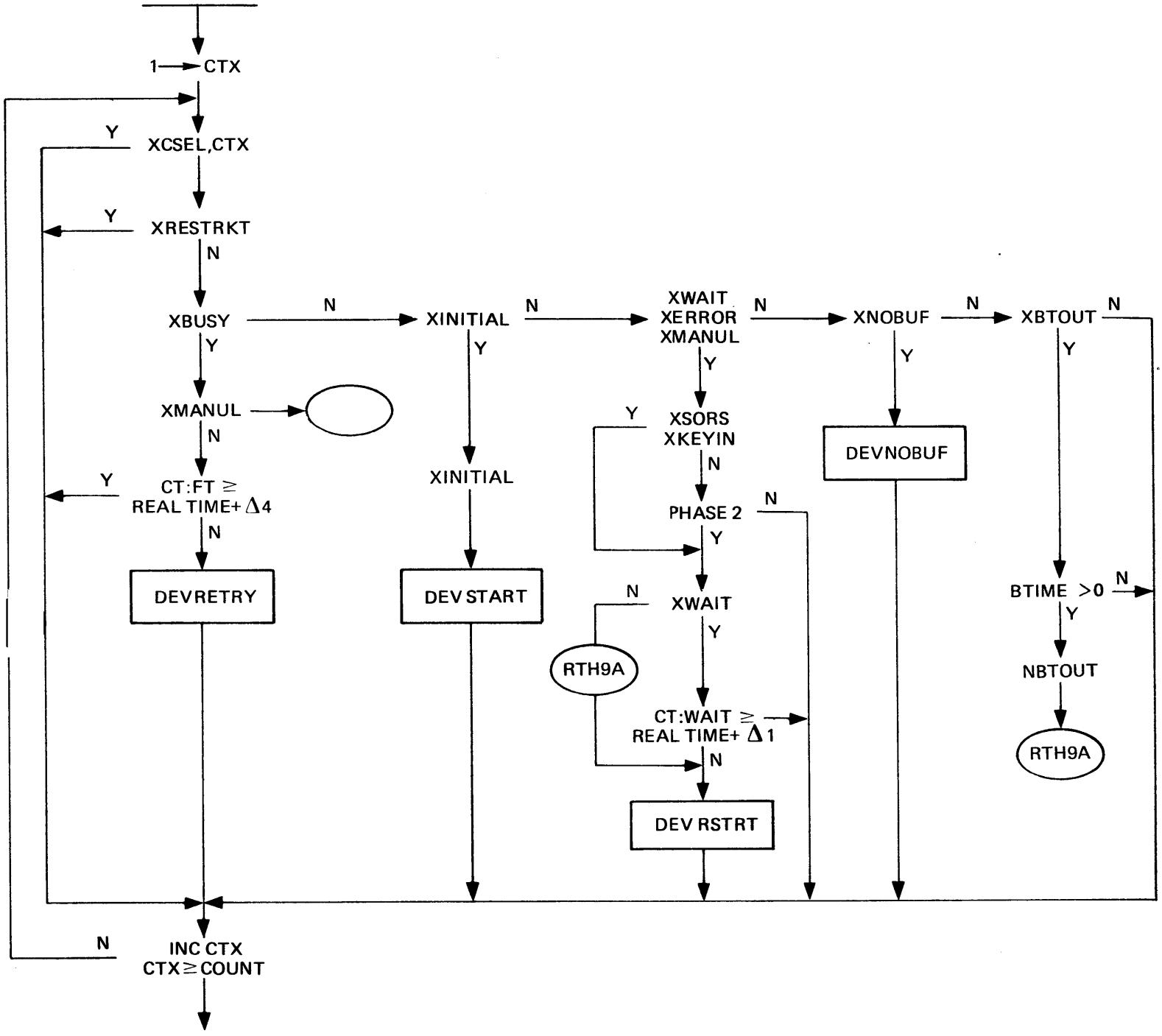
COCSTATES

0	=	NO COC ADDRESS
1 OR 3	=	COC ADDRESS NO ONE DIALED IN
41 OR 43	=	USER(S) LOGGED ON
81 - C3	=	COC USER IS CONTROL TERMINAL

