XEROX

Price $5.00

SYSTEM CONTROL UNIT (SCU)

Reference Manual
(Preliminary)

98 04 27A May 1973

Xerox Computer Systems

Western Technology Center
701 South Aviation Boulevard
El Segundo, California 90045
213 679-4511

Eastern Technology Center
1701 Research Boulevard
Rockville, Maryland 20850
301 424-6700

TABLE OF CONTENTS

APPENDIX

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I
GENERAL DESCRIPTION

1-1 INTRODUCTION

The Xerox System Control Unit (SCU) is a highly modular, microprogrammable data processor organized to provide a flexible, efficient and economical solution to a broad variety of systems or control tasks. It is designed to interface to a central computer, to peripheral and analog devices, and to many kinds of line protocol.

Because of the modular nature and the exclusive three-bus architecture, the SCU can be configured in many forms, from a simple device controller to (with the addition of a translator function) a complete computer emulation system.

1-2 GENERAL ARCHITECTURE

The Xerox System Control Unit consists basically of input-output interfaces, general registers, micro-control elements, an arithmetic logic unit, and scratch pad/main memory. These elements are connected between three 16-bit data buses used in common: A-Bus, B-Bus and C-Bus. (Refer to Figure 1-1.) To this basic system many other capabilities can be added, including up to 64K words of data memory, I/O controllers, special arithmetic functions, and hardware emulation logic.

Up to 128 input/output interfaces can be "plugged" in to the SCU directly to the three buses.

The eight general registers can be used interchangeably as accumulators or as index registers. They can be used to store intermediate results, or to move data from the C-Bus to the A- or B-Buses.

The micro control elements include the control memory, micro address register, and micro control register. The control memory holds a sequence of micro instructions. The micro address register provides the micro address of the location in control memory of the next micro instruction. The micro control register holds the current micro instruction while it is being executed.

The arithmetic logic unit is capable of 32 arithmetical or 16 logical operations, taking its two operands from the A-Bus and B-Bus and placing the results on the C-Bus.

The optional scratch pad/main memory provides up to 1024 words of high-speed storage at a system cycle rate of 350 nanoseconds, and from 4096 to 65,535 words of slower storage at a 700-nanosecond rate. The scratch pad can be used for intermediate storage beyond the limits of the eight general registers or for other purposes. The main memory can be used to store data, results, micro instructions awaiting transfer to a variable control memory, or macro instructions in the emulation mode.

