Honeywell

# SYSTEM CONTROL UNIT

# WSCU66LA

# UNIT MANUAL

58009922

## DPS 8 (FREESTANDING UNIT)



58009922-040
DIST. CODE NCO, XAN

The following notice is provided in accordance with the United States Federal Communications Commission's (FCC) regulations.

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. As temporarily permitted by regulation it has not been tested for compliance with limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

A ISSUED
JULY 1981

| RECORD OF REVISION | | | |
|---|---|---|---|
| REV | DATE | AUTHORIZATION | PAGES AFFECTED |
| A | 7/81 | PHAFPS933 | 58009922-040,1F<br>58009922-014,1F<br>58009922-017,1F<br>58009922-015,1-1 & 1-2F<br>58009922-001,1F<br>58009922-051,2-1F<br>58009922-030,1F<br>58009922-100,3-1 thru 3-30F<br>58009922-200,4-1F |

| RECORD OF REVISION | | | |
|---|---|---|---|
| REV | DATE | AUTHORIZATION | PAGES AFFECTED |
| | | | |

# TABLE OF CONTENTS

## VOLUME 1F

# MANUAL CONTENTS

## VOLUME 1F

# MANUAL CONTENTS(CONTINUED)

## VOLUME 1F

# 1.0 GENERAL

## 1.1 INTRODUCTION

### 1.1.1 SCOPE

This manual is written primarily for the Field Engineering Division (FED) specialists and LISD Manufacturing specialists, and is, Honeywell Confidential and Proprietary. The CSU product maintenance documentation structure is contained in System Manual 58009888.

Unit manuals are sectionalized under six tabs and may consist of one to several volumes (binders) depending on the complexity of the functional unit. The information contained under each tab is as follows:

#### 1.1.1.1 General Tab

This section contains the introductory material.

#### 1.1.1.2 Physical Outline Tab

This section contains a physical outline of the functional unit and its major features, options and ORU's.

#### 1.1.1.3 Theory of Operation Tab

This section contains a multi-level description of unit operation keyed to major block, intermediate block and flow diagrams Coverage will give the specialist the information necessary to trace signals in the logic block diagram (LBD).

#### 1.1.1.4 Maintenance Aids Tab

This section contains references to existing test and diagnostics (T&D) procedures for the functional unit and appropriate trouble-shooting techniques. This material supplements the system maintenance software documentation.

#### 1.1.1.5 Wire Lists Tab(Not Required)

The Microfiche Index for the logic wire lists that are on microfiche in the microfiche card box plus pertinent hard-copy wire lists are located in 58009914, WMMU66LA Option Manual.

#### 1.1.1.6 CKT/Logic Tab(Not Required)

The LBD's for the circuit boards and/or the electrical schematics of power supplies and configuration, control and maintenance panels are located in 58009914, WMMU66LA Option Manual. It also contains the revision status sheets (RSS) for each circuit board and the circuit board component installation list (CIL), or installation instructions.

### 1.1.2 PURPOSE

This unit manual is applicable to multiple users and user environments as follows:

#### 1.1.2.1 Specialist Training

This manual can be used for preparing lesson plans, for classroom use by Field Engineering Training and for field study reference. It is expected that training will always be supported with additional prepared material (handouts, etc.) but the student should become familiar with and be able to effectively use this manual as a primary reference.

#### 1.1.2.2 FED Field Maintenance

This manual can be used for direct maintenance of equipment on customer sites to isolate and replace the optimum replaceable unit (ORU) and to resolve failures not corrected by ORU call-out and replacement.

#### 1.1.2.3 Technical Assistance Center

This manual can be used for remote support of field sites. It should be part of a primary reference library for use in the Technical Assistance Center (TAC) environment by TAC personnel in remote contact with a maintenance computer on site.

## 1.2 FEEDBACK

Send any comments on this manual to Honeywell Large Systems Product Support, P. O. Box 6000, MS K92, Phoenix, Arizona, 85005.

## 1.3 REFERENCE DOCUMENTATION

Option Installation Manual ---------- 58009914

## 2.0 PHYSICAL OUTLINE

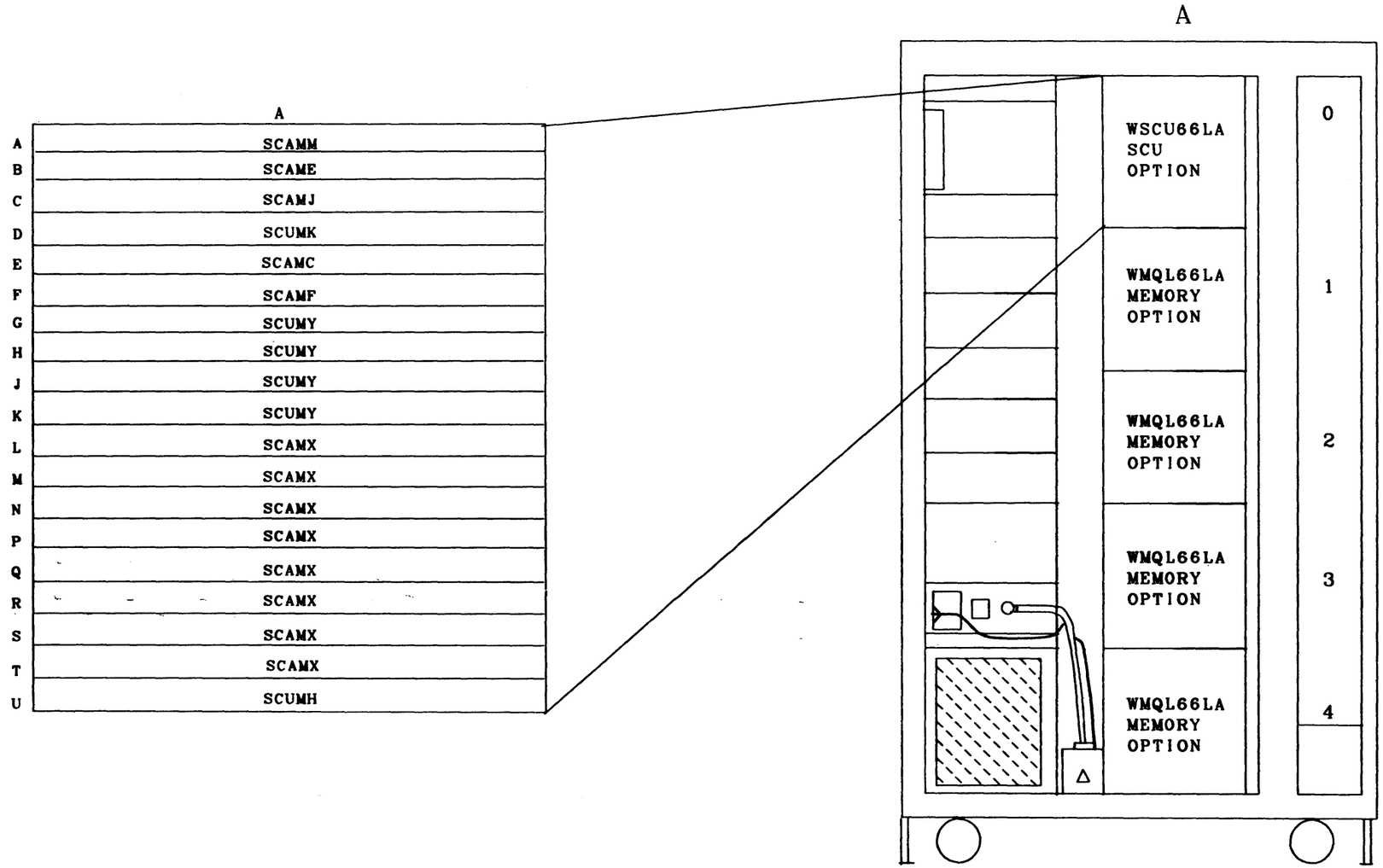



FIGURE 2-1. SYSTEM CONTROLLER WSCU66LA PHYSICAL OUTLINE

# 3.0 THEORY OF OPERATION

## 3.1 SYSTEM OVERVIEW

### 3.1.1 System Controller Functions

Functions of the SCU are as follows:

- Synchronize communications between active system modules.
- Provide the operating system with a centralized hardware time base.
- Service all valid requests from active system modules in a predetermined port priority scheme.
- Provide the necessary logic control to interface with as many as four attached storage units.
- Provide interlace capability between each pair of the attached store units.
- Provide the necessary maintenance logic to test the SCU and the attached store units.

### 3.1.2 Configuration Panel

The configuration panel of the SCU is located in the Main Memory Cabinet. The panel is utilized by the operator to configure the system. The panel provides switchable control of the following functions:

- Storage capacity for both storage units (A and B).
- On line/off line mode for store units.
- Interrupt cell mask assignment switches for mask A and B only.
- Lower store.
- Port enable.
- Interlace.
- Cyclic priority.
- Upper address boundary.
- Nonexistent address detection enable.
- Program/manual configuration mode.
- Initialize function

### 3.1.3 Interface Description

The 4MW SCU has two major interfaces to external system modules. The STORE UNIT INTERFACE is the link between the SCU and the storage units (maximum of four store units options allowed). The SYSTEM PORT INTERFACE is the link between the SCU and the active system modules (processors, input/output modules, etc.).

#### 3.1.3.1 System Port Interface

The SCU has 8 board locations dedicated to interfacing with active system modules. The board locations are dedicated to port option boards (MX). Depending on the customer's needs, varying quantities from two through eight MX boards may be added to the SCU. A basic system with a central processor and an input/output multiplexer would require two active port options (two MX boards and associated cable harnesses). A more complex system with multiple central processors, input/output multiplexers, etc., could utilize all port options (8 MX boards and associated cable harnesses).

It should be noted that the physical positioning of the port options determines the hard wired port priority. In the SCU, port 0 has the highest port priority while port 7 has the lowest port priority. Port priority simply means if two or more ports request a cycle from the SCU simultaneously, the higher priority port will be serviced first, followed by the next highest priority port requesting a cycle, etc. As a general rule of thumb, the slowest active modules (input/output multiplexers) are placed in the highest priority port locations while the fastest active modules (central processors) are placed in the lower priority port locations. With the exception of port priority, the port options are identical and may be installed in any other. (For a complete description of the active port interface, see paragraph 3.2.1.)

The System Port Board (SCAMX)

Series 60 Level 66, Level 68, and 6000 active system modules can interface with the System Controller using the system port options listed below. The active module must contain an active system port for each SCU connection.

FREE STANDING SCU

| Location | Port | Board |
|---|---|---|
| A0L | 0 | SCAMX |
| A0M | 1 | SCAMX |
| A0N | 2 | SCAMX |
| A0P | 3 | SCAMX |
| A0Q | 4 | SCAMX |
| A0R | 5 | SCAMX |
| A0S | 6 | SCAMX |
| A0T | 7 | SCAMX |

3.1.3.2 Store Port Interface

The system controller provides connection to as many as four WMQL66LA Memory Modules. When two modules (store units) (or two pairs of store units) of equal size are connected, the SCU provides for interlacing of addresses and overlapping of cycles in the two (pairs of) units.

For a complete description of the Store Port Interface see paragraph 3.2.2.

The Store Port Board (SCUMY)

| Location | Store Number |
|---|---|
| A0G | A |
| A0H | A1 |
| A0J | B |
| A0K | B1 |

3.1.3.3 SCU Logic Boards

The following paragraphs present a brief description of the functions of the various logic boards used in the SCU, including the Hardware Cache Clear (Write Notification) Option, WHCC66LB.

SCAMM (AØA)

This is the early cycle timing board which performs the following logic functions:

- Port priority control.
- Poll for system interrupts.
- Develop early cycle strobes.
- Detect illegal commands from system ports.
- Stop on illegal action.
- Cyclic priority.

- Micro Processor logic for the following functions:

  A. Interupt control for ports 0-8.
  B. Read address logic for bite 02-23.
  C. Read ØRCV ports 0-8.
  D. ZAC logic and control.

SCAMC (AØE)

This is the configuration register board which contains the following logic/performs the following functions:

- Configuration register.
- Configuration register mask A.
- Configuration register mask B.
- Configuration register port mask.
- Configuration register store control.
- Mask port decode.
- Mask display logic.
- Mask select.
- Reset mask select.
- Write data drivers.

- Microprocessor logic for the following functions:

  A. Microprocessor chip 8085A, address/data base control logic.
  B. USART chips for Remote and Local communication interface logic.
  C. Program interrupt timer chip.
  D. 8K RAM memory and 12K EPROM memory.
  E. Read logic for Read Data bite 0-17 and write data drivers for bits 0-17.

SCUMK (AØD)

This board contains the interrupt cells and mask registers. The SCUMK is a high density board that was designed for the low profile equipment. The board contains the following logic/performs the following functions:

- Execute interrupt cells.
- Interrupt cell mask registers.
- Interrupt cell enable logic.
- Execute interrupt (XIP) logic.
- Interrupt cell priority.
- Execute address generation.
- RMSK and RGR display drivers.

SCAME (AØB)

This is the look ahead logic board. It performs store select tasks for all models of the SCU. The SCAME board contains the following logic/performs the following functions:

- Address look ahead.
- Store selection.
- Configuration distribution.
- Illegal action register.
- Illegal action encode.
- Illegal action decode and stop logic.

- Microprocessor logic for the following functions:

  A, Port 8 look ahead logic.
  B. Read illegal action information.

SCAMF (AØF)

This is the clock and history register board for all models of the SCU. It contains the calendar clock, history registers, and the control logic.

- Microprocessor logic for the following functions:

  A. Read the read-data bits 18-71.
  B. Write the write-data bits 00-71.
  C. Address counter and logic bits 02-23.
  D. Read the write-data bits 18-71.
  E. Select logic for writing address as data or data-bits 00-71.
  F. Parity logic for write-data bits 00-71.

SCAMJ (AØC)

This is the late cycle control board which contains the following logic/performs the following functions:

- Logic initialize for SCU.
- Connect ($CON) to system ports.
- $ANS and $EOC processing.
- Generates Data Available ($DA) strobe.
- Store A and store B control.
- Early cycle and late cycle counters.
- History register (partial) and controls.
- Mode register.
- Display drivers.

- Microprocessor logic for the following functions:

  A. 16K EPROM memory.
  B. $DA logic.
  C. History register and display logic.
  D. Stop on IA control logic.

SCUMH (AØU)

This board is part of the write notification (hardware cache clear) option. The function of the cache clear option is to inform the CPU(s) when the contents of a location in main memory is changed. The CPU initiating the change is exempt from notification.

SCAMX (AØL-AØT)

This is the system port board which serves as the interface between the active system modules (CPU, IOM, etc.) and the SCU. The quantity of SCAMX boards in any SCU is a function of the number of active modules in the particular system. The board contains the following logic/performs the following functions:

- Port control.
- Parity check.
- Generates receive control (ØRCV).
- Generates transmit ($XMT) strobe.
- Generates zone addresses.
- Transfers commands and addresses to memory.
- Transfers illegal action codes to active modules.
- Transfers data between active modules and SCUMY board.

- Microprocessor logic to disable MX board.

SCUMY (AØG-AØK)

This is the memory port board which serves as the interface between the main memory and the SCU. The quantity of SCUMY boards in any SCU is a function of the quantity of main memory modules in the particular system. The board contains the following logic/performs the following functions:

- Transfers control signals between the memory modules and the SCU.
- Decodes memory commands.
- Stores and transfers ZAC information to the memory modules.
- Transfers data between the active modules and the memory modules.

## 3.2 GENERAL

The System Control Unit (SCU) has two basic functions:

1. Providing priority-controlled memory access for information storage and retrieval functions for the active system modules.

2. Providing intermodule communications paths for control of hardware operations within the computer.

Figure 3-1 presents the major block diagram for the Four Megaword (4MW) SCU.