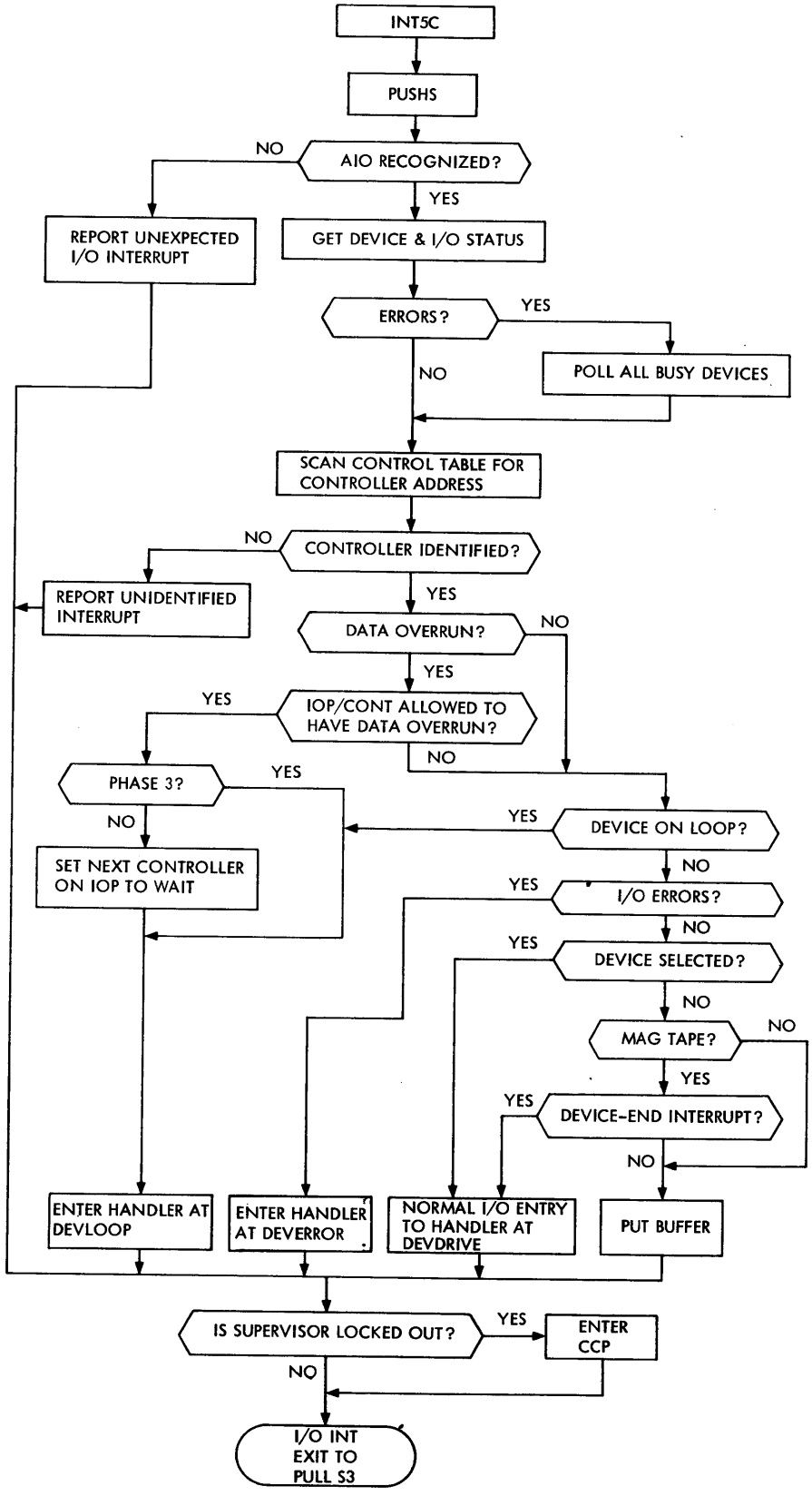
COCLINE IS CONTROL TERMINALS LINE NUMBER
COLNR IS NUMBER OF LOGGED ON USERS


RTC4
PUSHS
CCP
ENTER PHASE POINTED TO BY PHASE-COUNTER
PHASE 3
PHASE 2
PHASE 1
PHASE 0
DEC. PASSTIME
PASSTIME ≤ 0?
N
YES
SHUTDOWN
HALT ALL SEL. DEVICES
SELECT NEW PASS, BLKSIZ, SOURCE; SET INPUT FLG FOR ALL DEVICES
SET PHASE=0, SOURCE TO W ONLY, SEQ. POS, & INITIAL SET RTC4 RETURN TO 'STARTRU'
DEC. PASSTIME
PASSTIME ≤ 0?
N
YES
INC. PHASE
PS/2 → PASSTIME
SET RAND. POS
BOF: DEC. BTIME
BON: INC. BTIME
EXPIRED: SET UP
INC. PASSTIME
ALL BIDIRECTIONAL DEVICES FULL
NO
YES
SAVE PASSTIME IN PS
2 * PS → PASSTIME
SET PHASE=2, SEQ. POS. SET INITIAL FOR ALL REMAINING DEV
MEM FULL?
NO
YES
RESET 'NOBUF' FOR ALL DEVICES
SOURCE FULL?
NO
YES
SET PASSTIME 0
INC. PHASE, SET SOURCE TO READ ONLY, SET INITIAL AND SEQ FOR ALL BI-DIR. DEV
RTH1A
SCAN CON. TABLE FOR EACH CONTROLLER
SELECTED?
NO
YES
BUSY?
NO
YES
INITIAL?
NO
YES
HANDLER: DEVSTART
IS IT AN EVEN CYCLE?
NO
YES
SELECT MAP AND NEW REG. BLOCK
UPDATE REAL TIME
EXIT VIA PULLS 3
EXIT VIA START RU TO BUFFER TESTER
X WAIT
X ERROR
X MANUAL
NO
YES
SOURCE, KEYING
Y
N
PHASE 2 OR 3?
N
Y
NOBUF?
Y
NO
BURST DEV?
NO
YES
XWAIT?
N
Y
BON?
NO
YES
WAIT EXPIRED?
Y
N
HANDLER: DEVNOBUF
HANDLER: DEVRSTRT
DEVWAIT
DEC. FAULT TIMER TIME ≤ 0?
NO
YES
ENTER HANDLER AT DEVRETRY
RTH3
DONE?
NO, GET NEXT CONT
YES