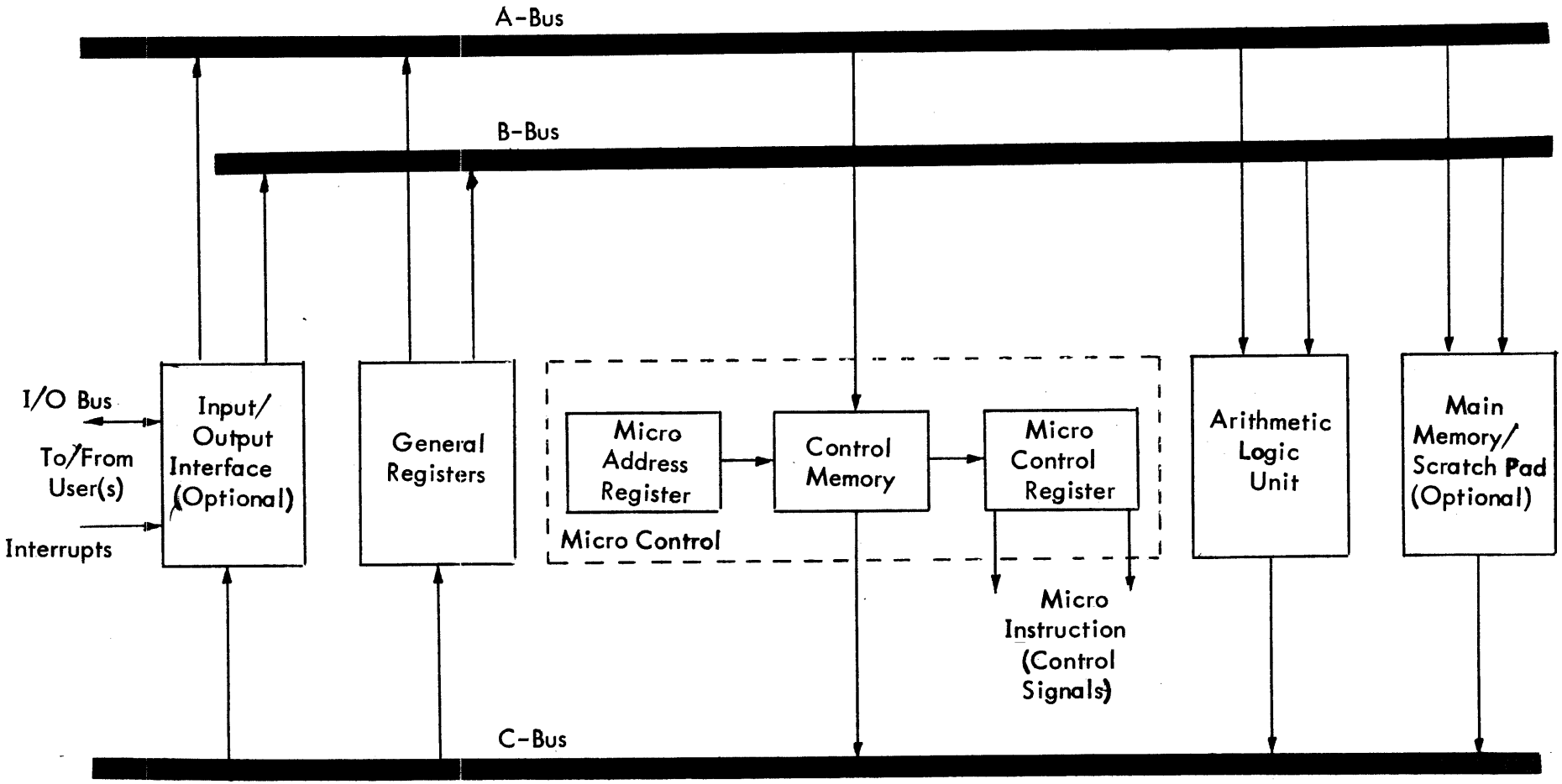


Figure 1-1. Simplified Block Diagram, Systems Control Unit

1-3 MICRO INSTRUCTIONS

The 32-bit micro instruction is divided into a number of fields that control all the operations of the machine. For example, the micro instruction determines the source of data for the A-Bus and B-Bus, whether it is from an input/output interface of from a general register. It determines the function to be performed in the arithmetic logic unit on data from the A- and B- Buses. It determines that data from the A-Bus is to be stored in scratch pad/main memory at an address contained on the B-Bus. Complete definitions of the micro instruction fields are described in Section 3.

It is apparent from Figure 1-1 that the micro instructions can direct the flow of data through many paths. Incoming data from the input/output bus, for example, can be routed directly to scratch pad/main memory, or can first be operated on in the arithmetic logic unit. Results from the arithmetic logic unit can be transmitted immediately via an input/output register to the input/output bus, or can instead be held in a general register pending further manipulation within the machine. Similarly, information from scratch pad/main memory can be routed to the input/output bus or to a general register.

The general registers are often used to buffer data between the C-Bus and the A-Bus or B-Bus for further processing or storage within the machine. In a somewhat similar way, the arithmetic logic unit can also be used as a path to move data unchanged from the A- or B- Buses to the C-Bus.

Thus, multiple fields enable the micro instruction to control a number of operations and to address several locations simultaneously. A tightly packed micro instruction provides parallel processing and results in efficient machine utilization.

1-4 FEATURES

The generalized input/output structure and the corresponding micro fields permit the machine to interface all kinds of devices and buses through either standard or custom-designed input/output modules.

External connections are made with up to four 14-conductor cable-connector assemblies on the front edge of each input/output module.

Two fast interrupts provide high-speed input/output data transfer within the cycle time of a single micro instruction (350 nanoseconds).

A very fast general multiplexed interrupt structure is provided. Internal status and the micro address of the micro instruction are nested up to 16 times in push/pull stack, permitting the control micro program to branch to a higher priority task, and then return to a lower priority task.

The state of one or more external conditions for each input/output interface can be directly tested.

Direct coding of the multiple-field micro instruction achieves maximum speed and efficiency in data processing.

Three types of control memory provide permanence or flexibility in the firmware:

1. Fixed-- in a read only memory (PROM)

2. Alterable off-line-- in a plated-wire electrically alterable read only memory (EAROM)

3. Alterable on-line-- in a high-speed bipolar LSI random-access memory (RAM)

A field verification micro program verifies the operation of all configurations of standard and optional elements, except input/output, translators, and the maintenance control panel.