The SCU contains the port interfaces which enable communication between the system modules and the storage unit, and intercommunication between system modules. Each cycle begins when a system module issues an interrupt strobe to the SCU via one of the active port interfaces.

The SCU also contains store interfaces which act as communications links between the SCU and the storage units.

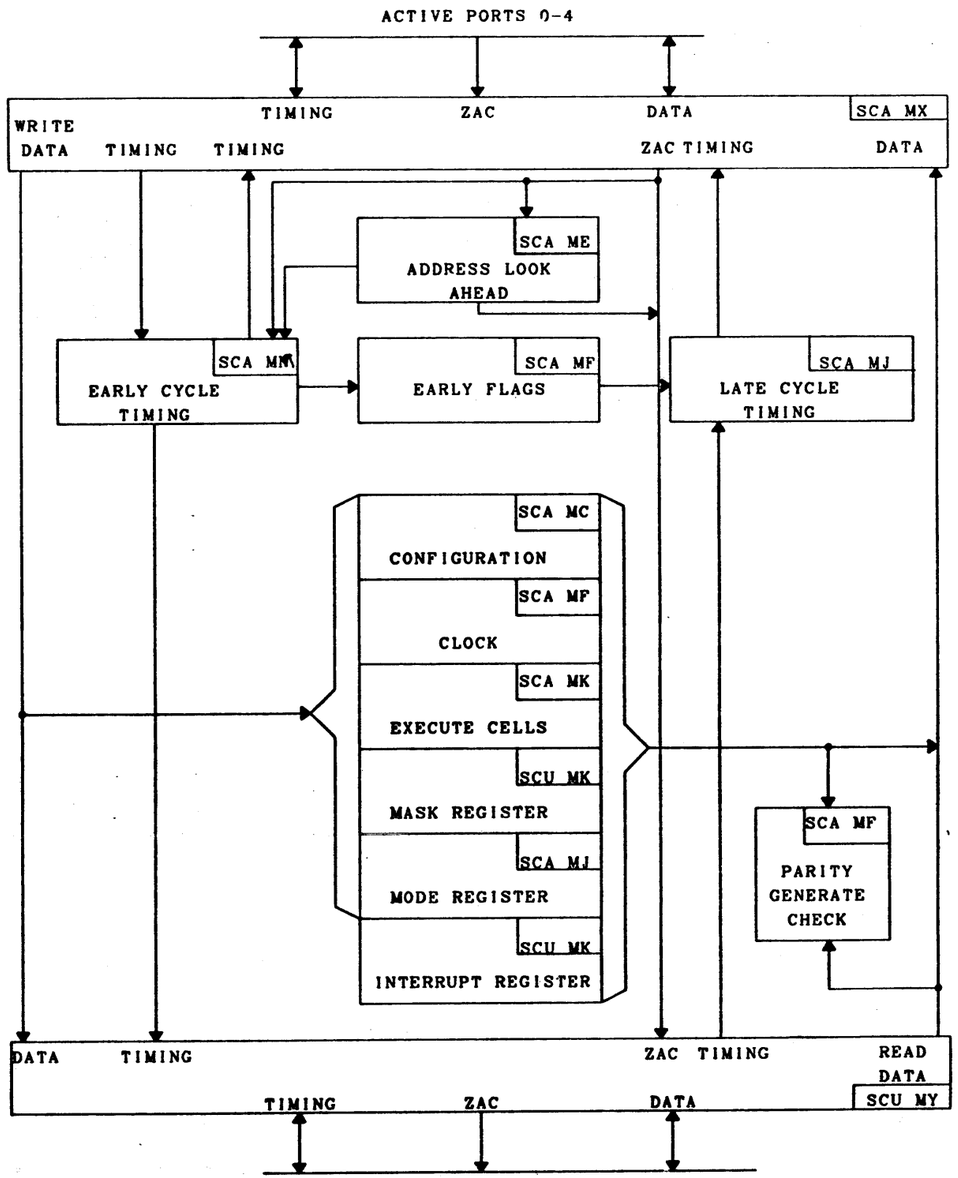



**Figure 3-1 4MW SCU MAJOR BLOCK DIAGRAM**

3.2.1 Port Interfaces

The port interfaces, Figure 3-2, provide the necessary hardware and logic to connect the Communications Processor Unit (CPU), the Input/Output Multiplexer (IOM), and the other active modules to the SCU.

Data at the port interfaces is 72 bits wide plus two parity bits. As with all system modules, the parity is odd. Bit 00 of the data is most significant and bit 71 is least significant. Although the data bus is bidirectional, the parity lines are unidirectional.

ZAC lines (8 Zone, 22 Address, 5 Command) are unidirectional and carry information from the requesting active module to the SCU with each cycle request.

Timing and control signals include the port interface control necessary for a complete SCU cycle, plus various non-cycle oriented controls. The cycle controls include the $INT (request for cycle), $PIN (request honored, cycle in progress), $DA (data available), and the illegal action lines from the SCU indicating status of the just-completed cycle.

In addition to normal cycle controls, others include the Connect Strobe ($CON) and the Execute Interrupt Present (XIP) to enable intercommunication between active units. To register data from an active unit to the SCU, the active unit issues a Strobe Transmit ($XMT-AP) pulse. To register data from the SCU to an active unit, the SCU issues a Strobe Transmit ($XMT-PA) pulse.

The SCU is also the distribution center for the System Initialize (INZ) signal. This signal may be issued by any active unit and is distributed by the SCU to all enabled active units attached to the interface.

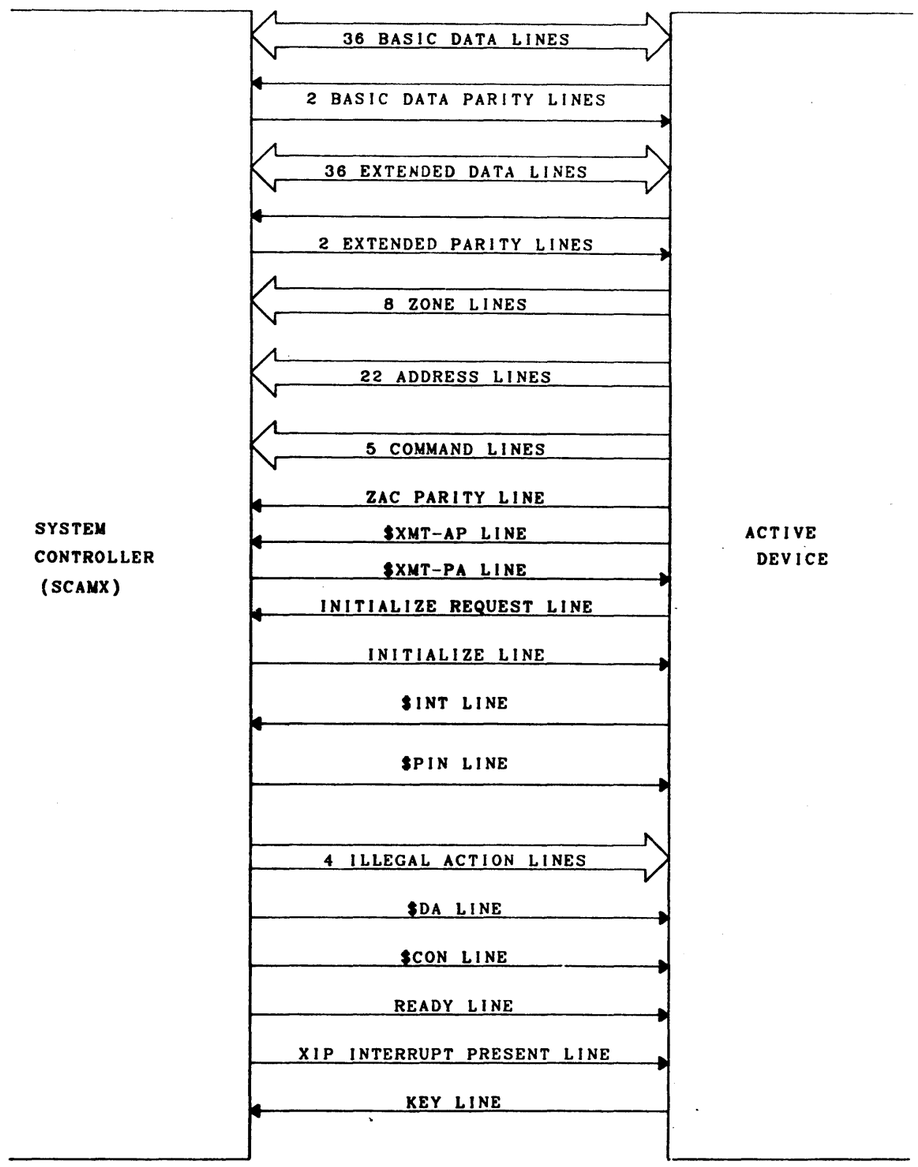



**FIGURE 3-2 SYSTEM PORT INTERFACE**

The Ready Line originates with the SCU. It is sent to the attached active units to notify them that the SCU and the storage unit are available for system access.

3.2.2 Store Interfaces

The store interfaces (Figure 3-3) contain the necessary hardware and logic to accomplish interface between the SCU and the storage unit(s).

Data and ZAC exchanges between the SCU and the storage units are the same as between the active units and the SCU. Refer to paragraph 3.2.1 for details.

Timing and control lines at the store interface differ slightly from the port interface. The cycle controls include the following:

1. Storage unit start pulse ($TS) - similar to $INT.
2. Storage unit answer pulse ($ANS) - similar to $PIN.
3. Storage unit end of cycle pulse ($EOC) similar to $DA.
4. Illegal action lines from the storage unit, indicating status of the just-completed cycle.

In addition to these control signals, four command strobes are issued by the SCU to the storage unit. These command pulses inform the storage unit what type of cycle to perform.

1. $RRS - Read Restore
2. $RCL - Read and Clear
3. $CW1 - Clear Write Zone (Single Precision)
4. $CW2 - Clear Write Double Precision

The storage units will also inform the SCU, via the control lines, of power and temperature failures.

3.2.3 SCU Cycles

To start a cycle, an active module (CPU, IOM, etc.) gates the ZAC lines and then transmits an Asynchronous Interrupt Strobe ($INT) to the SCU. This $INT strobe, requesting a cycle, initiates the following sequence of events. For a complete early cycle, refer to Figure 3-4.

1. SCU uses the look ahead bits of the address to ensure that the memory storage unit may be accessed by the active module.
2. SCU receives an $INT strobe from an active module requesting access to a memory unit.
3. SCU stores the strobe on an FREQUEST bistable (FREQ).
4. SCU checks priority of the requesting active module. If the requesting port has the highest priority, the SCU gates the data and ZAC onto the data bus.