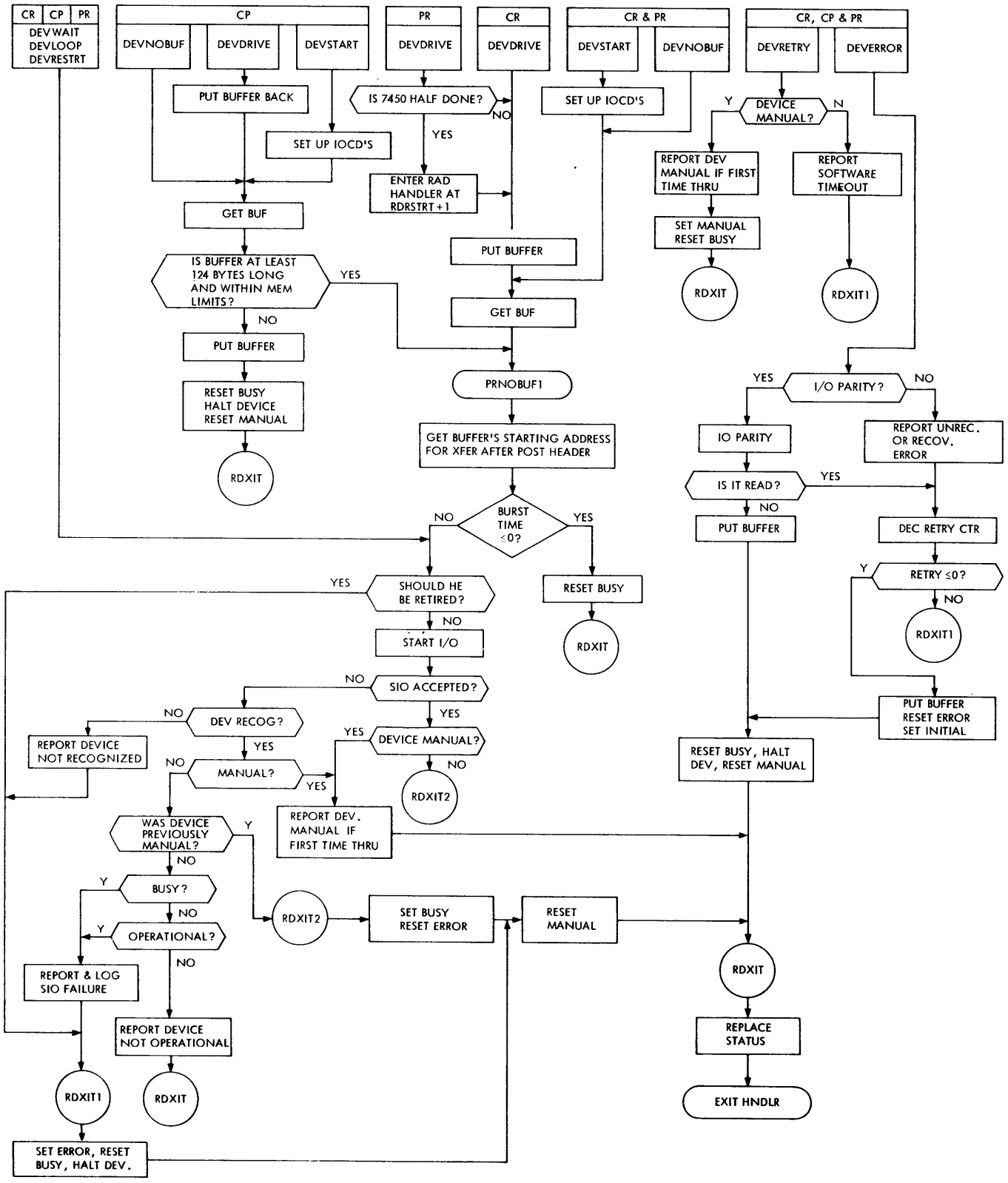
3.5.4.1 Control Table Scan

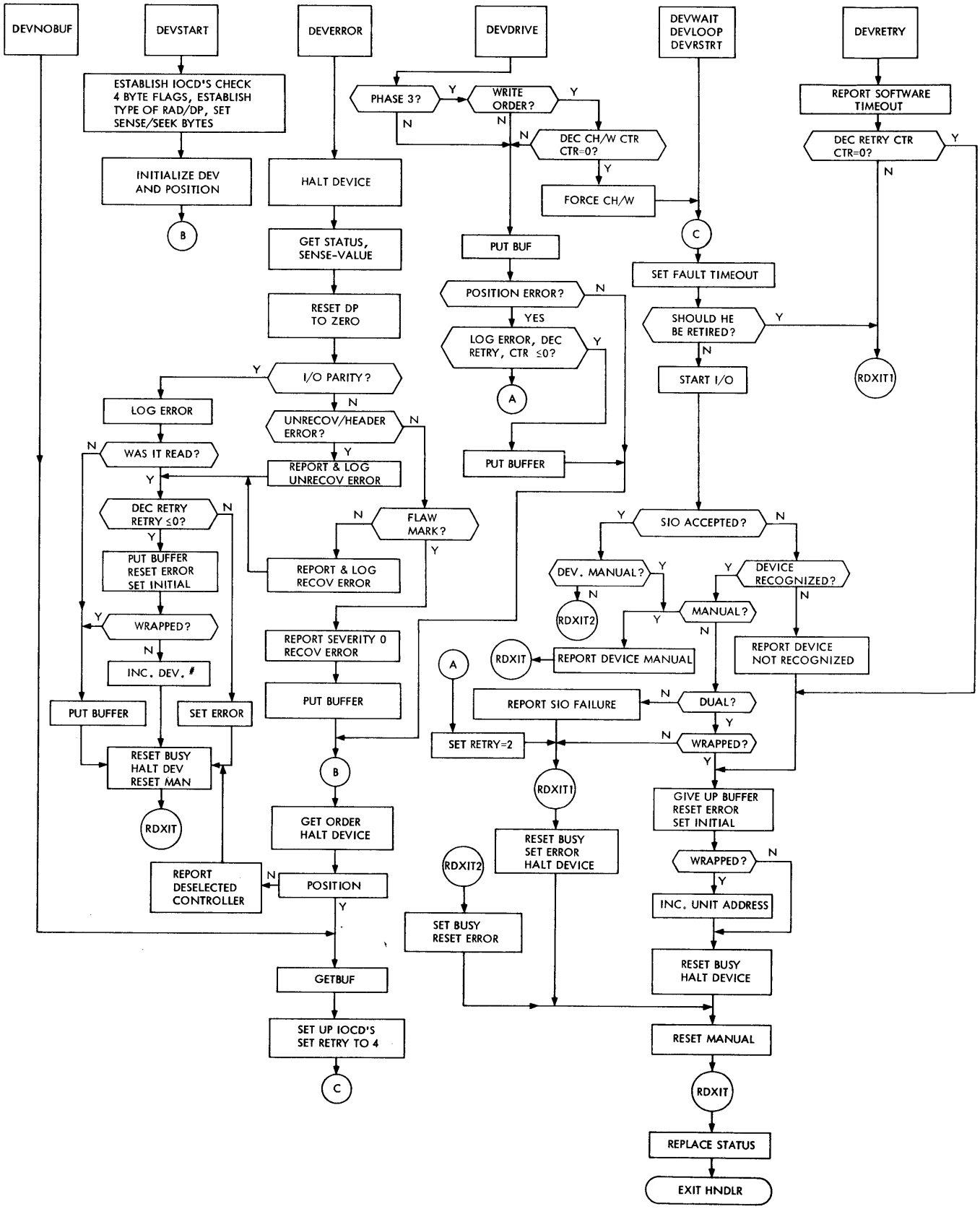


3.5.4.2 I/O INTERRUPT

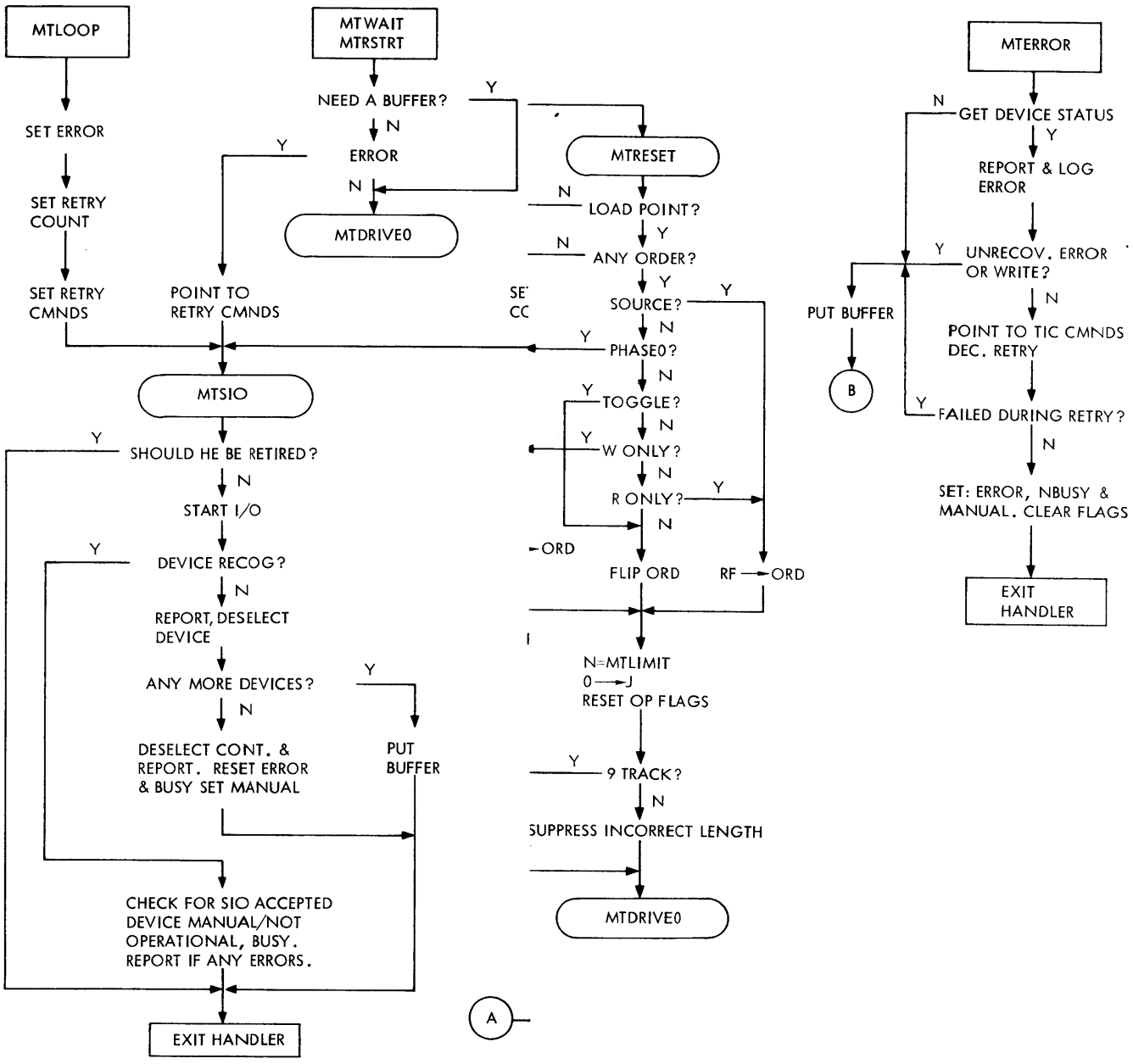


CR, CP & PR HANDLERS



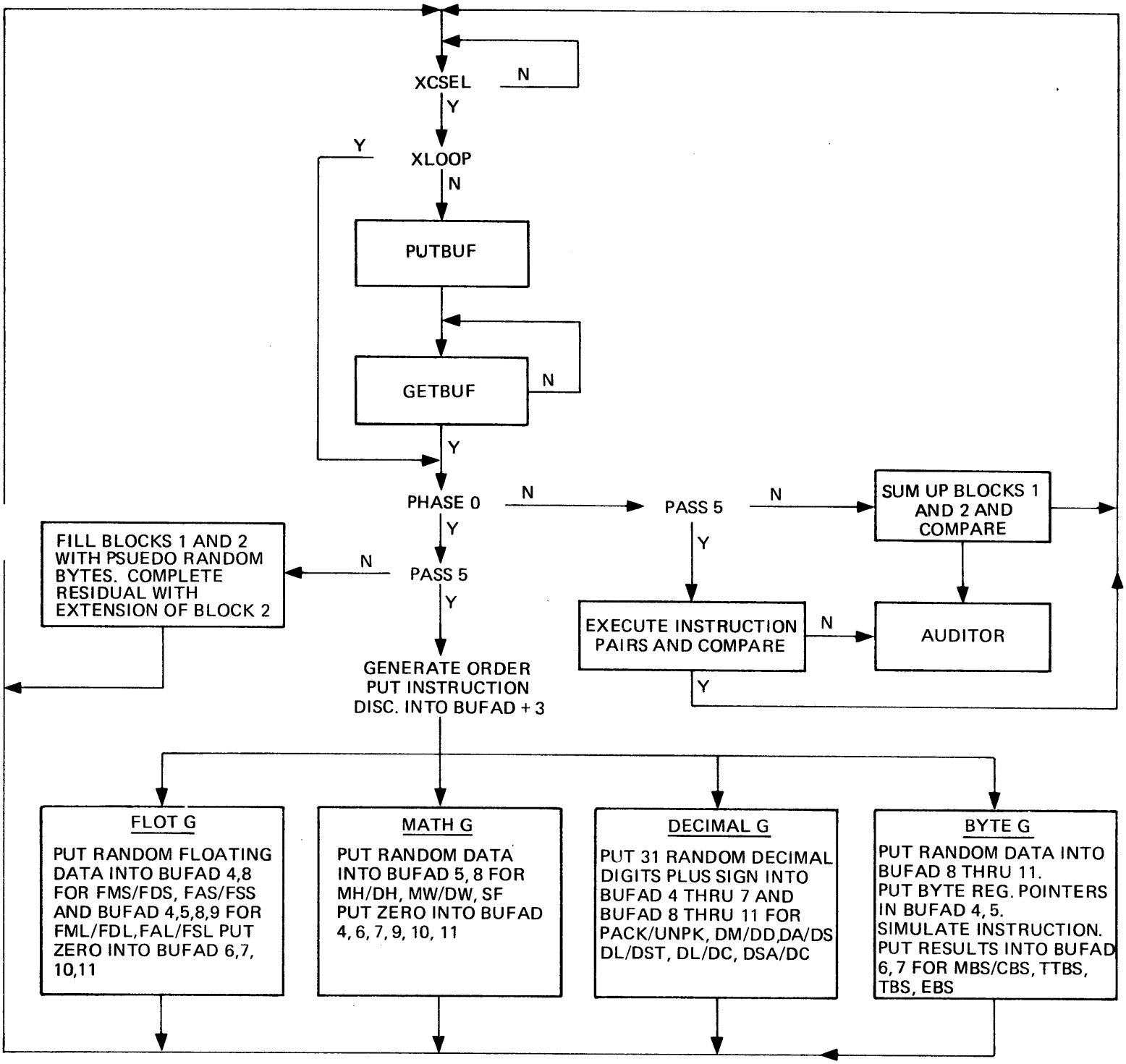

DEVNOBUF
DEVSTART
ESTABLISH IOCD'S CHECK 4 BYTE FLAGS, ESTABLISH TYPE OF RAD/DP, SET SENSE/SEEK BYTES
INITIALIZE DEV AND POSITION
B
DEVERROR
HALT DEVICE
GET STATUS, SENSE-VALUE
RESET DP TO ZERO
I/O PARITY?
Y
N
LOG ERROR
UNRECOV/HEADER ERROR?
N
Y
WAS IT READ?
N
Y
REPORT & LOG UNRECOV ERROR
DEC RETRY RETRY ≤0?
N
Y
FLAW MARK?
N
Y
PUT BUFFER RESET ERROR SET INITIAL
REPORT & LOG RECOV ERROR
WRAPPED?
Y
N
REPORT SEVERITY 0 RECOV ERROR
INC. DEV. #
PUT BUFFER
PUT BUFFER
SET ERROR
RESET BUSY HALT DEV RESET MAN
RDXIT
B
GET ORDER HALT DEVICE
REPORT DESELECTED CONTROLLER
POSITION
N
Y
GETBUF
SET UP IOCD'S SET RETRY TO 4
C
DEVDRIVE
PHASE 3?
Y
N
WRITE ORDER?
Y
N
DEC CH/W CTR CTR=0?
N
Y
FORCE CH/W