Classical single-address instructions (or macro instructions) can be emulated in micro instructions by the addition of a translator option.

Data can be manipulated on either a byte (8 bits) or a word (16-bits) basis.

1-5 PHYSICAL ARRANGEMENT

The basic System Control Unit is housed in a two-high Sigma mounting case that holds up to 32 double J-Series modules. This case holds all standard and optional modules and up to 16,384 words of main memory. Modules providing an additional 49,152 words of main memory may be housed in a second two-high Sigma mounting case. Additional input/output modules may be housed in a separate mounting case.

A fan case holding two fans is located below the mounting case. The power supply is contained in a separate chassis. All units are capable of being mounted in 19-inch cabinets with the use of additional hardware.

1-6 STANDARD AND OPTIONAL ELEMENTS

The basic unit is composed of the following standard elements:

- Chassis, backplane, and blower
- Arithmetic Logic Unit
- General Registers (8)
- Micro Address Register
- Micro Control Register

The basic unit, however, does not include the following elements necessary for minimum operation: Control Memory, and Input/Output, Power Supply. These and other options are described briefly in the following paragraph.

1-7 INPUT/OUTPUT MODULES

Locations are provided in the basic chassis to install up to six input/output modules. Additional input/output modules can be installed in an expansion chassis. A variety of input/output interfaces are available for use with the SCU. These interfaces are described in detail in Section 4.

A Teletype and High-Speed Paper Tape Input/Output Module interfaces an ASR, KSR, or RO teletype in either ANSCII or binary code, a high-speed (300 characters/second) paper-tape reader, and a high-speed (75 characters/second) paper-tape perforator. This input/output module also contains a real-time clock capable of 256 counts (4.2675 seconds) at 16.67 milliseconds/count (60 Hz).

1-8 CONTROL MEMORY, PROM

PROM memory provides 256 to 1024 32-bit words per module in 256-bit increments of programmed read-only memory for micro instructions. Locations are provided for up to four modules (4096 words). Parity checking capability is optional. At least one PROM module must be included in each configuration unless an EAROM plated-wire control memory is used.

1-9 CONTROL MEMORY, RAM

RAM memory provides 256 32-bit words per module of read/write random access memory for micro instructions. Locations are provided for up to four modules (1024 words). Parity checking capability is optional.

1-10 CONTROL MEMORY, EAROM

EAROM memory provides 1024, 2048, or 4096 32-bit words of electrically alterable (off line) read-only memory for micro instructions. Parity checking capability is included. This type of memory cannot be intermixed with PROM or RAM options. A portable or rack-mounted load box is required to alter the EAROM off line.

1-11 FIELD VERIFICATION MEMORY

Field verification memory provides 256 32-bit words of pre-programmed read-only memory that performs a go/no go quality check of the basic unit, control memory, scratch pad, and main memory, but not of input/output modules, translators, or maintenance control panel. This memory is required in units to be maintained by Xerox. Refer to Section 5.

1-12 SCRATCH PAD

Scratch pad memory provides 512 16-bit words per module of high-speed (350 ns) read/write random-access memory for data use. The chassis accommodates zero, one, or two modules, providing up to 1024 words, using addresses 0 - 1023 of main memory. Parity checking capability is optional.

1-13 MAIN MEMORY

Main memory provides 4096 16-bit words of 700 nanosecond read/write random-access memory for data use. The basic chassis accommodates up to 16,384 words in 4096-word increments. Additional memory increments up to 65,536 words can be installed in a memory expansion chassis. Parity checking capability is optional.

1-14 TRANSLATOR

The translator translates a macro instruction of a computer instruction set into a vector (micro address) and argument that accesses the first of a series of micro instructions that may be required to implement the macro instruction. With each translator a control memory containing emulation firmware and input/output modules is required, as well as a scratch pad/main memory in sufficient size to run desired programs.

1-15 MAINTENANCE CONTROL PANEL

The maintenance control panel provides controls and indicators for program debugging and manual troubleshooting. It also provides autoload switch to trap to autoloading micro program, indicates register contents and machine status and shows data to be inserted into registers manually. Refer to Section 5.

1-16 POWER SUPPLY

The power supply provides +5 volts ±5 percent at 70 amperes, ±15 volts ±3 percent at 2 amperes. Input power requirements are 120 vac, 60 Hz, or 220 vac, 50 Hz.