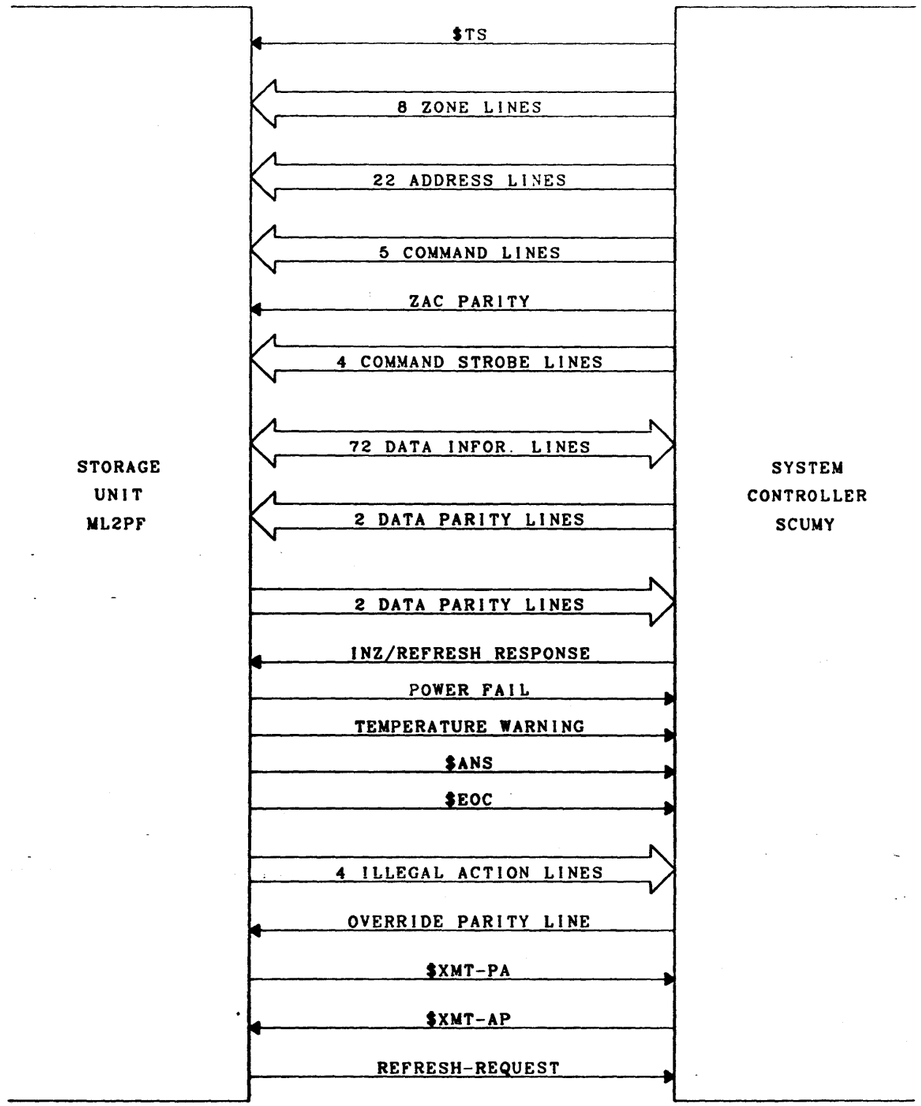



**FIGURE 3-3 STORE PORT INTERFACE**

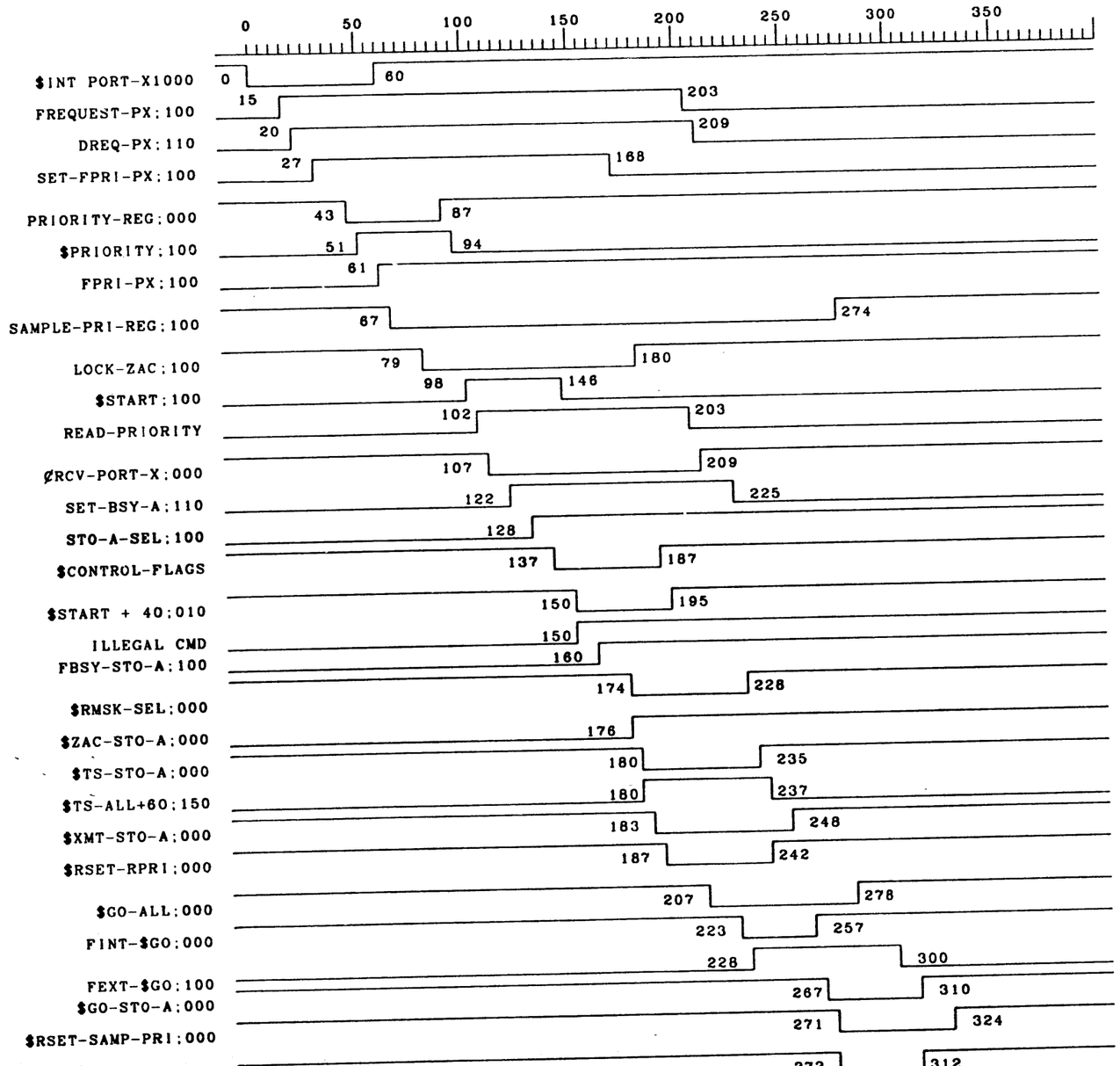



**FIGURE 3-4**

**EARLY CYCLE TIMING**

5. SCU performs a parity check on the zone, address, and command (ZAC) lines. (SCUMX board)

6. SCU sets an error flag (ZPE) if an error is detected. This sets the abort bistable which inhibits the command strobe to the store port.

7. SCU sends a Control strobe ($PIN) to the active module as notification that the requested cycle is being processed.

8. SCU accesses the memory storage unit by generating a $TS pulse.

9. SCU decodes the zone, address, and command (ZAC) lines to determine the type of cycle the active module has requested.

10. SCU issues a command strobe to the memory storage unit. The strobe is determined by the type of cycle requested by the active module. The strobe may be aborted by an error.

11. The accessed memory storage unit sends an Answer Strobe ($ANS) to the SCU, verifying readiness to receive the data.

Triggering of the $ANS strobe by the memory storage unit signals the end of the early cycle and the beginning of the late cycle.

The sequence of events in the late cycle are as follows:

1. SCU enables the data paths as determined by controls set in the early cycle.

2. SCU enables the controls and issues a response according to the type of cycle requested by the active module.

3. SCU issues a Data Available ($DA) strobe to the memory storage unit, signalling transfer of data from the SCU.

4. Accessed memory storage unit issues an End of Cycle strobe ($EOC) to the SCU, indicating that the cycle is nearing completion.

5. Illegal action codes from the accessed memory storage unit and within the SCU are examined, encoded, and sent to the requesting active module.

## 3.3 FUNCTIONAL DESCRIPTION

### 3.3.1 Port Interface Signals

The system port interface consists of signal lines to/from the active module as shown in Figure 3-5. All communication between the active module and the SCU takes place over these lines. Subsequent paragraphs describe these lines and their basic timing relationships.

The signal lines are listed below. The designation "true" indicates the interface receives/sends a high-level (nominal =5 volts) when the signal is enabled. The designation "false" indicates the interface receives/sends a low level (0 volts) when the signal is enabled.

Basic Data Lines (36 bidirectional data lines, and two unidirectional parity lines (true = 1).

Extended Data Lines (36 bidirectional data lines, and two unidirectional parity lines (true = 1).

Zone Lines (8 unidirectional lines from active module) (true = 1)

Address Lines (22 unidirectional lines from active module) (true) = 1)

Command Lines (5 unidirectional lines from active module) (true = 1)

ZAC Parity (1 unidirectional line from active module) (true = 1)

Transmit Strobe ($XMT-AP) Line (1 unidirectional line from active module) (true)

Initialize Request Line (1 unidirectional line from active module) (false)

Access Request Strobe ($INT) Line (1 unidirectional line from active module) (false)

Acknowledge Strobe ($PIN) Line (1 unidirectional line from SCU) (false)

Key Line (1 unidirectional line from active module) (true)

Illegal Action Lines (4 unidirectional lines from SCU) (true = 1)

Data Available Strobe ($DA) Line (1 unidirectional line from SCU) (false)

Connect Strobe ($CON) Line (1 unidirectional line from SCU) (false)

Transmit Strobe ($XMT-PA) Line (1 unidirectional line from SCU) (true)

Initialize (INZ) (1 unidirectional line from SCU) (false)

Ready (RDY) (1 unidirectional line from SCU) (false)

Interrupt Present (XIP) Line (1 unidirectional line from SCU) (false)

#### 3.3.1.1 Basic Data and Extended Data

The 38 basic data lines (36 bidirectional data lines and 2 unidirectional parity lines) plus the 38 extended data lines (36 bidirectional data lines and 2 unidirectional parity lines) make up a 74 line data path for transfer of data with parity to/from the SCU. The sense of parity is odd with bit 72 as the basic data parity bit and bit 73 as the extended data parity bit.

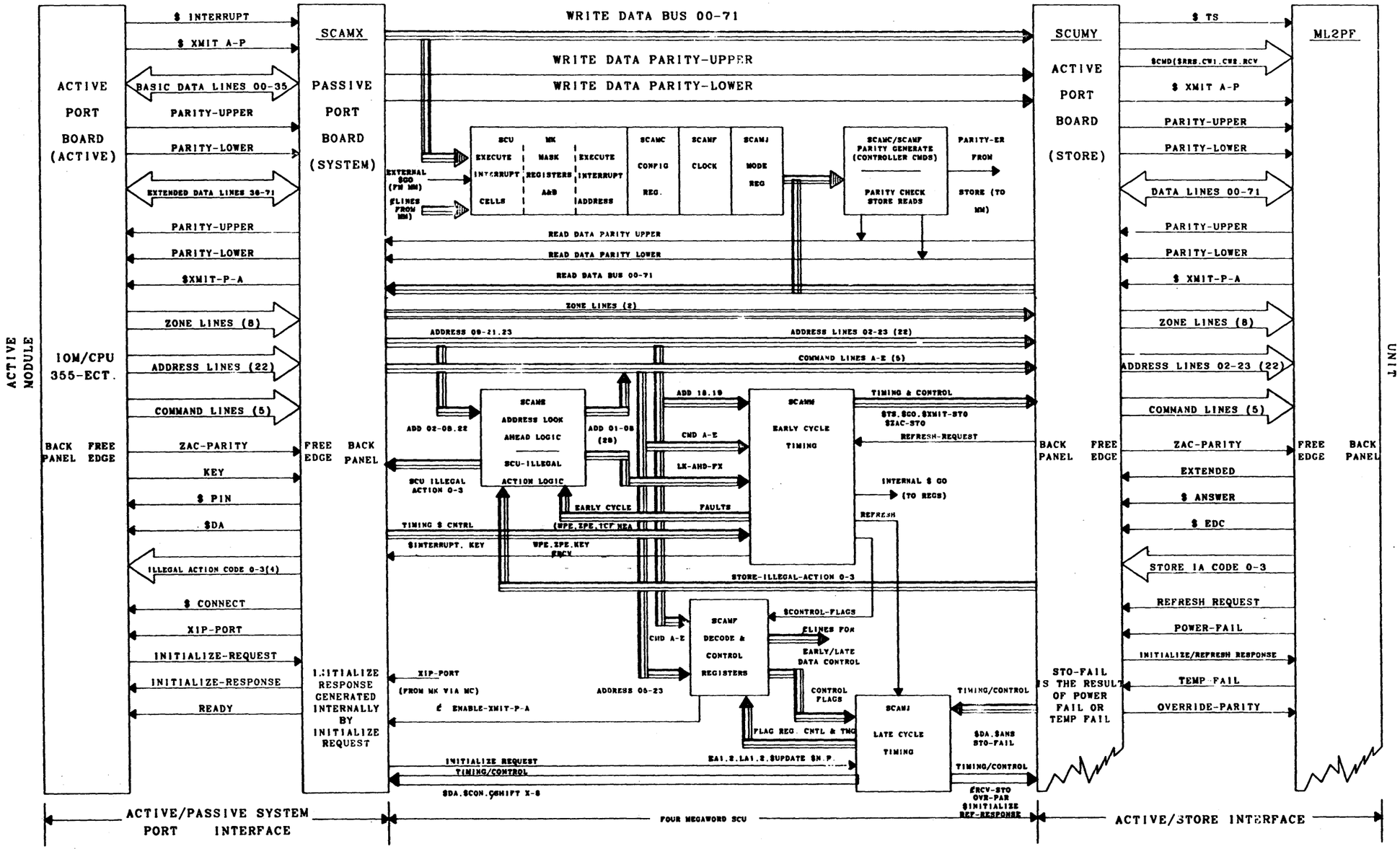



FIGURE 3-5. INFORMATION FLOW 4 MEGAWORD SCU

All single precision operations will utilize the basic data lines. During single precision operations, the extended data lines and extended parity line are ignored.

3.3.1.2 Zone Lines

The (8) unidirectional zone lines are used to specify which 6- or 9-bit characters are to be operated on during a single precision write operation (clear write). The store unit will ignore the zone lines for all commands except single precision write.

The following table shows the zone assignments:

| Data Bits Affected | Field Size Character | Field Size Byte | Zone | Line |
|---|---|---|---|---|
| 00-05 | 6 | 9 | ZO | Zone Zero |
| 06-08 | 6 | | Z1U | Zone One Upper |
| 09-11 | | | Z1L | Zone One Lower |
| 12-17 | 6 | 9 | Z2 | Zone Two |
| 18-23 | 6 | 9 | Z3 | Zone Three |
| 24-26 | 6 | | Z4U | Zone Four Upper |
| 27-29 | | | Z4L | Zone Four Lower |
| 30-35 | 6 | 9 | Z5 | Zone Five |

3.3.1.3 Address Lines

The 22 unidirectional address lines (A2-A23) are used to specify the location in which a store module access is to occur or, in the case of internal SCU cycles, the address of an internal register. Address line (A2) is the most significant address bit and line (A23) is the least significant address bit.

3.3.1.4 Command Lines

The (5) unidirectional command lines are used to specify the command to be executed when the access request ($INT) pulse from the active module is acknowledged. Illegal combinations of command lines are trapped and the active module notified via the illegal action lines.

3.3.1.5 ZAC Parity

The active module generates odd parity for the combined zone, address and command (ZAC) lines and furnishes this parity to the system port.

3.3.1.6 Transmit Strobe ($XMT-AP)

The transmit strobe ($XMT-AP) is utilized by the active module to control the strobing of data into the SCU port board receive register.

3.3.1.7 Initialize Request

The system port accepts the initialize request signal from the active module (usually originated at the IOM or system console). This signal is accepted by the SCU and distributed through other system ports via the initialize line.

3.3.1.8 Access Request Strobe ($INT)

The access request strobe ($INT) is used by the active module to initiate an SCU command.

3.3.1.9 Acknowledge Strobe ($PIN)

The acknowledge strobe ($PIN) notifies the active module that the requested cycle is in progress. The time period (at the system port) from $INT to $PIN is a function of whether the SCU is currently busy servicing another port, and the priority of the port that is requesting access.

3.3.1.10 Illegal Action Lines

The system port provides four unidirectional lines to the active module coded to provide illegal action information. These four lines are valid following the transfer of the data. Parity is not generated on the illegal action lines.

3.3.1.11 Data Available Strobe ($DA)

The data available strobe ($DA) is the pulse issued by the SCU indicating that the requested data will be valid in the active module port board a short time later. The active module receives $DA pulse for all SCU cycles (write or read) with $DA preceding the illegal action lines. For write cycles, $DA is effectively and End of Cycle pulse since there is no data sent to the requesting active module.

3.3.1.12 Connect Strobe ($CON)

The connect strobe ($CON) is a pulse from the SCU to alert the active module of intermodule communication. $CON is the result of a connect command to the SCU and will be generated as a function of the $DA pulse of the connect command. The SCU sends $CON to the designated system port provided the port is enabled. $CON to a not enabled port will result in an illegal action to the system port originating the connect command.

3.3.1.13 Execute Interrupt Present (XIP)

The execute interrupt present (XIP) is a level from the SCU to an active module designated as a system control module indicating a program interrupt cell requires service. The SCU can designate up to two system ports as control ports via the program interrupt mask register assignment switches on the SCU configuration panel. Under program control, up to eight system ports can be designated as control ports.

3.3.1.14 Transmit Strobe ($XMT-PA)

The transmit strobe ($XMT-PA) is the signal used by the SCU to strobe data into the active module port board data receive register.

3.3.1.15 Initialize (INZ) Line

The Initialize (INZ) signal is a level from the SCU which causes all on-line active modules to go to the initialized state. The initialize signal occurs asynchronous of all system timing.

3.3.2 Data Flow

Data flow through the SCU is in two directions and is accomplished on two unique sets of lines.

1. Data coming from a requesting active module, and destined for either memory or one of the SCU's internal registers, is transferred via the write-data bus.

2. Data read from memory or one of the SCU's internal registers, and destined for a requesting active module, is transferred via the read-data bus.

3.3.2.1 Write Data Bus

Incoming data is transferred from the system ports to the store ports via a common bus referred to as the "write-data bus". Data is gated onto the write-data bus by the "receive control" signal ¢RCV. This control signal is generated by the port priority and timing logic which is located on the SCAMM board. Since this logic determines which requesting system port will be granted store access, only the data from one port will be present on the bus at any given time.

Data on the write-data bus is gated to the memory logic by a transmit strobe $XMT; XXX. This logic, located on the memory port boards, places the data on the interface line to the memory unit designated by the look ahead address. The contents of these lines is gated into the memory write register by the control logic. The write data is gated internally by the internal command decode logic and is placed on the memory data lines, but it is ignored because the SCU is executing a read data command.

3.3.2.2 Read Data Bus

Data read from memory is transferred to the requesting active module via the read data bus. The read data bus is organized in the same manner as the write data bus since a common bus is used to transfer data from the store ports and internal registers to the active system ports. The main difference is that the parity check logic for outgoing data is on the SCAMF board. This logic checks and generates parity on all outgoing data. Control logic for outgoing data is provided on the SCAMX board which formats data, as required by the command being executed, and places it out on the output data lines. It should be noted that the data on the read data bus is presented to the input of all system ports, but is only transmitted by the port that requested the data. Requesting port identification is obtained from the history register logic on the SCAMF board.

3.3.2.3 ZAC Bus

Zone, address, and Command information is carried on 35 unidirectional lines referred to as the ZAC bus.

- o Zone information is carried on 8 lines.
- o Address information is carried on 14 lines plus 8 lines for look ahead.
- o Command information is carried on 5 lines.
- o Parity information is carried on one line.

As shown in Figure 3-6, information on these lines flows from the requesting active module to the system port on the SCAMX board. From there it is routed to the SCAMM board where it is decoded and registered. The one exception is that information on the look ahead lines is routed to the SCAME board.