PUT BUF
POSITION ERROR?
N
YES
LOG ERROR, DEC RETRY, CTR ≤0?
Y
A
PUT BUFFER
DEVWAIT
DEVLOOP
DEVRSTRT
C
SET FAULT TIMEOUT
SHOULD HE BE RETIRED?
Y
N
START I/O
SIO ACCEPTED?
Y
N
DEV. MANUAL?
Y
N
RDXIT2
DEVICE RECOGNIZED?
Y
N
MANUAL?
Y
N
REPORT DEVICE MANUAL
RDXIT
A
REPORT SIO FAILURE
N
DUAL?
Y
N
WRAPPED?
Y
SET RETRY=2
RDXIT1
RESET BUSY SET ERROR HALT DEVICE
RDXIT2
SET BUSY RESET ERROR
REPORT DEVICE NOT RECOGNIZED
GIVE UP BUFFER RESET ERROR SET INITIAL
WRAPPED?
N
Y
INC. UNIT ADDRESS
RESET BUSY HALT DEVICE
RESET MANUAL
RDXIT
REPLACE STATUS
EXIT HNDLR
DEVRETRY
REPORT SOFTWARE TIMEOUT
DEC RETRY CTR CTR=0?
Y
N
RDXIT1

3.5.4.2.3


MTLOOP
SET ERROR
SET RETRY COUNT
SET RETRY CMNDS
MTWAIT MTRSTRT
NEED A BUFFER?
Y
N
ERROR
Y
N
MTDRIVE0
POINT TO RETRY CMNDS
MTRESET
N
LOAD POINT?
Y
N
ANY ORDER?
Y
SOURCE?
Y
N
Y
PHASE0?
N
Y
TOGGLE?
N
Y
W ONLY?
N
R ONLY?
Y
N
ORD
FLIP ORD
RF → ORD
N=MTLIMIT
0 → J
RESET OP FLAGS
Y
9 TRACK?
N
SUPPRESS INCORRECT LENGTH