1-17 CONFIGURATIONS

The available options enable the user to configure for a variety of applications. A minimum configuration, however, must include (in addition to the standard modules) at least one input/output module and one control memory module. With this configuration it is possible to implement a small fixed control program with minimal communication with the external world. By the addition of more modules of control memory and input/output, it is possible to expand the micro control program and external communications.

The addition of a scratch pad module increases the scope of high-speed data processing and arithmetical and logical data manipulation. Also, implementation of main memory provides much larger amounts of data storage, permitting expanded program capabilities. In addition to the conventional uses of main memory, this memory can store blocks of micro instructions for transfer into control memory RAM when needed, thereby adding to the scope of the control program.

The addition of a translator, together with an emulation micro program in control memory, permits the machine to emulate a computer instruction set. In addition to these configurations, the generalized three-bus structure of the System Control Unit makes it possible to add other functions as further needs are formulated, such as high-speed multiply/divide, square root, trigonometic functions, and code conversion. Additional functions can be designed to take input from one or two buses and output to the third bus under the control of appropriate micro instruction fields.

1-18 APPLICATIONS

The flexibility of the unit's input/output scheme facilitates connection to other units in a system. The addition of optional modules makes it possible to install only those capabilities needed by the application. With appropriate input/output modules, the System Control Unit can be interfaced to several computers, teletypes, high-speed paper-tape readers/punches, discs, analog-to-digital converters, digital-to-analog converters, digital inputs/outputs, communications modems, and other peripheral devices.

Because the SCU is a general-purpose data processor, it can be used for a wide variety of applications; for example,

Engineering Test, Scientific Experiment

- Data acquisition and test control
- Preliminary data sorting and computation
- Data compression and buffering
- Transmission of intermediate results to central computer
- Control of A/D and D/A converters

Process Control

- Logging
- Limit checking and flagging

Conversion to engineering units, scaling

Outputs to process controllers

Outputs to monitors and displays

Machine Control, Program Control

Complex machine or process direction through a sequence of activities

Video Display

Communications interfacing

Refresh memory for display clusters

Supervisory and editing console

Communications Controller

Message switching and storage

Formatting and forwarding

Code conversion

Monitoring and reporting

1-19 SOFTWARE

A number of software packages may be used with the System Control Unit.

1-20 MICRO CROSS ASSEMBLER

The Micro Cross Assembler (Catalog Number 706 450) enables a user to prepare micro programs in a simplified manner using mnemonics instead of binary digits. Each micro instruction takes the form of a command, represented by a mnemonic, followed by several arguments. For example, the instruction LOAD R2, R1, A, reads the contents of the scratch pad/main memory location addressed by the content of general register 1 to general register 2. Approximately 60 commands are available in this assembler.

The mnemonic instructions are assembled by the Xerox Meta Symbol assembly system on a computer of the Sigma 5 - 9 series into binary object code in the form required for entry into control memory. The micro program printout from the Sigma computer shows the micro instructions in both mnemonic form and as hexadecimal digits for the eight 4-bit fields of the 32-bit micro instruction.

1-21 INTERPRETER/SIMULATOR

The Sigma 5 - 9 SCU Interpreter (Catalog No. 706437) enables a user to run a set of SCU micro instructions on a computer of the Sigma 5 - 9 series, with 52K bytes of storage, employing the Batch Processing Monitor or the Universal Timesharing System. The Interpreter simulates a System Control Unit. The user can initialize registers and push stack and specify inputs to the SCU program from outside devices.

The Interpreter provides tracing capability, main memory dump at the end of a run, and the time required to run the micro program on the SCU. The tracing capcbility prints out for each micro instruction such information as content of the three buses, status bits, and the contents of the registers affected by the micro instruction. Thus, a control memory micro program can often be verified or debugged by means of the Interpreter before the control memory PROM's are implemented on an SCU.

BOOTSTRAP

The SCU Bootstrap Loader (Catalog No. 880603) is the most primitive program in the software package. It is used to load the SCU Loader Program.

The Bootstrap Loader program is available in two forms: teletype and high-speed paper-tape reader. Both forms consist of nine micro instructions. If control memory RAM is installed, these micro instructions can be loaded manually from the maintenance control panel; or the Bootstrap Loader can be permanently located in fixed control memory ROM.

1-22 LOADER

The SCU Loader Program (SCULE) (Catalog No. 880600) loads paper-tape data from an ASR 33/35 teletype into control memory RAM or into main memory. The Loader Program occupies 153 words of control memory. It may be loaded into a control memory RAM using the Bootstrap Loader, or may be fixed in a control memory ROM.