The zone, address, and command switches permit the information contained on 27 ZAC lines to be gated to the internal SCU logic, and to the SCUMY boards when ¢RCV for the port is detected.

The look ahead portion of the address is gated from the SCAMX board to Look Ahead Address Board (SCAME) by the look ahead address switch. These bits are used as an advance check to determine whether the request is directed at Store A or Store B.

The parity check logic looks for odd parity on the ZAC information. If an error is detected, it is reported to the SCU as a Port-ZAC-Error and a ZPE flag is set. Discovering the error, the SCAMM board aborts the command strobe to the store unit and the SCU's internal registers. The cycle is aborted by an illegal action code of 14-octal.

3.3.2.4 Parity

The system extensively checks transmission parity on intermodule data transfers. The SCU checks all data and ZAC parity received from the active modules as well as data parity received from the storage units, and generates an illegal action code if a parity error is detected. For data involved in reading the internal registers, the SCU generates the parity bit and passes it on to the requesting active module.

3.3.3 ZAC Logic Operation

The ZAC bus should be present at the input of the SCAMX board at least 10 nanoseconds prior to arrival of the $INT. $INT causes the SCU to initiate a new early cycle if it is not busy with another port. This is conditional on the previously sent look ahead address.

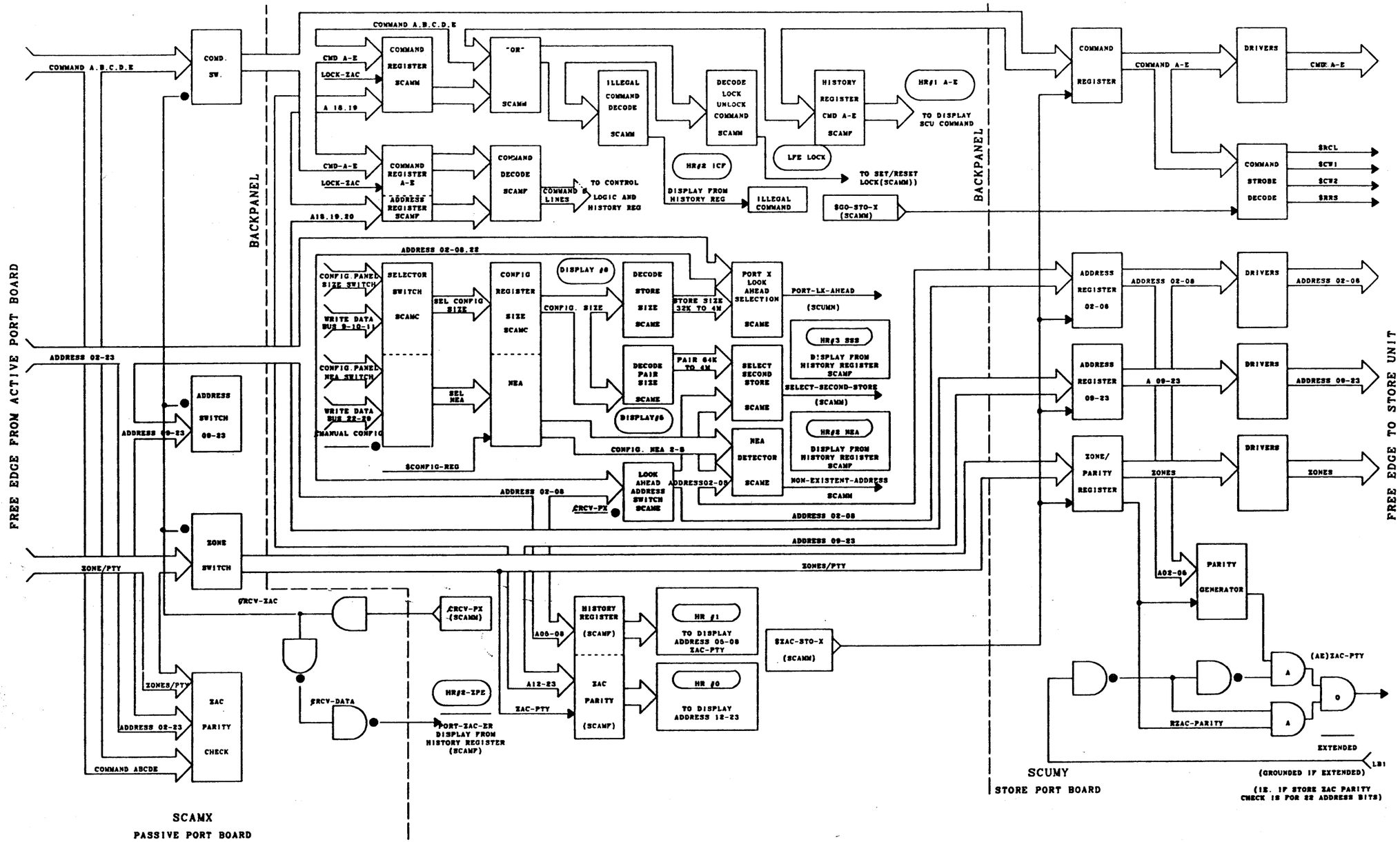



**FIGURE 3-6 FUNCTIONAL DIAGRAM ZAC FLOW**

In Figure 3-6, note that all ZAC lines are only enabled to the SCU's common ZAC bus when the CRCV for the port is received by the SCAMX port board. The look ahead bits are wired directly to the input of the SCAME look ahead address board. The SCAME board has one look ahead circuit for each of the system ports. The logic for each port is constantly comparing the look ahead bits with the decode of the store size. The store size is made up by decoding the store size field of the programmable configuration register. The output of the decoder is 7 lines, each representing one of the 8 possible sizes of the lower store unit. The configuration register is loaded three ways:

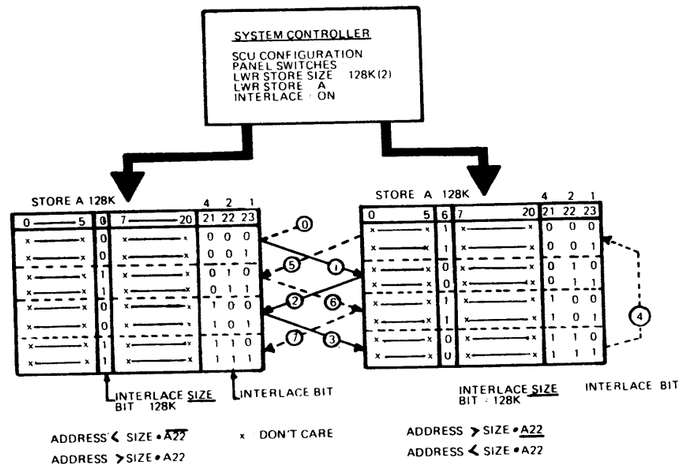
| Conf. Panel Mode Switch | Conf. Reg. Load Source | When Loaded |
|---|---|---|
| Manual | Conf. Pnl. Sws | Unconditionally |
| Program | Conf. Pnl. Sws | Sys. or Pnl INZ |
| Program | SGR-Data Word | SGR-Conf.Reg Cmd |

As can be seen from the above chart, when the unit is in manual mode or the unit is in program mode with the system or unit initialized, the configuration register SIZE field is loaded from the LWR-Store-Size switch on the configuration panel. System initialize will occur every time the system is bootloaded. So, for program or manual mode, the initial state of the configuration register is set to correspond with the switch settings. In the program mode, the register retains the information read from the switches until such time as a Set General Register Configuration Register is performed. At the present time, only the T&D programs use this command.

Address line 22 has to do with interlacing, which may or may not be enabled. Interlace is a method of forcing the SCU to address different store units (pairs) for every other sequential access to store. This permits full use of the overlap between the early and late cycles, thus increasing system thruput. For example, if the SCU has two 128K Store units attached as Store A and Store B, (see Figure 3-7), and an active module requests a cycle with all of the address look ahead lines in their zero state (false), the lower store unit (normally address 0-128K-1) is selected for the cycle. If the lower store unit is busy with a current cycle, the request is postponed until it becomes available for a request.

Taking another example, where the interlace function is enabled and the active module requests a cycle with all address look ahead lines in their zero (false) state except line 22 which is a one (true), the upper store unit (normally 128K-256K-1) is selected for the cycle. This is accomplished by decoding the store size from the size field of the configuration register. (See Figure 3-7) The interlace function is controlled by the configuration panel INTERLACE ON/OFF switch or the programmable function in the configuration register.

The address look ahead lines, in effect, interrogate the status (busy or not busy) of the store unit prior to letting the requested cycle proceed.




Rules for interlace:
1. Store A and B must be of equal size.
2. Store A and B must be of equal speed.

Numbers in circles indicate the interlace pattern, assuming the configuration given. The numbers 0 thru 3 show how the sequential addresses are threaded into both Store A and B dependent upon the Interlace bit 22. Note that at 4 the Size bit Address 6 goes to the ONE state. This assumes that all addresses from 0 to 128K-1 follow the same pattern of operation. When the SIZE bit determined by the LWR - STORE - SIZE configuration, goes to the ONE state the pattern reverses itself. Numbers 4 thru 7 show this effect. The SIZE bit corresponds to the following settings of the LWR - STORE - SIZE switch.

| Switch Seting | Size | Size Bit (Address Bit) |
|---|---|---|
| 0 | 32K | 08 |
| 1 | 64K | 07 |
| 2 | 128K | 06 |
| 3 | 256K | 05 |
| 4 | 512K | 04 |
| 5 | 1M | 03 |
| 6 | 2M | 02 |

**FIGURE 3-7 SYSTEM CONTROLLER TWO WORD INTERLACE**

The address look ahead lines are redriven from the SCAME address look ahead board to become part of the common ZAC bus to the store units. These lines are gated off the SCAME board with ¢RCV for the port.

The look ahead address, in addition to being gated to the input of the store port boards with the remaining portion of the ZAC bus, are also sent to the select second store and non-existent address logic. The select second store logic on the SCAME board compares the look ahead address with a decode of the pair size to determine which store unit of a pair is to be selected. This is done by developing seven unique pair size lines (Pair - 64K to Pair - 4M), one of which will be true. The pair size bit is then ANDed with an address line one binary power less than the pair size. If the AND condition is met, a check is made to determine if the store unit is on and, if so, the control term Select-2nd-Store is true. This term is used in conjunction with STO-A-SEL to control which store port board will receive $TS and External $GO ($ COMMAND) during the early cycle.

The Non-Existent-Address (NEA) logic on the SCAME board compares address bits 02 to 08 with the contents of the NEA portion of the configuration register. If the address is equal to or greater than the register an NEA results. If the cycle is not a controller only operation, the level ABORT is generated causing the store function to be aborted. Abort inhibits the command strobe from being sent to the selected store unit.

The command lines, in addition to being transmitted to the store port boards, are latched up in two registers. One register is located on the early cycle control board (SCAMM). The other register is located on the SCAMF board.

The command register on the SCAMM board includes address bits 18 and 19 and is used for the decode of all illegal commands.

Of the 32 possible commands with the 5 command lines, only 14 commands are legal. This leaves 18 illegal combinations. Any SCU command which addresses upper store is also defined as an illegal command. Bits 18 and 19 are included in the register and decode because the RGR/SGR commands use this address to define the type of operation to be performed. All the variations of the RGR/SGR commands, except for store mode register operations, are considered controller operations. For store mode register operations, address bits 18 and 19 are true (high). This inhibits the illegal command fault if the upper store is addressed.

The early cycle timing board decodes the read lock and write lock instructions to permit set or reset of the lock bistable.

The WRT-CMD is decoded for loading into the control flag register for data control during the late cycle. Controller-Command is also a term which is decoded for loading into the control flag register.

The command register (SCAMF) is loaded at the same time as the command register on the SCAMM board with LOCK-ZAC. This register includes address bits 18-20 along with command lines A to D. The function of this register is to hold the current early cycle command for decoding. The address bits are included for the RGR/SGR commands. These commands use the address to define the specific variation of the operation. The early cycle operations (writes and set register commands) obtain their control lines directly from the decode of this register. Decodes of commands for late cycle operations (reads) are loaded into the control flag register (SCAMF) during the early cycle. The commands loaded into the control flag register are gated out during the late cycle and become the sense lines to control data flow.

The command lines are loaded into the history register along with address lines 5-8, 12-23, and the ZAC parity bit for display for troubleshooting.

All ZAC lines internal to the SCU feed the inputs of all SCUMY boards. Selection of which store unit pair receives the ZAC input is a function of which store busy bistable is set during the early cycle. Both store units of the pair receive the $ZAC that causes the ZAC lines to be loaded into the ZAC registers. As soon as these registers are loaded, the output is redriven thru drivers to the store units.

| Store Operation | COMMAND | Command Strobe |
|---|---|---|
| Read Data | RRS-SP, RRS-DP | $RRS |
| Read Data | RDLCK | $RRS |
| Read Data | CON | $RRS |
| n/a | RMSK-DP, SMSK-DP | $RRS |
| n/a* | RGR/SGR | $RRS |
| n/a | XEC | $RRS |
| n/a | SXC | $RRS |
| Write | CWR-SP | $CW1 |
| Write | WRLCK | $CW1 |
| Write | CWR-DP | $CW2 |

*$RGR/SGR Store Unit Mode Register commands use $RRS in the store unit.

There is also parity generation logic on the SCUMY board which is used to generate parity on 22 address bits. LB21, a pin on the free edge connector of the SCUMY board determines that parity will be generated for 22 address lines. This free edge connector is connected by cable to the port board of the store unit. The port board has internal jumpers to denote the address bus size. The term EXTENDED is true (high) and parity is generated on 22 address lines (ZAC bus = 35 bits).

### 3.3.4 Timing and Control

Logic for all SCU timing and control is divided into two major functional units:

- o Early cycle timing and control logic.

- Late cycle timing and control logic.

An early cycle is initiated when an interrupt strobe ($INT) from an active module is accepted by the SCU. The cycle is concluded when the command strobe is sent to memory. A late cycle is initiated when the answer strobe ($ANS) is received from memory. The cycle is concluded when the illegal action code is placed on the output lines.

The early cycle timing and control logic is physically located on the SCAMM board. The late cycle timing and control logic is on the SCAMJ board.

3.3.4.1 Write Data and Parity Control Logic

Write Data and parity control logic permits the contents of the write register 00-71 to be gated to the SCU's internal write bus when SENSE RECEIVE (¢RCV) is detected by the board. The write parity upper and lower are also gated to the SCU's internal lines by ¢RCV. Two control terms from the SCU (COMMAND-D and ¢RCV) govern transmission of data from the write register to the SCU's write data bus.

- COMMAND-D (CMD-D) is the command-D received from the active module via the ZAC bus and is locked up in a register chip on the SCAMX board.

- SENSE RECEIVE is transmitted to the SCAMX board by the SCAMM board during the early cycle.

SENSE RECEIVE and CMD-D, both true (high), cause the SCAMX board to generate two internal control lines: ¢RCV-DATA and ¢RCV-SHIFT.

¢RCV-DATA causes write register 00-35 to be gated to the SCU's write data bus 00-35 along with the upper parity bit. SENSE RECEIVE DATA also controls the output of the upper word parity. If a parity error is detected, the term Upper-Data-Error is true which, enabled by SENSE RECEIVE DATA, is ORed to the SCU internal logic as WRITE-DATA-ERROR. The SCU responds by causing the Write Parity Error (WPE) fault to be generated, aborting the operation in the early cycle.

¢RCV-SHIFT causes write register 36-71 to be gated to the SCU's write data bus 36-71 along with the lower parity bit. It also permits the lower word parity check logic output to be ORed to the SCU internal logic as WRITE-DATA-ERROR if a parity error exists. The SCU responds by aborting the operation with a WPE.