MTDRIVE0
MTSIO
Y
SHOULD HE BE RETIRED?
N
START I/O
Y
DEVICE RECOG?
N
REPORT, DESELECT DEVICE
Y
ANY MORE DEVICES?
N
DESELECT CONT. & REPORT. RESET ERROR & BUSY SET MANUAL
PUT BUFFER
CHECK FOR SIO ACCEPTED DEVICE MANUAL/NOT OPERATIONAL, BUSY. REPORT IF ANY ERRORS.
EXIT HANDLER
A
MTERROR
N
GET DEVICE STATUS
Y
REPORT & LOG ERROR
Y
UNRECOV. ERROR OR WRITE?
N
PUT BUFFER
B
POINT TO TIC CMNDS DEC. RETRY
Y
FAILED DURING RETRY?
N
SET: ERROR, NBUSY & MANUAL. CLEAR FLAGS
EXIT HANDLER

3.5.4.3 CPU Test Module

Data Checker



SIGMA 5/6/7/8/9 SYSTEMS EXERCISER

PASS	BLKSIZE (BYTES)	BOUNDARY
0	2048	Page
1	16	Word
2	Max (65K Limit)	Byte 1
3	Random (16 To 2048)	Byte 2
4	Random (2048 To Max)	Byte 3
5	48 (Instruction Test)	DBL Word
6	6144 (Surface Test)	Page
7	Manual	

Phase 0 source only
Phase 1 = key all multi unit cont.
Phase 2 = sequential positions, initial start of unidirectional devices
Phase 3 = random positions