1-23 SPECIFICATIONS

Specifications for the System Control Unit are provided in Table 1-1.

Table 1-1. System Control Unit Specifications

Data Buses	A, B, C
Data Bus Width	16 bits
Byte Operation	8 bits
Work Operation	16 bits
Micro Instruction	32 bits
Micro Instruction Cycle Time	350 nanoseconds, including scratch pad access
Main Memory Access Time	700 nanoseconds
Input/Output	Four ET10 14-conductor cable-connector assemblies per input/output module
Standard Input/Output	TTL compatible open-collector outputs, ground level true
Interrupts (in order of priority)	Control Memory Location (Hexadecimal) / Interrupt X'4' — Power Off X'5' — Power On X'6', X'7' — Fast interrupts X'8' — Multiplexed IO X'9' — Error (in control memory, scratch pad, main memory) X'A' — Autoload with Run switch Off X'B' — Autoload with Run switch On. Serves as Console Interrupt
Mounting Case	15.75 inches high, (including fans), 19 inches wide, 7.80 inches deep
Weight	35 pounds
Power Supply Dimensions	6.25 inches high, 17.06 inches wide, 13.50 inches deep
Weight	70 pounds
Power	120 vac, 60 Hz, 7.5 amp. 220 vac, 50 Hz, 4.0 amp.

Table 1-1. System Control Unit Specifications (Continued)

Environmental	
Temperature	10°C to 50°C, operating
Humidity	10% to 90%, without condensation

SECTION II
SYSTEM ORGANIZATION

2-1 INTRODUCTION

This section describes the flow of micro addresses, micro instructions, and data throughout the system structure. In Section 3, this flow is related specifically to the various fields of the micro instruction.

The System Control Unit consists of three types of elements: clocked registers or memory units; combinational gating structures, such as 2:1 multiplexers or the arithmetic logic unit; and the three bus structures, A, B, and C.

The system has a single clock period of 350 nanoseconds, with no clock phases. Thus, each clocked element assumes a state at one clock pulse and holds that state until the next clock pulse appears. During the clock period, the state of each register passes through the intervening gating structures and buses and is ready to enter the following register at the next clock pulse.

There is one exception to operation at clock speed, however. The main memory and reads/writes at a 700-nanosecond rate, or two clock periods. Therefore, when a main-memory operation is specified, one clock is dropped and the intervening-time period between the two clock pulses is 700 nanoseconds.

2-2 DATA FORMATS

The System Control Unit has three formats: micro addresses, micro instructions, and data.

The micro address consists of 12-bits capable of addressing up to 4096 control memory locations. The micro address register, micro address push stack, and related gating operate on a 12-bit basis. When a micro address is routed over one of the three buses, it occupies the least significant 12 positions of the 16-bit positions.

The micro instruction consists of 32-bits. The control memory and the micro control register operate on a 32-bit basis. When a micro instruction is routed over one of the buses, it is handled in two 16-bit sections.

Data is processed on either a word (16 bits) or byte (8 bits) basis. Input/output interfaces, general registers, the arithmetic logic unit, scratch pad, main memory, and the three buses operate on either a word or byte basis.

2-3 INFORMATION FLOW

Information flow refers to the path data takes from one register via gating and buses to another register. The flow of information will be described in four categories: micro addresses, micro instructions, data, and interrupts.

2-4 MICRO ADDRESSES

The micro addresses are normally held in the micro address register. From the micro address register they are applied via the 4:1 memory address multiplexer to select the control-memory location in which a micro instruction is stored. The micro address in the micro address register can be changed in two ways:

a. Incremented to the next micro address by the clock.

b. Jumped to another micro address supplied from the C-Bus. The flow of micro addresses is shown in Figure 2-1.

The micro address applied to control memory can be multiplexed from three sources in addition to the micro address register, the A- and B-Buses, and the interrupt logic. The interrupt logic generates micro addresses corresponding to each interrupt. These micro addresses are listed in Section 1.

The current micro address in the micro address register can be stored in the push stack and replaced with a micro address from the C-Bus. Also, the micro address from the push stack can be routed via the C-Bus to the micro address register. At the same time the micro address is stored in the push stack, six status bits are entered into the push stack.

2-5 MICRO INSTRUCTIONS

The micro instructions are stored in three types of control memory: programmable read only memory (PROM), random access memory (RM), and electrically alterable read only memory (EAROM). In addition, the field verification memory can be considered to be a type of control memory. Micro instruction processing is described in Figure 2-2.