3.3.4.2 Write Data and Parity Control - Single Precision

CMD-D false (low) indicates a single precision operation. When ¢RCV is detected by the port and CMD-D is false, the SCAMX generates two internal control lines which are ¢RCV-DATA and ¢RCV-SHIFT. ¢RCV-DATA functions the same as the double precision operation. It gates write register 00-35 onto the SCU write bus 00-35 along with the upper parity bit and the output of the write data parity checker for bits 00-35. ¢RCV-SHIFT causes write register 00-35 to be gated onto the SCU write data bus 36-71.

The parity bit received from the active module is gated to the SCU as WRITE DATA PARITY LOWER. The output of the write data parity checker for the upper word is enabled to the SCU.

3.3.4.3 Read Data and Parity Control - Double Precision

Control for transmission and shifting of data is handled by three input signals to the SCAMX port board. The signals are:

- $DA-PORT
- ENABLE-XMIT-PA
- SENSE SHIFT-XB-PA

$DA is sent during the late cycle for any command requested by the active module. ENABLE-XMIT-PA is a function of the control register write bit being decoded during the late cycle. If the write term is low, ENABLE-XMIT-PA is sent to all SCAMX boards. SENSE-SHIFT-XB-PA is decoded by the SCAMJ board by looking at CMD-D and address bit 23 (even/odd). If CMD-D is false (single precision) and address bit 23 is true (odd address), the term SENSE-SHIFT-XB-PA is generated and sent all SCAMX boards. For double precision operation $DA-PORT is generated, since the port has been flagged for response during the late cycle. SENSE-SHIFT-XB-PA is false because the CMD-D line is true. ENABLE-XMIT-PA is true because a read operation is being performed. The SCAMX board enables two internal control lines ($XMIT and $XMIT-SHIFT) and one external strobe ($XMIT-PA). $XMIT-SHIFT causes the SCU's read data bus 00-35 to be gated onto the basic bidirectional data lines (00-35) to the active module. This control also gates the read data parity upper bit from the parity register on the SCAMX board to the active module.

- $XMIT always strobes the parity bits received from the SCU into the port board parity register.

- $XMIT always enables the SCU read data bus 36-71 to the extended bidirectional data lines to the active module.

- The lower parity bit from the SCU is always gated to the parity lower bit to the active module.

- $XMIT-PA is used by the active module to indicate that data has been trasmitted, and latches it into the receivers on the MY board.

3.3.4.4 Read Data and Parity Control - Single Precision (Even Address)

Data and parity control for read operations that are directed to even addresses are identical to the double precision operation except that the active module only accepts the data on the basic data lines. The extended data lines are latched into the active module's MY port board receivers, but are ignored by the module.

## 3.4 SCU COMMANDS

Commands applicable to the SCU are described in this section and summarized in Table 3-1. Symbols and terms used in the command descriptions are as follows:

- o Y = absolute address received by the SCU from the active module initiating the command.
- o C(Y) = contents of storage location specified by Y.
- o $\Longrightarrow$ = replaces.
- o Basic Data Lines = bits 00-35.
- o Extended Data Lines = bits 36-71.

Table 3-1.
COMMAND SUMMARY

| Command | Mnemonic | Command Line Code A | B | C | D | E | Octal Code* |
|---|---|---|---|---|---|---|---|
| READ RESTORE SP | RRS-SP | 0 | 0 | 0 | 0 | 0 | 00 |
| READ RESTORE DP | RRS-DP | 0 | 0 | 0 | 1 | 0 | 04 |
| READ CLEAR | RCL | 0 | 0 | 1 | 0 | 0 | 10 |
| READ MASK DP | RMSK-DP | 0 | 0 | 1 | 1 | 1 | 16 |
| CLEAR WRITE SP | CWR-SP | 0 | 1 | 0 | 0 | 0 | 20 |
| CLEAR WRITE DP | CWR-DP | 0 | 1 | 0 | 1 | 0 | 24 |
| SET MASK DP | SMSK-DP | 0 | 1 | 1 | 1 | 1 | 36 |
| LOCK | RDLK | 1 | 0 | 0 | 0 | 0 | 40 |
| READ GEN REGISTER | RGR | 1 | 0 | 1 | 1 | 0 | 54 |
| SET GEN REGISTER | SGR | 1 | 0 | 1 | 1 | 1 | 56 |
| UNLOCK | WRLK | 1 | 1 | 0 | 0 | 0 | 60 |
| CONNECT | CON | 1 | 1 | 0 | 0 | 1 | 62 |
| EXECUTE | XEC | 1 | 1 | 0 | 1 | 1 | 66 |
| SET EXECUTE | SXC | 1 | 1 | 1 | 0 | 1 | 72 |

*Octal Code adds a pseudo zero-bit in the low order bit position of each command.

### 3.4.1 Memory Read Commands

Memory read commands are used by the active modules to extract data which is stored in memory.

#### 3.4.1.1 Read Restore Single Precision (RRS-SP) ($00_8$)

The C(Y) are sent to the requesting system port on the basic data lines. The C(Y) in memory are not altered. The data word is selected using address bits 2-23 of address Y. The C(Y) equals 36 data bits and one parity bit.

#### 3.4.1.2 Read Restore Double Precision (RRS-DP) ($04_8$)

The C(Y) are sent to the requesting active module on the basic and extended data lines. The C(Y) in memory are not altered. The data consist of C(Y) (72 data bits) plus two parity bits.

#### 3.4.1.3 Read and Clear (RCL) ($10_8$)

The RCL command functions the same as RRS-SP except C(Y) are restored to memory as all zeroes.

### 3.4.2 Memory Write Commands

Memory write commands are used by the active modules to transfer data into memory. Two types of write commands are used:

- o SINGLE PRECISION COMMAND - used with the contents of the zone lines to transfer up to 36 bits of data into memory. Data may be transferred in 3 or 6 bit bytes.
- o DOUBLE PRECISION COMMAND -transfers contents of basic and extended data lines (72 bits) into memory.

Odd parity is checked and generated on each 36 bit increment of write data by the SCU. If an illegal action is detected, the cycle is aborted and the appropriate illegal action code is sent to the active module.

#### 3.4.2.1 Clear/Write Single Precision (CWR-SP) 01000 ($20_8$)

C(Y) = The contents of the basic data lines from the requesting port.

C(Y) Are written into memory in the location specified by the contents of the address lines. The bit positions to be altered must be specified by the contents of the zone lines. Unaltered bits are restored to memory without change.

#### 3.4.2.2 Clear/Write Double Precision (CWR-DP) 01010 ($24_8$)

C(Y) = The contents of both basic and extended data lines.

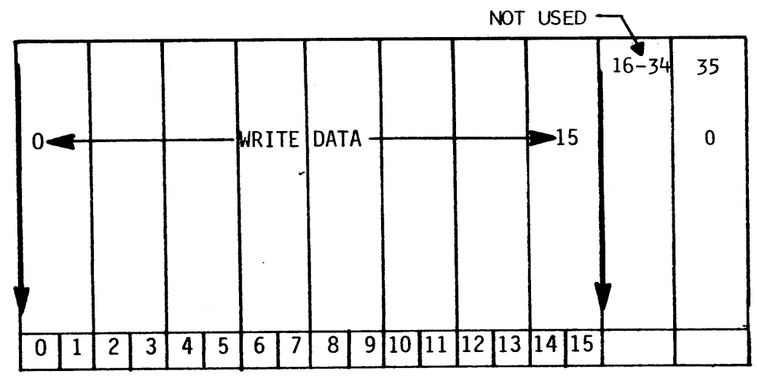
C(Y) Are written into the memory location specified by the contents of address lines.

### 3.4.3 System Control Commands

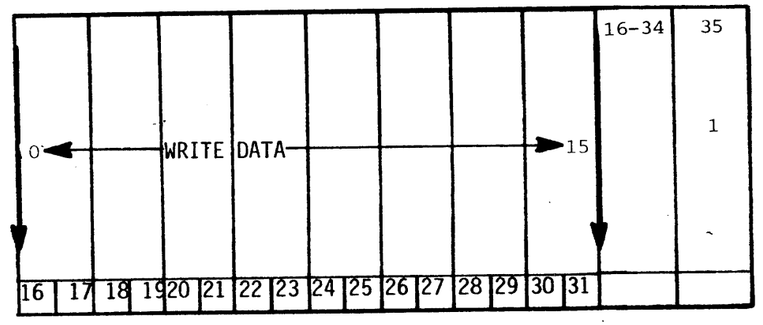
These commands are used by the active modules to control system operation and to communicate with each other. Intersystem communication and control are implemented by means of registers which are physically located in the SCU.

3.4.3.1 Set Execute Cells (SXC) 11101 ($72_8$)

The SXC command is used by the active modules to set (write) the SCU execute interrupt register. The execute interrupt cells (register bits) are used by the active modules for intermodule communication. The most common usage is communication between the IOM and the CPU. In this application, the IOM sets a particular cell with an SXC command. The CPU responds by executing (reading) that cell with an XEC command. SXC is always single precision. SXC is similar in operation to CWR-SP. Figure 3-8 shows the structure of the execute interrupt register and the alignment of the data field presented to the SCU by the active module. As shown in Figure 3-8, data bit 35 of the SXC is the stacking bit during execution of the command. If bit 35 is "0" (false), the upper portion of the execute interrupt register (0-15) is selected. If bit 35 is "1" (true), the lower portion of the execute interrupt register (16-31) is selected. The total 36 bits of the data word are checked for parity and the SXC cycle is aborted if an error is detected.




A. INTERRUPT REGISTER FORMAT - WRITE DATA 35 = 0
SET UPPER CELLS




B. INTERRUPT REGISTER FORMAT - WRITE DATA 35 = 1
SET LOWER CELLS

**FIGURE 3-8 INTERRUPT REGISTER FORMATS**

As with all internal SCU commands, the address field associated with the cycle must access the lower assigned store unit attached to the SCU. Failure to comply results in an illegal command IA code (12 octal) and abort of the cycle. Data sent to the active module during an aborted cycle is whatever happens to be in the store unit location designated by the address. The SCU performs a read restore for aborted cycles and sends the appropriate illegal action code. An SXC command is also aborted if a ZAC parity error is detected by the SCU.

3.4.3.2 Execute Interrupt Cells (XEC) 11011 ($66_8$)

The XEC command is used by the active modules to read/execute an SCU execute interrupt cell (bit). The most common usage is communication between the IOM and the CPU. In this application, the IOM sets one of the 32 execute interrupt cells, using an SXC command. The CPU responds by executing that cell with an XEC. The XEC command is always single precision. It is similar in operation to RRS-SP.

Each execute interrupt cell has a hardwired address associated with it. The system operating software will allocate these addresses in the memory as the vectors associated with each individual interrupt cell. Figure 3-9 shows the hardwired address associated with each execute interrupt cell and the data lines on which this address is transferred to the active module executing the interrupt.

The SCU responds to an XEC command by transmitting (to the requesting module) the address associated with the highest priority interrupt bit that is set and unmasked. This address is presented to the active module in positions 12 thru 16 of the data field. If no unmasked cell is set when the active module attempts to execute an XEC command, the data field is returned with data bit 17 enabled (=1), indicating there are no cells to be executed.

As with all internal SCU type commands, the address field associated with the cycle must access the lower assigned store unit attached to the SCU. Failure to comply with this requirement results in an illegal command IA code (12 octal) and aborting of the cycle. The data sent to the active module during an aborted cycle will be whatever happens to be in the store unit location designated by the address. The SCU performs a read restore for aborted cycles and sends the appropriate illegal action code. An XEC command will also be aborted if a ZAC parity error is detected by the SCU.

3.4.3.3 Connect (CON) 11001 ($62_8$)

The connect command (62 Octal) is used for intercom comunication between active modules. The most common is the CPU to IOM communication. The opposite direction (IOM to CPU) uses the SXC command for communication. In applying the connect command, the CPU initiates an SCU cycle with its command lines equal to a 62 octal (CON). The SCU responds by reading from the storage unit the addressed location specified by the CPU. The word read from the storage unit (single precision) is examined to identify the active port to which the CPU wishes to have the connect pulse ($CON)

| Cell Number | Address (Octal) | Cell Number | Address (Octal) |
|---|---|---|---|
| 0 (Highest Priority) | 00 | 16 | 40 |
| 1 | 02 | 17 | 42 |
| 2 | 04 | 18 | 44 |
| 3 | 06 | 19 | 46 |
| 4 | 10 | 20 | 50 |
| 5 | 12 | 21 | 52 |
| 6 | 14 | 22 | 54 |
| 7 | 16 | 23 | 56 |
| 8 | 20 | 24 | 60 |
| 9 | 22 | 25 | 62 |
| 10 | 24 | 26 | 64 |
| 11 | 26 | 27 | 66 |
| 12 | 30 | 28 | 70 |
| 13 | 32 | 29 | 72 |
| 14 | 34 | 30 | 74 |
| 15 | 36 | 31 | 76 (Lowest Priority) |

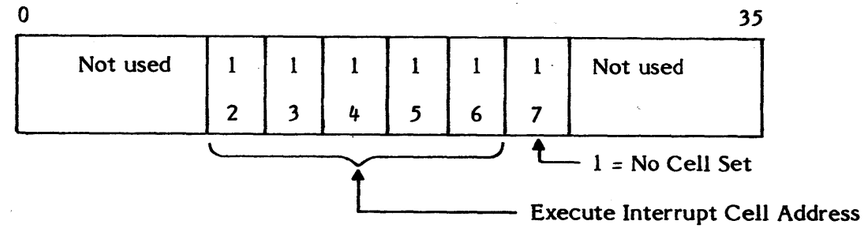
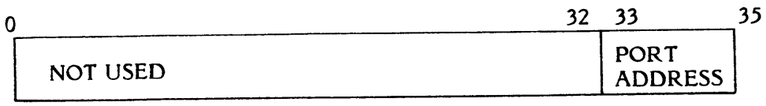



**FIGURE 3-9 INTERRUPT CELL DDRESS (VECTOR)**

issued, and a $CON is sent to that port. The structure of the connect word read from the storage unit during execution of the connect command is shown below.




WORD STRUCTURE FOR CONNECT COMMAND

During execution of CON, the C(Y) are sent to the SCU. The C(Y) bits 33-35 are latched into a 3-bit register in the SCU. The contents of the connect register are decoded to select the port and generate a connect strobe ($CON) which is sent to the device connected to that port.

3.4.3.4 Set Mask Register (SMSK-DP) 01111 ($36_8$)

The SMSK-DP command is used by the active module to set (write) data into the execute interrupt mask register assigned to its associated port and to set the port mask register. If neither of the two 32-bit execute interrupt mask registers has been assigned to the active port, only the port mask register is set. Execution of this command causes the SCU to set both halves of the mask word. The mask bits are set from the contents of the data lines as follows:*

- Contents of basic data lines, bits 0-15, set the upper mask register which consists of the interrupt mask for cells 0-15 and 32-35, which comprises the port mask for ports 0-3.
- Contents of the extended data lines, bits 36-51, set the lower mask register, which consists of interrupt mask cells 16-31 and 68, which comprises the port mask for port 4.

The address field of the command must address the lower store unit. Failure to comply with this requirement results in an illegal command IA code (12 octal) being transmitted to the originating active module and the cycle being aborted. The set mask command will also be aborted if a ZAC or data parity error is detected.