Odd cycles also exercise MAP, EXT. INT. and additional reg. blocks.

ERROR LOG OUTPUT

SEQ = Errs seq. nr
IDENT = A 5 letter code - SIDCC
S = Severity level
ID = Trap or int
CC = Sub code or cond. codes
ISSBMICO = The is, should be, modifier and constant of a data error
ERRADD = BA of error
BUFADD = BA of buffer
IN_1 IN_2 = Failing inst pair
Status 0-3 = Inst, eff add, group 8 and 9 on a wdt and snapped status 0-2 plus contents of parity loc on mem parity
Extended Status - WDT
Groups 0-3 plus 4 words of FAM (MIOPS) or groups 0-7 (RIOPS)
Groups 8-F (MIOPS) or groups 0-7 (RIOPS)
Instruction Errors
Orig regs (4) Orig mems (4)
Resl regs (4) Partial resl regs (4)
Byte String Inst 41 61 63
Orig regs (2) Expt mems (2) Orig mems (4)
Resl regs (2) Resl mems (2) Orig mems (4)
Byte String 40
Orig regs (2) Expt regs (2) Orig mems (4)
Resl regs (2) Resl regs (2) Orig mems (4)
Buffer Organization
Data Buffers (not pass 5)

ERROR LOG OUTPUT (Continued)

Header 5 or 12 bytes (function of BLKSIZ)
Reflection Fields = 2 identical data fields BLKSIZ-12/8 bytes in length
Residual = Whats left
Instruction Buffers
Header = 3 words
Instruction = 1
Reg (4 words)
Mem (4 words)
Header Organization
Post Header (not XFERRED) = 4 bytes
BS CTX UA ORD
BS = Buffer Status Byte
80 = Busy
10 = Checked
8 = Bad
CTX = Cont Tbl Index
UN = Unit Nr.
OP = Command
Post Header (XFERRED) = 1 or 8 bytes
PS BUFFAD
Seek (4 bytes)
PS = PARSHIZ
BUFFAD = BA of orig buf
SEEK = Seek Add

DIRECTIVES

Abstract - outputs explanation of SEX to TTY Q1≠0) or LP (Q1=0).
ABS, Q1 (CR)

AIO - reboots SEX from boot device specified by Q1
AIO (CR)

Boot - reboots and branches to MTL. Boot device specified by Q1
Boot,Q1 (CR)

Branch - transfers control to location specified by Q1.
BRA, Q1, (CR)

Compare - compares the contents of distination from Q1 to Q2 on the value specified in Q3 and outputs the location address and contents of every successful compare.
Compare, Destination, Q1, Q2, Q3 (CR)

Configur - reloads and enters the configurator.
CON (CR)

Display - outputs the contents of the specified destination from Q1 to Q2 on the TTY.
DIS, Destination, Q1, Q2 (CR)

DIRECTIVES (Continued)

Explain - outputs explanation of specified directive or destination to TTY (Q1#0) or LP (Q1=0).
EXP, Destination, Q1 (CR)

Errors - outputs sequences Q1 thru Q2 of idents Q3 to log device. If Q3=0, all idents are outputted.
ERR, Q1, Q2, Q3, (CR)

Halt - imposes halt mode (H>).
HA (CR)

HIO - halts and outputs the I/O status of all device addresses from Q1 to Q2.
HIO, Q1, Q2 (CR)

Read-Write-Read or write the device specified by Q1 and the position specified by Q2(Q2=-1 means use current position) to or from byte add specified by Q3(Q3=0 means use current buffer).
Read, Q1, Q2, Q3 (CR)
Write, Q1, Q2, Q3 (CR)

Print - outputs the contents of the specified destination from Q1 to Q2 on the LP.
P, Destination, Q1, Q2, Q3 (CR)

Redump - writes program or data to the base RAD and restarts the exercise. (Set SM=20)
Q1 specifies
The options:
0 Prog and Data
1 Prog
2 Data
3 Program Restart
RED, Q1, (CR)

Reload - reads configurator, program or data from the base RAD and restarts the exercise.
Q1 specifies the options:
0 Config and Prog and Data
1 Config and Prog
2 Config and Data
3 Config Only
4 Prog and Data
5 Prog Only
6 Data Only
7 Program Restart
REL, Q1 (CR)

Replace - displays then stores into the specified destination from Q1 to Q2.
REP, Destination, Q1, Q2, (CR)

DIRECTIVES (Continued)

Run - imposes run mode (R>)
RU (CR)

Select-Deselect-Select or deselect the devices specified by CTX in Q1 and unit in Q2 (bit 0 = unit 0, etc). Q3 = new contents of OPS (C:3) for select only
SEL, Q1, Q2, Q3, (CR)
DES, Q1, Q2, (CR)

Search - searches the destination from Q1 to Q2 on the value specified in Q3 and outputs the location address and contents of every successful search.
SEAR, Destination, Q1, Q2, Q3 (CR)

SIO - starts and outputs the I/O status of all device addresses from Q1 to Q2.
SIO, Q1, Q2 (CR)

Snap - causes the contents of Q2 to be output whenever location Q1 is accessed. If Q3 is negative, the LOC where the registers are save is outputted and halt mode will be imposed.
SN, Q1, Q2, Q3 (CR)

Spread - spreads the value Q3 into the specified destination from Q1 to Q2.
SPRE, Memory, Q1, Q2, Q3 (CR)

Start - starts the exerciser at pass and phase specified in Q1 and Q2 respectively.
STA, Q1,Q2 (CR)

Store - stores into the specified destination from Q1 to Q2.
STO, Destination, Q1, Q2 (CR)

Switch - switches the control device to observer and back. Q1 = line NR (80 for LOC TTY)
SW, Q1 (CR)

TDV - TIO - tests and outputs the I/O status of all device addresses from Q1 to Q2. Outputs to LP if Q3 $\neq$ 0.
TIO, Q1, Q2, Q3 (CR)
TDV, Q1, Q2, Q3 (CR)

Unsnap - removes the snap and restores normal operation.
UNS (CR)

CONTROL TABLE

CTX = Cont Tbl Index
C:1*Dev = Dev Add
C:2 Unit = Unit Select (8000 = Unit 0, etc)
C:3 OPS = Operation Selection
8000 = Selected
4000 = Loop
2000 = Sourcable
1000 = Already Keyed
800 = PM
100 = Burstable
80 = Overrunable
40 = Inhibit Buffer and Header Check
20 = Read Enable
10 = Write Enable
8 = Serial Unit Action
4 = Sequential Positions
2 = Must check buf after read
1 = Must check buf before write
C:4*CS = Cont Operational Status
8000 = Busy
2000 = Sourcing
1000 = Keying
800 = PM
100 = On Burst
80 = On Overrun Wait
40 = Surface is Keyed
20 = Needs a Buffer
10 = Dual Access
8 = Manual
4 = Errored
2 = Suspended 4 Byte
1 = Initialize
C:5*Seek = Seek Address
C:6*Buffer = Buffer Add
C:7*COMS = LOC of Command String
C:8*MAPS = LOC of Maps
C:9*OP = Order
C:10*CA = Cont Add
C:11 EC = Err Count
C:12 ETX = Element Tbl Index
C:13* REL = IOP Related CTX
CTX DEV Unit Ops CS Seek Buffer Coms Maps OP CA EC ETX REL