3.4.3.5 Read Mask Double Precision (RMSK-DP) 00111 ($16_8$)

RMSK-DP causes the causes the SCU to transmit the upper mask on the basic data lines and the lower mask on the extended data lines for the mask register assigned to the requesting active module.*

C(MASK REGISTER UPPER) $\Longrightarrow$ Basic Data Lines 00-35
C(MASK REGISTER LOWER) $\Longrightarrow$ Extended Data Lines 36-71

*The data format is the same as for RGR/SGR Execute Interrupt Mask Register (See 3.4.4.2.15)

3.4.3.6 Lock (RD-LOCK) 10000 ($40_8$)

The C(Y) (bits 00-17) are sent, unaltered, via the basic data lines to the active module. The C(Y) equals 36 bits and one parity bit. RD-LOCK is identical to RRS-SP except that it sets the lock bistable on the SCAMM board. When set, the lock bistable prevents any other port from gaining priority unless its key line is true.

The key line is always true from all other active modules except NSA(Level 66B) or MULTICS processor (6180 CPU). These units are also the only units that use the lock and unlock commands. Basically, this command is only needed by the NSA/MULTICS system when multi-processor operation is utilized.

The CPU uses the lock and unlock commands whenever it executes a Read-Alter-Rewrite (RARW) type instruction or modification.

There are certain control words used by the CPU to control memory assignment. If they are to be altered, the CPU uses an RARW type instruction. If, during the interval between the read (Lock) and write (unlock), the other appending unit/virtual unit attempts to access the same SCU to load an appending word, it will do so with its key line false. This prevents the second CPU from gaining priority until the first CPU has issued the unlock command, resetting the lock bistable.

If an unlock command is not received by the SCU within 20 microseconds after a lock instruction, a lock timeout occurs, resetting the lock bistable. No fault occurs or is reported to the system.

3.4.3.7 Unlock (WRLK) 11000 ($60_8$)

The character positions to be altered by the unlock command are specified by the zone lines. The C(Y) are selected by bits 00-17 of address (Y). This instruction is the same as the clear write single precision except that it resets the lock bistable that was set by the lock command. The SCU should never see an unlock command which was not preceded by a lock command.

3.4.4 General Register Commands

The general register commands permit the program to implement test and diagnostic routines on both the SCU and its connected memory units. Certain registers have been incorporated in the SCU and memory control logic which are used only for testing purposes.

There are two general register commands, set general register and read general register. Each command has several variations, shown in Figure 3-10 on the following page.

| Octal Address | Designated Register | RGR $54_8$ Readable | SGR $56_8$ Settable |
|---|---|---|---|
| XYYY 0X(a) | Mode Register (MR)(SCU) | Yes | Yes |
| XYYY 1X(a) | Configuration Register (CFG) | Yes | Yes |
| XYY0 2X(a) | Interrupt Mask Register, Port 0 (MSK0) | Yes | Yes |
| XYY1 2X(a) | Interrupt Mask Register, Port 1 (MSK1) | Yes | Yes |
| XYY2 2X(a) | Interrupt Mask Register, Port 2 (MSK2) | Yes | Yes |
| XYY3 2X(a) | Interrupt Mask Register, Port 3 (MSK3) | Yes | Yes |
| XYY4 2X(a) | Interrupt Mask Register, Port 4 (MSK4) | Yes | Yes |
| XYY5 2X(a) | Interrupt Mask Register, Port 5 (MSK5) | Yes | Yes |
| XYY6 2X(a) | Interrupt Mask Register, Port 6 (MSK6) | Yes | Yes |
| XYY7 2X(a) | Interrupt Mask Register, Port 7 (MSK7) | Yes | Yes |
| XYYY 3X(a) | Interrupt Cells (IC) | Yes | Yes |
| XYYY 4X(a) | Elapsed Time Clock (ETC) | Yes | Yes |
| XYYY 5X(a) | Elapsed Time Clock (ETC) | Yes | Yes |
| XYYY 6X(a) | Store Unit Mode Register (SU) | Yes | No |
| XYYY 7X(a) | Store Unit Mode Register (SU) | Yes | No |

X = Address of SCU or Store
Y = Don't care

Notes: (a) The address field must address the SC configured lower storage unit

**FIGURE 3-10 GENERAL REGISTER COMMAND FORMATS**

3.4.4.1 Set General Register (SGR) 10111 ($56_8$)

The SGR command is similar to the CWR-DP command with the exception that an SCU or store unit register is altered, not a store location. Acceptable variations of SGR are:

| Octal Address | Register | Mnemonic |
|---|---|---|
| XYYY 0X* | SCU Mode Register | (MR) |
| XYYY 1X* | Configuration Register | (CFR) |
| XYYn 2X* | Execute Interrupt Mask Register | (MSKn) |
| XYYY 3X* | Interrupt Cells | (IC) |
| XYYY 4X* or XYYY 5X* | Elapsed Time Clock | (ETC) |
| XYYY 6X or XYYY 7X | Store Unit Mode Register | (SU) |

Where: X = Address of SCU or Store.
Y = Don't care.
n = Port number.

*Address must be directed to the lower configured store (pair) on the SCU or an illegal command fault will occur.

3.4.4.1.1 SGR - SCU Mode Register Address = XYYY0X

The SCU mode register is 20 bits in length and is intended to be utilized by the T&D programmer to margin the voltages and timing strobes, as well as exercising the fault and parity logic within the unit.

3.4.4.1.2 SGR Configuration Register Address = XYYY1X

The SGR-CFR command will load the 55-bit programmable configuration register if the mode switch on the configuration panel is in the PROGRAM position. This command allows the executive program to control the configuration of the SCU, dynamically. See the RGR command variation for details and data format.

3.4.4.1.3 SGR Execute Interrupt Mask Register Address = XYYn2X (n = Port #)

The SGR-MASKn is the only means available to the user to set another port's executive interrupt mask register. If the port designated by address bits 18, 19 and 20 (n) does not have a mask register assigned, no mask register will be affected, and no illegal action returned. The command does not affect the port mask register.

3.4.4.1.4 SGR Interrupt Cells Address = XYYY3X

The SGR-IC command causes the data received by the SCU to be used to load the execute interrupt cell register. Since SGR-IC is a T&D function, this command can set or reset any cell or combination of cells. The data format is the same as for the RGR-IC command.

3.4.4.1.5 SGR Elapsed Time Clock Address = XYYY4X or XYYY5X

The ETC command causes the elapsed time clock to be loaded from the data received by the SCU from the active module.

3.4.4.1.6 SGR Store Unit Mode Register Address = XYYY7X

The SGR-SU command permits the T&D programmer to margin the timing and voltages of the store units.

3.4.4.2 Read General Register (RGR) 10110 ($54_8$)

The active modules use this command to access registers in the SCU and memory logic. This command uses the formats shown in Figure 3-10. The selected SCU does not transmit the contents of storage location. Rather the contents of the SCU register specified by address Bits 9-14 are transmitted to the active module via the requesting port. This command causes a double precision transfer, hence, both basic and extended data lines are used.

3.4.4.2.1 Mode Registers

The mode registers are used by the T&D and Executive programs to test and monitor the operation of the SCU and its related store unit(s). These mode registers may be set (SCU only) by the SGR memory command. The mode registers may be read with the RGR memory command and will return the contents of the register or the manual margin switches if the unit is not in program mode. The address field is used in conjunction with the command to determine whether the SCU or store mode register is to be selected.

The SCU mode register allows control of the following functions:

- Timing Margins
- Voltage Margins (SCU and Store)
- Parity Disable SCU
- Parity Override
- Disable IA (from store unit)
- Disable History Register on IA

The manual voltage margins for the store are controlled by switches on the power control module (PCM) of the store power supply.

3.4.4.2.2 RGR Configuration Format

The configuration of the SCU is controlled by the configuration register, most of which is contained on the SCAMC board. This register can be set or read by program or manual control. Manually, the register may be set by the switches on the configuration panel. The program/manual switch on the configuration panel determines the source to be used to set the register. If the switch is in the program position, the register may be set/read by the SGR/RGR SCU commands with an address field of XYYY1X (Y = Don't Care). Address field must address the lower store.

When in the program mode, an initialize will cause the configuration register to be set with the contents of the configuration switches. This initialize can be caused by powering the unit up, maintenance panel initialize, or system initialize.

The configuration format is presented in Figure 3-11.

not settable

| 00 06 | 07 | 08 11 | 12 15 | 16 19 | 20 | 21 | 22 29 | 30 | 31 | 32 35 | Bit Number |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mask Reg. A Assignment | OFF | Store Size | Store On Line | Port # Req. | N/U | MODE | ON Non-Existant Address | INT | LWR | Port Mask | Name |
| 0 1 2 3 4 5 6 7 Port Numbers | | 4 2 1 | A A B B 1 1 | 8 4 2 1 base 10 | | | 2 3 4 5 6 7 8 Addr. Bits | | | 0 1 2 3 Port# | Field Information |
| DS #8 | | DS #6 | nd | | | nd | DS #5 | #6 | | DS #4 | Display Information |
| MC10, 12 | | MC08 | | MF19 | | * | MC07 | ** | | MC05 | Logic Page Location |

* = MC04 ** = MC12 nd = Not Displayable

not settable not settable

| 36 43 | 44 | 45 56 | 57 63 | 64 67 | 68 71 | Bit Number |
|---|---|---|---|---|---|---|
| Mask Reg. B Assignment | OFF | not used | Cyclic priority Group Ports | not used | Port Mask | Name |
| 0 1 2 3 4 5 6 7 Port #s | | | 0 1 2 3 4 5 6 7 1 2 3 4 5 6 7 | | 4 5 6 7 Port # | Field Information |
| DS #9 | | | DS #4,5 | | DS #4 | Display Information |
| MC 11, 12 | | | MC06 | | MC05 | Logic Page Location |

**FIGURE 3-11 CONFIGURATION FORMAT**

3.4.4.2.3 Mask Register A Assignment (00-08)

PROGRAM MODE

A "one" in one or more of the bit positions 0 to 7 indicates that the corresponding ports will receive all XIP's that are unmasked by interrupt mask "A". Mask "A" may be assigned to more than one port under program control (only). Assignment of an interrupt mask register to a system port designates that port as a "control port". System software would not normally assign mask register ("A and B") to any one port as the results of such an action would be unpredictable.

A "one" in bit position 8 indicates the interrupt mask register is unassigned, and one bits in positions 0 to 7 are ignored by the SCU.

MANUAL MODE

A switch "MASK/PORT ASSIGNMENT (MASK A)" on the configuration panel provides the ability to assign interrupt mask register "A" to one of any eight ports (0-7) or none (Mask "A" = Off). When a port is assigned the "other" seven Mask "A" port assignment bits are forced to the zero state in the configuration register; i.e., interrupt mask "A" is not assigned to those "other" ports.

3.4.4.2.4 Store Size 09 - 11

PROGRAM MODE

Data bits 09, 10, 11 indicate the address boundary between store (or store pairs) "A and B", i.e., the total memory assigned to the lower store unit or pair of units.

| 09 | 10 | 11 | Lower Store Size | 09 | 10 | 11 | Lower Store Size |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 32K | 1 | 0 | 0 | 512K |
| 0 | 0 | 1 | 64K | 1 | 0 | 1 | 1M |
| 0 | 1 | 0 | 128K | 1 | 1 | 1 | 2M |
| 0 | 1 | 1 | 256K | 1 | 1 | 1 | 4M |

MANUAL MODE

An eight position switch (LWR STORE SIZE) provides the same function.

3.4.4.2.5 Store on Line

PROGRAM MODE

A one in bit positions 12, 13, 14 or 15 indicate that a particular store unit is ON LINE and able to accept requests. Store nits "A" and "B" are considered primary store units "A1" and "B1" are considered secondary.

It is illegal to configure a secondary store unit on line unless its respective primary is already on line.

MANUAL

Four switches, (STORE ON LINE/OFF LINE A B) one for each port, provide the same function.

3.4.4.2.6 Port Request (16-19)

PROGRAM

On an RGR configuration command, data bits 16, 17, 18 and 19 return the SCU port number of the requesting port (Who am I?). This field is not settable by program.

MANUAL

This field is not settable manually.

3.4.4.2.7 Mode (Program/Manual) (21)

PROGRAM

If data bit 21 is a one during an RGR configuration register command, the configuration mode switch is in the PROGRAM position indicating all settable bits of the configuration register may be altered. This bit is not settable under program control. If bit 21 is a zero and an SGR configuration register command is attempted, the command will terminate normally. (No illegal action will be reported.)

MANUAL

One 2-position switch which determines whether configuration information is controlled by switches or by program. The "TEST/NORMAL" switch on the configuration panel does not affect the functioning of the PROGRAM/ MANUAL switch.

3.4.4.2.8 Non-Existent Address (22 - 29)

PROGRAM

A one in bit position 22 indicates the NEA logic is enabled and any store unit addresses equal to or greater than the address in bits 23 through 29 will be faulted with an NEA. This logic is not active for controller commands.

| 23 | 24 | 25 | 26 | 27 | 28 | 29 | Configuration Register Bit Number |
|---|---|---|---|---|---|---|---|
| 01 | 03 | 04 | 05 | 06 | 07 | 08 | Address Bit Number |
| 2 M | 1 M | 5 1 2 K | 2 5 6 K | 1 2 8 K | 6 4 K | 3 2 K | Bit Config. Reg/Address Value |

MANUAL

NON-EXISTENT ADDRESS - ON/OFF - 2-8 switches provide the same function in manual mode.

3.4.4.2.9 INT (Interlace) (30)

PROGRAM

A one in bit position 30 indicates store units A and B are to be selected dependent on address bit 22 and a higher order address bit determined by the SIZE field. A zero in this field indicates store selection is to be made solely on the highest address bit (noninterlace).

MANUAL

The INTERLACE ON/OFF switch provides the same function in the manual mode.

3.4.4.2.10 LWR (Lower Store) (31)

PROGRAM

A one in bit position 31 indicates store "B" contains the "lower addresses". A zero in this position indicates store unit "A" contains the lower "addresses". The "lower" store unit may or may not contain the systems zero address. All nonmemory commands must have an address pointing to the "lower" store unit.

MANUAL

The LWR STORE A/B switch on the panel provides the same function.

3.4.4.2.11 Port Mask 0-3 (32 - 35)

PROGRAM

Data bits 32, 33, 34 and 35 indicate the state of SCU ports 0, 1, 2 and 3, respectively. A "one" bit indicates port enabled, a "zero" indicates port off. When a port is off, no $ Interrupt will be accepted, all XIP's will be inhibited, and the READY line to the active module will be disabled. These bits may be altered by both the SMSK-DP and the SGR-CONF command if the CONFIGURATION MODE switch is in the PROGRAM position.

MANUAL

The PORT ENABLE 0 - 3 switches on the configuration panel provide the same function.

3.4.4.2.12 Mask Register B Assignment (36 - 44)

PROGRAM

Data bits 36 to 44 have the same meaning for interrupt mask register "B" as bits 0 through 8 have for interrupt mask "A".

MANUAL

The MASK/PORT ASSIGNMENT - MASK B provides the same function for "B" as described for "A".

3.4.4.2.13 Cyclic Priority (57 - 63)

PROGRAM

Data bits 57 through 63 indicate which SCU ports are to be grouped together with equal priorities. Adjacent one bits increase the number of ports within a group with equal priority. A "zero" between groups of one bits separate the groups into two priority groups. System ports within a group have equal access to storage. The maximum number of ports within one grouping of cyclic priority is five. The operation within a group is such that no system port may have two consecutive accesses to storage after another port has requested access, unless a read lock command has been received (part of read lock/write lock or gating sequence provided for software).

In the case of continuous access requests from all ports within a group, access is granted in cyclic order starting with the highest priority (lowest numbered) system port.

Between groups, access requests by the highest priority group must all be satisfied before any access requested by a lower priority group are acknowledged.

If requests from a higher priority group are received while a lower priority group is being serviced, the lower priority group service stops at the end of the current operation and the higher priority group requests are serviced. The system controller stores the information necessary to return to the lower priority group after the higher priority group is serviced.

MANUAL

The CYCLIC PRIORITY 0/1 to 3/4 switches on the configuration panel provide the same function.

3.4.4.2.14 Port Mask 4 - 7 (68 - 71)

PROGRAM

Data bits 68, 69, 70 and 71 have the same relationship to ports 4, 5, 6 and 7 respectively as bits 32 through 35 have on ports 0 through 3.

MANUAL

The PORT ENABLE 4 - 7 switches on the configuration panel provide the same function.

3.4.4.2.15 Execute Interrupt Mask Register.

Address XYYn2Y (n = Port #)

The reading of the execute interrupt mask registers will return, on the data lines, the contents of the register specified by the address of the RGR command. If no mask register is assigned by the configuration register, zeros will be returned for the execute interrupt mask field. The common port mask register will be returned regardless of the mask registers.

| 00 EXECUTE INTERRUPT MASK CELLS 00 - 15 15 | 16 NOT USED 31 | 32 PORT MASK 4-7 35 |
|---|---|---|

EVEN WORD

| 36 EXECUTE INTERRUPT MASK CELLS 16 - 31 51 | 52 NOT USED 67 | 68 PORT MASK 4-7 71 |
|---|---|---|

3.4.4.2.16 Interrupt Cells Address XYYY3X

The reading of the interrupt cells will return the contents, on the data lines, as formatted below. It should be noted that the reading of the interrrupt cells in this manner does not change their state. A one indicates the cell is "SET". A zero means that cell is "RESET".

| 00 EXCUTE INTERRUPT CELLS CELLS 00 - 15 15 | 16 NOT USED 35 |
|---|---|

| 36 EXCUTE INTERRUPT CELLS CELLS 16 - 31 51 | 52 NOT USED 71 |
|---|---|

3.4.4.2.17 RGR Elapsed Time Clock Address XYYY4X or XYYY5X

The Elapsed Time Clock (ETC) of the the SCU is 52 bits and is counted up every one microsecond (1 MHz). It will roll over in approximately 142 years. It is readable with an RGR command with an address field of XYYY4X* or XYYY5X*. The clock is settable with an SGR command with the same address as used for the RGR command. Power up initialize will result in the ETC being reset to zero.

*X = Address must be directed to the lower configured store or an illegal command fault will result.

Y = Don't care

The format of the data for the elapsed time clock is:

| Not Used | | Double Precision Data Word | | | |
|---|---|---|---|---|---|
| 00 | 19 | 20 | 35 | 36 | 71 |
| | | 20 | 35 | 36 | 71 |
| | | Elapsed Time Clock | | | |

## 3.5 WRITE NOTIFICATION (HARDWARE CACHE CLEAR)

The write notification option (WHCC66LB), including the SCUMH logic board, is installed in all SCU's that are part of a data processing system in which the CPU(s) have the 8K cache memory. The function of write notification is to inform the CPU's when the contents of a location in main memory is changed. Notification is accomplished by transmitting the altered memory address to all CPU's in the system, except the one originating the change. The write notification control logic in the CPU uses this address information to determine whether the altered data is in cache memory and, if so, to clear that location in the cache directory.

Figure 3-12 is a block diagram showing the major functions performed by the SCUMH logic. Sheet numbers in the figure refer to drawing 58035764, the logic diagram for the SCUMH board.

Referring to Figure 3-12, a memory alter request to the SCU is defined in write notification logic by the terms ALTER-CMND and ¢RCV-ALL.

The term ALTER-CMND is developed by decoding command lines A, B, C, and E. When a write command, a read clear command, or an unlock command is decoded, ALTER-CMND goes to its true state, indicating that a memory alter request has been received from an active system port.

The term ¢RCV-ALL (control-receive-all) indicates that an active port has been granted memory access by the SCU. As shown in Figure 3-12, this is the OR function of the terms ¢RCV-PORT-0 thru ¢RCV-PORT-7 and will therefore be true when any active port has been granted memory access. ¢RCV-ALL, true, sets the BUSY bistable and resets the LWR-ADDR-SEL bistable which initiates an address transfer operation. The BUSY bistable set:

- Initiates the address timing sequence.
- Releases the ALTER-CMND logic so it can decode a new command.
- Transfers the contents of the address lines into the address register.

The LWR-ADDR-SEL bistable, set, transfers address bits RADDR-12 thru RADDR-21 plus parity to the write notification interface lines.

The following terms are generated by the timing logic:

- $BUSY +20

  This term transfers port update terms FUPDATE-PORT0 thru FUPDATE-PORT7 to the output drivers.

- $BUSY +30

  ANDed with ALTER-CMND to generate the lower address strobe. This strobe gates the lower address to the CPU's.

- $BUSY +55

  This term resets the BUSY bistable, terminating the 55 nsec BUSY strobe. BUSY reset also locks ALTER-CMND in the true state if it was already true.

- $BUSY +95

  Resets LWR-ADDR-SEL bistable which transfers bits RADDR-02 thru RADDR-11 plus parity to the write notification interface lines.

- $BUSY +120

  ANDed with ALTER-CMND to generate upper address strobe which gates the upper address to the write notification interface lines.

The timing sequence described above will be completed for all SCU operations. The address strobes, however, will not be sent during execution of read-restore commands or internal register commands because ALTER-CMND is false.

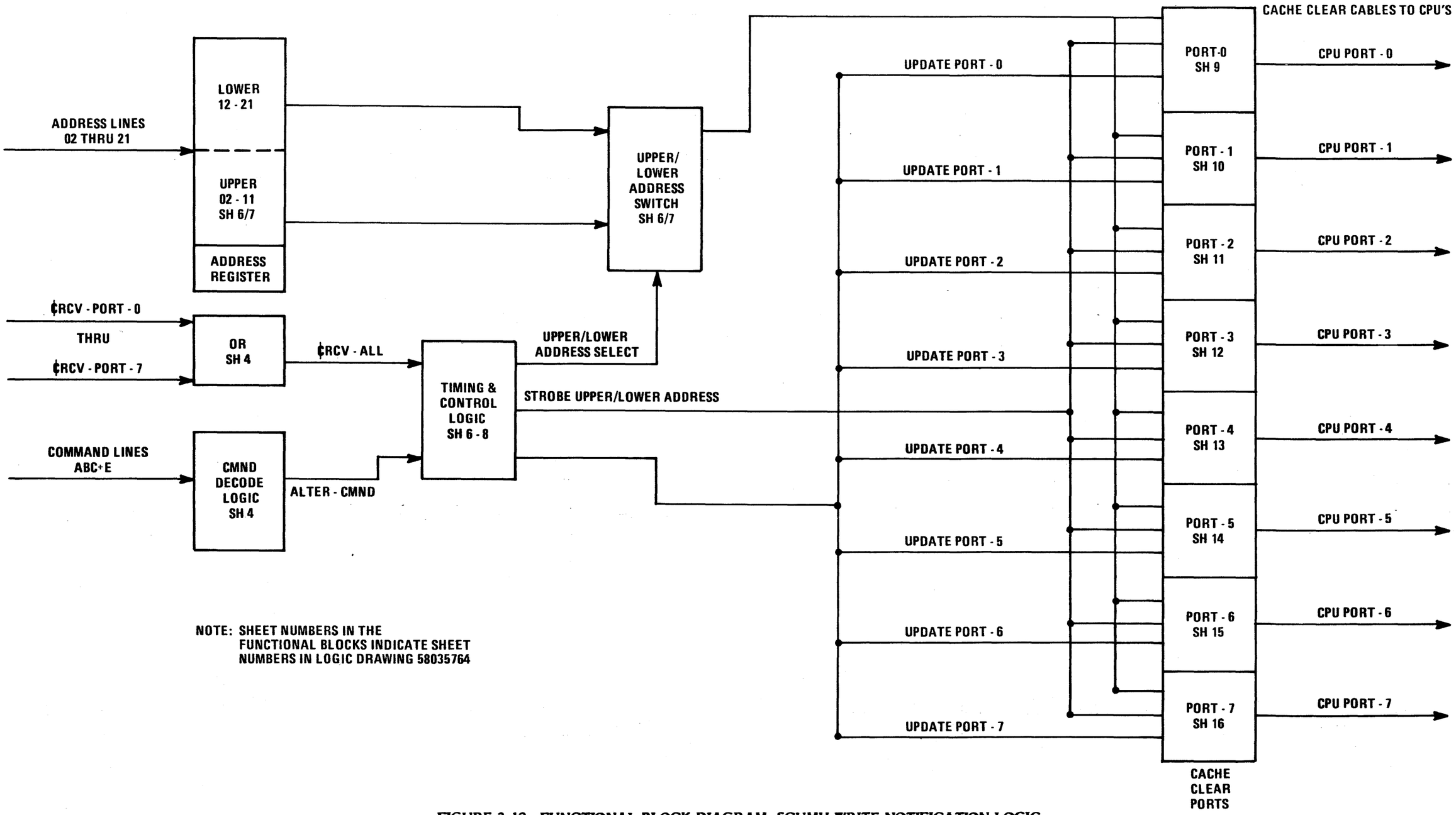



FIGURE 3-12 FUNCTIONAL BLOCK DIAGRAM, SCUMH WRITE NOTIFICATION LOGIC

The contents of the incoming address lines are transferred to the CPU's in 10 bit increments with odd parity for each increment. The low order half of the address (A21-A12) is placed on the output lines, followed by the high order half of the address (A11-A02). Figure 3-13 shows the logic used to multiplex A21/A11 and is typical for all address bit transfers. As indicated, address bits 21/11 are loaded into the address register when ¢RCV-ALL sets the BUSY bistable, causing the term $BUSY to go true. The LWR-ADDR-SEL bistable is then reset by ¢RCV-ALL so that RADDR-21;000 thru RADDR-12;000 are selected and placed on the ADDR-21/11 output lines. Later in the cycle (at BUSY +95), LWR-ADDR-SEL;100 goes to its true state which places RADDR-11;000 thru RADDR-01;000 on the output lines.

The contents of the address register are also applied to parity logic which generates odd parity on each 10 bit address increment. The parity bit is multiplexed on the parity line in the same manner as the address and transferred to the CPU's along with its associated address increment.

Transferring the address bits to the output lines, as described above, does not constitute an address transfer to the CPU. After the lines are changed, a strobe is generated which gates the contents of the address lines to the cache duplicate directory logic in the CPU. Also, it is necessary to inhibit the address transfer to the cache control logic of the CPU that initiated the memory change. If this is not done, the originating CPU will clear the data it just changed in its own memory.

Referring to Figure 3-12, two sets of control logic are used to meet the criteria described above.

- o The strobe generation logic which is common to all write notification ports. This logic provides precise timing of all address transfers.
- o The update logic which enables or disables each individual write notification port.

The port enable logic controls the strobe gates in each write notification port. This logic permits the upper and lower address strobes to be transmitted unless either the CPU and its input SCU port are turned off or the memory change was initiated by the CPU connected to the write notification port.

The terms ¢RCV-PORT-2;000 is derived from the port-2 SCAMX board. This signal is low (true) if port-2 is granted memory access and high (false) if any other port is granted memory access. In its true state, ¢RCV-PORT-2;000 forces UPDATE-PORT-2;000 high and FUPDATE-PORT2;100 low, disabling both strobe gates.

PORT-2-ON;000 is derived from the SCAMC board and is low (true) when port-2 is enabled and high (false) when port-2 is disabled. PORT-2-CPU-ON;000 is derived from the CPU via the cache clear cable. This term is low (true) when the CPU is turned on and high (false) when the CPU is turned off or is clearing cache. If either of these terms is low (true), the operation of the strobe gates is not affected. If both of the terms are high (false), PORT-2-ACTIVE;100 is false, thus, inhibiting the strobe gates.

The same logic scheme is used to generate both the lower address and the upper address strobes. The lower strobe bistable is set at BUSY +30(nsec) if ALTER-CMND;100 is true and resets itself via a 50 nsec delayed feedback from the Q output. This generates $LOWER-ADDR;000 which is inverted twice before it is transmitted to the CPU as $PORT2-LOWER;000. $PORT2-UPPER is generated in the same manner except that it is clocked at BUSY +120.

It should be noted that transmission of the upper and lower address strobes is the key to proper functioning of the write notification logic. The timing sequence is completed and the address placed on the write notification address lines each time an active port request is honored by the SCU. In its false state, ALTER-COMND prevents transmission of the upper and lower address strobes during execution of read-restore and internal register commands.

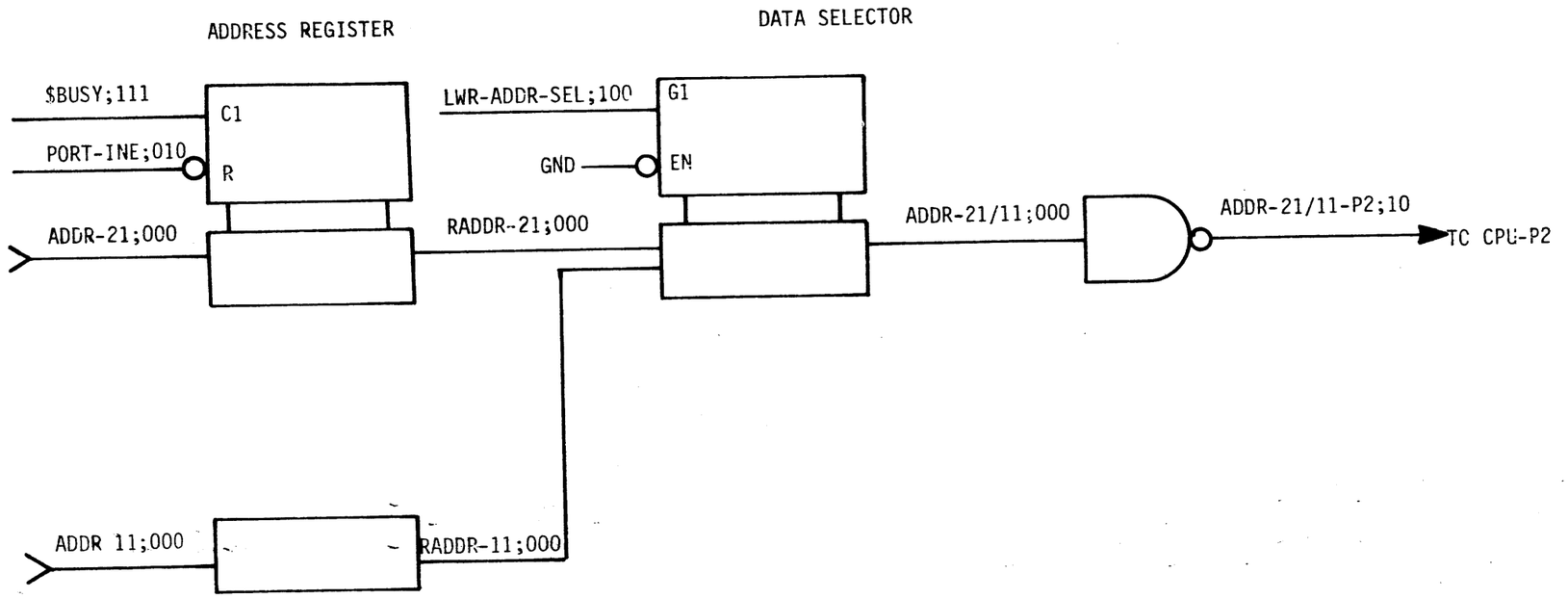



**FIGURE 3-13 ADDRESS TRANSFER LOGIC FOR A21/A11**

## 3.6 SCU CONFIGURATION PANEL

The SCU Configuration Panel is used to define to the SCU and the system, the associated store size, the control port(s) store(s) on line, the ports enabled, the ports to be granted equal priority, which store will be lower store, if interlace will be allowed, and finally if all these functions will be program or manually controlled.