*Write Protected Table

COC TABLE

COC Line Status
00 = Line is Hung
03 = Lines are Answered
10 - 1F = Xmitting LOG on MSG
20-22 = Waiting for Escapes
24-27 = Xmitting on MSG
40 = Observer
80 = Control

ELEMENT TABLE

ETX = Element Table Index in HEX
E:1*MOD = Model Number in HEX
E:2*MN = Device Mnemonic in EBCDIC
E:3*HANDLR = Handler Address in HEX
E:4*RPX = Relative Parameter Table Index in HEX
E:5*PBW = Percentage of IOP Bandwidth used
ETX MOD MN HANDLR RPX PBW

MEMORY TABLE

Memory treated as word table. Q1 and Q2 are add limits

BYTE TABLE

Memory treated as a byte table. Q1 is a byte address (XXXXX.Y where XXXXX = Word Address and Y = Byte Position).

OPERATOR TABLE

O:0 MSG = MSG Device in HEX
O:1 CH = Chars per Line in Decimal for Con Device
O:2 LOG = Error MSG Device in HEX
*Write Protected Table

OPERATOR TABLE (Continued)

O:3 L = Error MSG Severity Level in HEX
O:4 H = Halt Severity
O:5 * TTY = Local TTY Add
O:6 * COC = COC Add
O:7 MX = Max NR of COC Lines
O:8 * CS = COC States
O:9 * CL = Remote Cont Line NR
O:10 CN = NR of Users
O:11 *Base = Base Dev Add
O:12 *SORS = Current Source Dev Add
O:13 *NR of Buffers Accessed by CPU
O:14 *NR of IO Ints

MSG CH LOG L H TTY COC MS CS CL CN BASE SORS

SYSTEM TABLE

S:0 BLKSIZ = NR of Bytes/Buffer
S:1 BUFSIZ = NR of Bytes Transferred/Buffer
S:2 FIRSTBUF = BA of First Buffer
S:3 STRTCORE = User Selected Low Core Limit
S:4 ENDCORE = User Selected Upper Core Limit
S:5 SM = System Control Modes
80 = Freeze Pass and Phase
40 = Inhibit Position Check
20 = Inhibit Table Protect
10 = Inhibit Error LOG
S:6 DS = Data Selection
80 = Math
40 = Float
20 = Decimal
10 = Byte
4 = Random
2 = Sequential
1 = Fixed
S:7 SO = Selected Order During Pass 5 Only

BLKSIZ BUFSIZ FIRSTBUF STRTCORE ENDCORE SM DS SO

*Write Protected Table

STATUS TABLE

S:8 CS = Cycle Selector
S:9 * CI = Cycle Indicator
S:10 PS = Pass Selector (80 = pass 0, etc)
S:11* PI = Pass Indicator
S:12 FS = Phase Selector
S:13*FI = Phase Indicator
S:14 BON = Burst On Time (secs)
S:15 BOF = Burst Off Time (secs)
S:16 RECL = MT Record Limit in Hex
S:18 MSTART = First Memory Buffer location to be tested
S:19 MEMSIZ = Size of Memory in Hex
S:20 REGOPS = Reg. Page selection by list
CS CI PS PI FS FI BON BOF RECL PASSTIME MSTART MEMSIZE REGOPS

REGISTER TABLE

Current Register Contents

Results of a TIO, TDV, HIO or SIO Inst. :
DEV = Device Address in HEX
CC = Resulting Condition Codes in HEX
CPDA = Command Pair Doubleword Address in HEX
ST and BC = Resulting Status and Byte Count
COM_1 and COM_2 = Command Pair
DEV CC CPDA ST BC COM_1 COM_2

TABLE CONTAINING CURRENT DATE AND TIME

MO = Month
DY = Day
YR = Year
HR = Hours
MN = Minutes
SC = Seconds
MO DY YR HR MN SC

*Write Protected Table

RELATIVE PARAMETER TABLE

RPX = Relative Parameter Table Index From Element Table (E:4)
BPS = Bytes Per Sector
HPT = Heads Per Track
SPT = Sectors Per Track
TPD = Tracks Per Device
DPC = Devices Per Controller
INC = Sector Increment for Current Buffer Size
BPD = Buffers Per Device for Current Buffer Size
RPX BPS HPT SPT TPD DPC INC BPD

IOP CONTROL TABLE TABLE INDEXED BY IOP

I:1 DOC = IOP on Data Overrun Control. (Any HEX Value)
I:2*LST = CTX of Last Guy on Wait
I:3*KWPS = Transfer Rate of IOP by KWS
I:4*PCNT = Percent of IOP Bandwidth
IOP DOC LST KWPS PCNT

*Write Protected Table

Staple

Staple

Fold

First Class
Permit No. 229
El Segundo,
California

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by

Xerox Corporation
701 South Aviation Boulevard
El Segundo, California 90245

Attn: Field Engineering Publications

Fold

XEROX

701 South Aviation Boulevard
El Segundo, California 90245
213 679-4511