Note: The lower panel (INITIALIZE, STATUS, and PANEL CLEAR) is used in Maintenance functions only. For that reason this panel is not covered in this manual.

All of the switches not shaded on Figure 3-14 are used for configuration. The two shaded switches will be explained first. Item numbers refer to Figure 3-14.

### 3.6.1 Panel Test/Normal (Item 1 )

The Panel Test/Normal switch is used, in the TEST position, to enable the Initialize and stop on IA switches.

### 3.6.2 Alarm Enable/Disable (Item 2 )

The Alarm Enable/Disable switch, when in the DISABLE position, prevents the SCU from sounding the audible alarm.

### 3.6.3 Configuration Switches (Item 3 )

The following switches control the Configuration register when the MODE PROGRAM/MANUAL switch is in the MANUAL Position. In the PROGRAM position the register is loaded from the switches when the SCU is INITIALIZED either due to power up, Configuration Panel Initialize with the TEST switch on, or Port Initialize (Bootload Initialize). In all cases the configuration of the SCU is always controlled by the Configuration Register.

In the Program Mode when a configuration switch is changed, the contents of the Configuration Register will not be changed unless the SCU is initialized or the MODE PROGRAM/MANUAL switch is toggled to MANUAL. The register can be altered, and hence the configuration, by the SGR-Configuration Register command when in the PROGRAM mode.

The Configuration Register contents may be read at any time with a RGR-Configuration Register command.

Note: The state of the Program/Manual switch can be read by bit 21 of the data for a RGR-SCU configuration register command. This function may not be set.

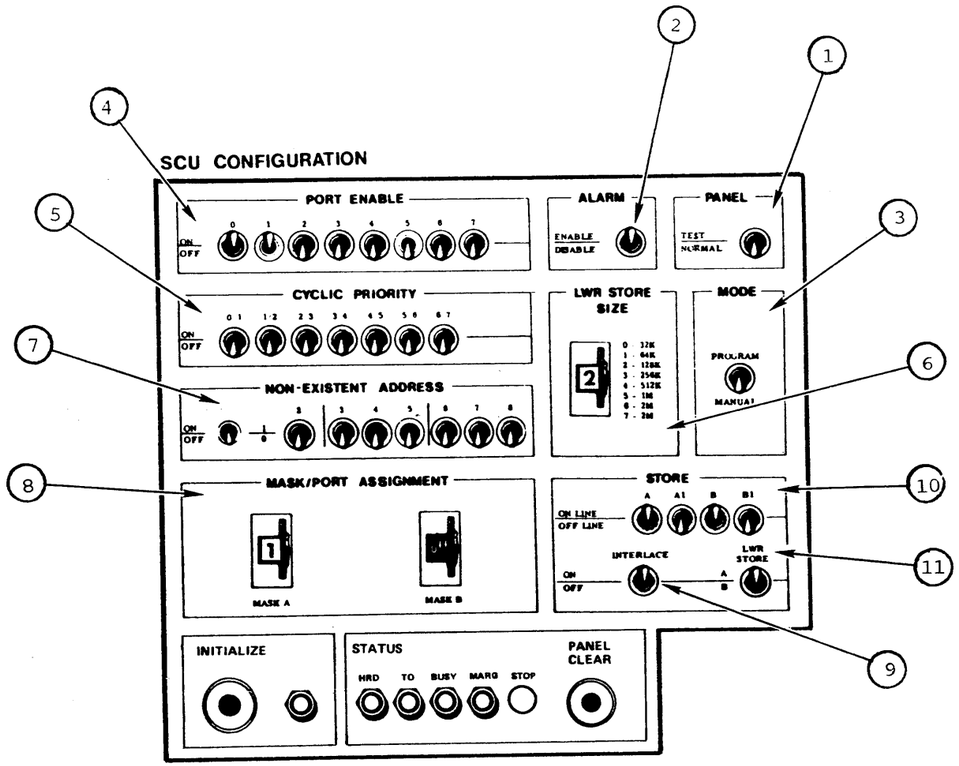


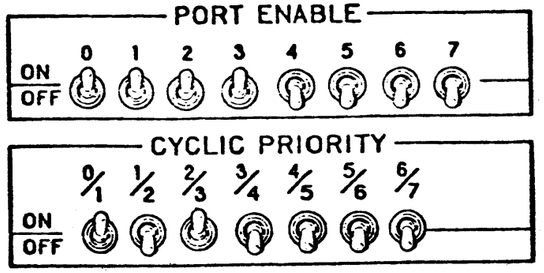

**FIGURE 3-14 SCU CONFIGURATION PANEL**

3.6.4 Port Enable Switches (Item 4 )

The port enable switches (one for each system port SCAMX board) allow ports to be turned on or off. In the "ON" position the port has the ability to perform all of its normal functions. In the "OFF" position no $Interrupt will be accepted, all XIP's will be inhibited, and the READY line to the active module will be disabled.

3.6.5 Cyclic Priority Switches (Item 5 )

The Cyclic Priority switches control which SCU ports are to be grouped together with equal priorities. Adjacent switches in the "ON" position increase the number of ports within a group with equal priority. A switch set to "OFF" between groups of switches set to the "ON" position separate the groups into two priority groups. For an example the illustration below shows Ports 0, 1, 2 and 3 are enabled by their Port Enable switches. Two Cyclic Priority switches are "ON"; 0/1 and 2/3. This means that ports 0 and 1 form a group with equal priority. Ports 2 and 3 also have equal priority to each other. If the 1/2 switch was also on, all ports (0 thru 3) would have equal priority, but since it is not, two distinct priority groups are formed.




System ports within a group have equal priority access to the SCU. The maximum number of ports within a group is five. The operation within a group prevents a system port from having two successive accesses to the SCU after another port has requested access, unless a RLCK command has been received.

In the case of continuous access requests from all ports within a group, access is granted in cyclic order starting with the highest priority (lowest numbered) system port. Between groups, access requests by the highest priority group must be satisfied before any access requested by a lower priority group are acknowledged. If requests from a higher priority group are received while a lower priority group is being serviced, the lower priority group service stops at the end of the current operation and the higher priority group requests are serviced. The SCU stores the information necessary to return to the lower priority group after the higher priority group is serviced.

The cyclic priority portion of the configuration register may be set/read when in the program mode with data bits 57-63 of an SGR/RGR-Configuration register instruction.

3.6.6 LWR STORE SIZE Switch (Item 6 )

The Lower Store Size thumbwheel switch is marked with positions 0 to 7. The number is the module 32K total size of lower store (pair). The legend to the right of the switch gives the settings for each lower store size. For example:

Store A = 128K

The size of the lower store is 256K and the switch should be set to 3. This assumes that the LWR STORE A/B defines store A as being lower store. The SCU logic will assume that the upper store is also 256K unless the NON-EXISTENT ADDRESS switches indicate otherwise. The Lower Store Size field of the configuration register may be set/read by data bits 9, 10, 11 of an SGR/RGR Configuration Register command, when the SCU is in the PROGRAM mode.

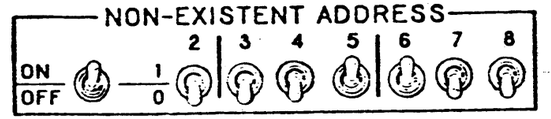
3.6.7 Non-Existent Address (Item 7 )

The eight Non-Existent Address switches are used to define the system, the size of the total store associated with the SCU whenever the store size cannot be exactly determined from the store size switch. The ON/OFF switch, when "ON", enables the NON-EXISTENT ADDRESS switches 2 to 8 to the Non-Existent Address fault (NEA) logic. If an address is equal to or greater than the settings of the switches 2 to 8, then a NEA fault will occur and the operation will be aborted.

Switch No./Addr Bit Value

| 02 | 03 | 04 | 05 | 06 | 07 | 08 |
|---|---|---|---|---|---|---|
| 2 | 1 | 5 | 2 | 1 | 6 | 3 |
| M | M | 1 | 5 | 2 | 4 | 2 |
| | | 2 | 6 | 8 | K | K |
| | | K | K | K | | |

Note that the switch number is the same as the 24 address bit number. For example if the SCU had a total of 384K, the switches would be set as shown below.



The NEA switches can be set/read with data bits 23 to 29 of an SGR/RGR Configuration register command when the SCU is in Program mode.

3.6.8 Mask/Port Assignment (Item 8 )

The Mask/Port Assignment switches are used to assign a mask register to a port (SCAMX). Assignment of a mask register to a system port designates that a port as a control port. This control port will receive all XIP's that are unmasked by that mask register. The SCU has two mask registers named "A" and "B". The mask register assignment thumbwheel switches positions are marked 0 to 7 and "OFF".

The Mask/Port Assignment portion of the configuration may be read/set with the RGR/SGR Configuration Register command if the SCU is in the PROGRAM mode. The read/write data format is:

Mask Register A Assignment

Bits 0 to 7 = Port Assignment

Bit 8 = 0-mask register A assigned

1-mask register A unassigned

Mask Register B Assignment

Bits 36 to 43 = Port Assignment

Bit 44 = 0-Mask register B assigned

1-Mask register B unassigned

Mask registers A and B assignment is displayed by positions 8 and 9 of the SCU display panel.

3.6.9 Store Interlace On/Off (Item 9 )

This switch, if on, will cause store units A and B to be selected dependent on address bit 22 and a higher order address bit determined by the SIZE field. If in the "OFF" position, selection will be determined by the high order bits only. The configuration register bit for this function can be read/set with data bit 30 for a RGR/SGR Configuration register command when the SCU is in PROGRAM mode.

3.6.10 Store On Line/Off Line (Item 10 )

The Store On Line/Off Line switches A, A1, B and B1 allow the individual Store ports (SCUMY boards) to be enabled or disabled by switch. The switch, if "ON LINE", permits the store to be able to accept requests. Store "A" and "B" are considered primary store units and units "A1" and "B1" are considered secondary. It is illegal to configure a secondary store unit on line unless its respective primary is already on line. Bits 12, 13, 14, 15 of the data for a RGR/SGR Configuration register command allow the ON LINE/OFF LINE portion of the Configuration register to be read/set if the SCU is in the PROGRAM mode.

3.6.11 LWR STORE A/B (Item 11 )

This switch allows Store A or Store B to be selected as the lower addressed store unit. Store unit A or B can be made up of one or two store units.

This function can be read or set with bit 31 of the data for a RGR SGR Configuration register command if the SCU is in the PROGRAM mode.

## 4.0 MAINTENANCE AIDS

The 4MW SCU designed for use in the 66L System provides microprocessor logic on the SCU logic boards to enable troubleshooting of the MMU on a local or remote basis. When a fault is detected in the MMU, one of the following capabilities may be selected in an attempt to isolate the problem to an ORU.

## 4.1 DYNAMIC MAINTENANCE PANEL (DMP)

The DMP consists of the microprocessor logic contained on the SCU logic boards which is specifically oriented to fault isolation within the MMU. Access to the microprocessor logic is accomplished through the 58056792 Connector Plate shown on this page. The connectors on the plate are used as follows:

### 4.1.1 MAINT PANEL - J01

Connector J01 povides an interface between the DMP and the Portable Maintenance Panel (PMP). When a fault is detected at a site where no maintenance contract has been purchased, the PMP provides off-line testing of the MMU. For troubleshooting procedures necessary to isolate the fault to an ORU, refer to 58009928, Test and Repair Manual.

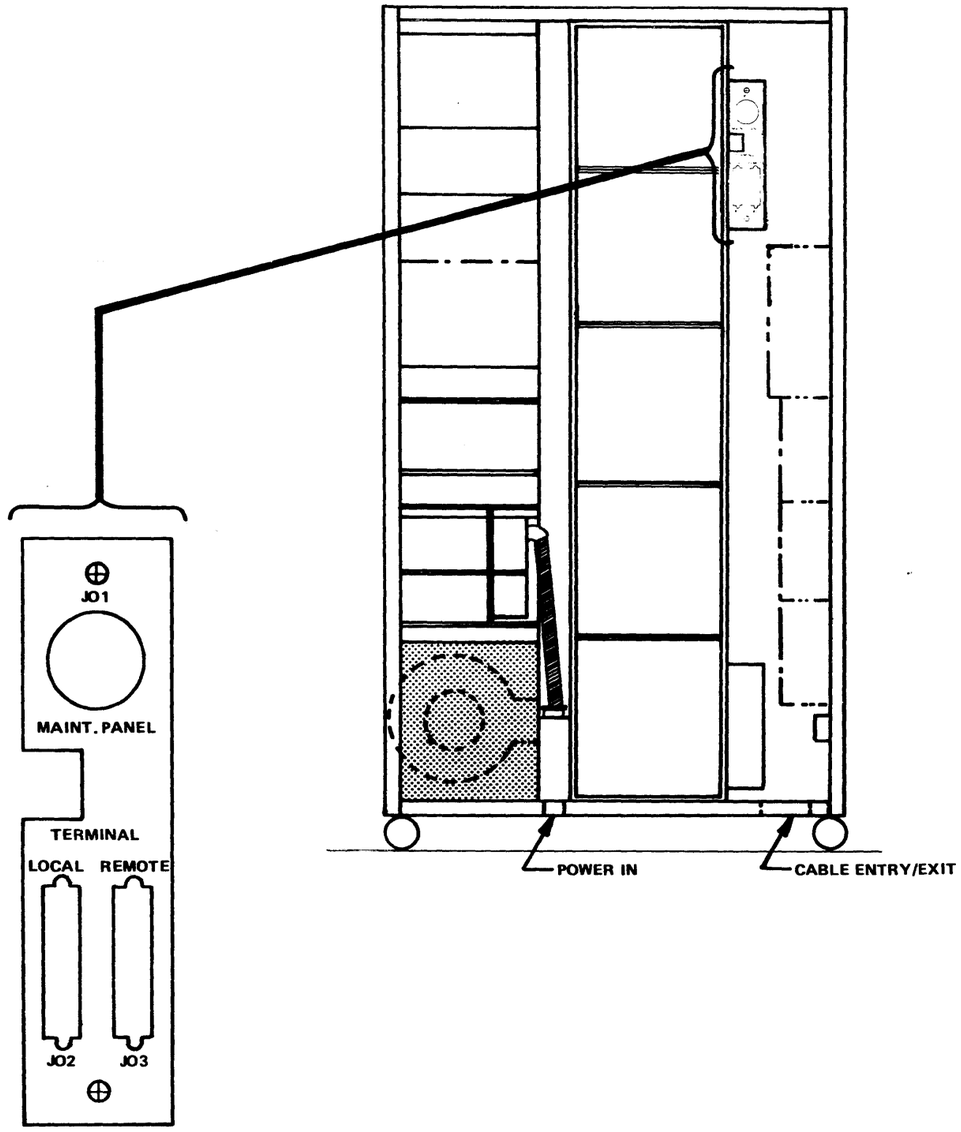
### 4.1.2 TERMINAL, LOCAL - J02

Connector J02 provides an interface between the DMP and a VIP Terminal. The VIP utilizes the microprocessor logic for dynamic testing to troubleshoot the MMU on a local, basis at the user's facility. If it becomes necessary to introduce additional T&D routines, the DPU may be plugged into J02 and the VIP plugged into the DPU. In this configuration, all diagnostic routines necessary to fault isolate the MMU may be introduced.

### 4.1.3 TERMINAL, REMOTE - J03

Connector J03 provides the same interface capabilities as the J02, except on a remote basis via MODEM to enable fault isolation from a remote location, such as the TAC Center, while the customer's equipment is on line. In this case, the customer's DPU is accessed via MODEM with the equipment operator reconfiguring the system if necessary.

For details in troubleshooting the MMU, refer to 58009928 Test and Repair Manual.




**58056792 CONNECTOR PLATE**