When addressed by a micro address, a micro instruction is transferred from control memory via a 2:1 multiplexer to the micro control register. A micro instruction can also reach the micro control register from the maintenance control panel via the C-Bus and the 2:1 multiplexer in two 16-bit increments.

In the case of the random-access type of control memory, a new micro instruction can be written into the location addressed by the micro address. The new micro instruction is obtained in two 16-bit increments from the A-Bus. A micro instruction can be read out from any type of control memory via a 2:1 multiplexer in two 16-bit increments to the C-Bus.

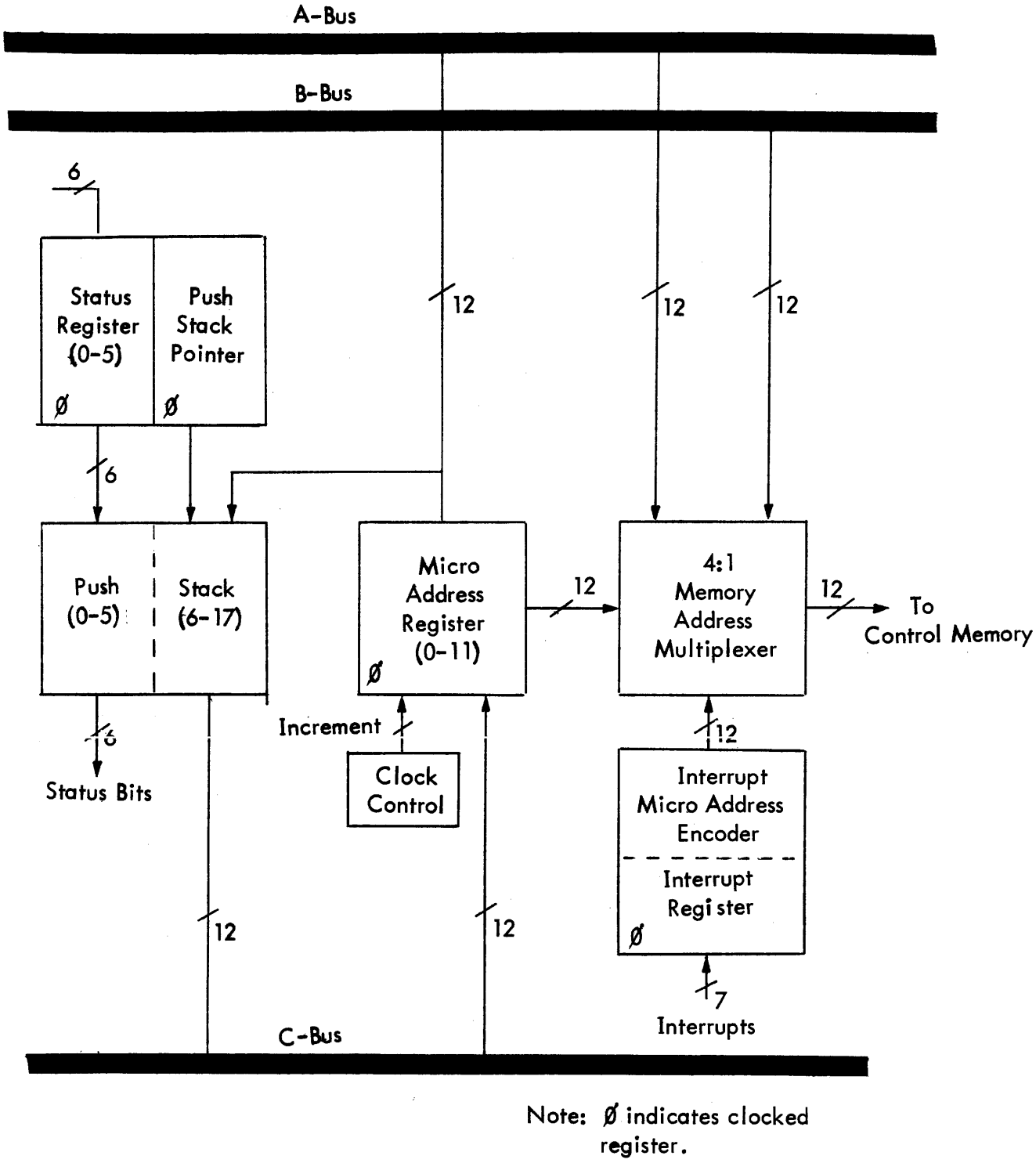
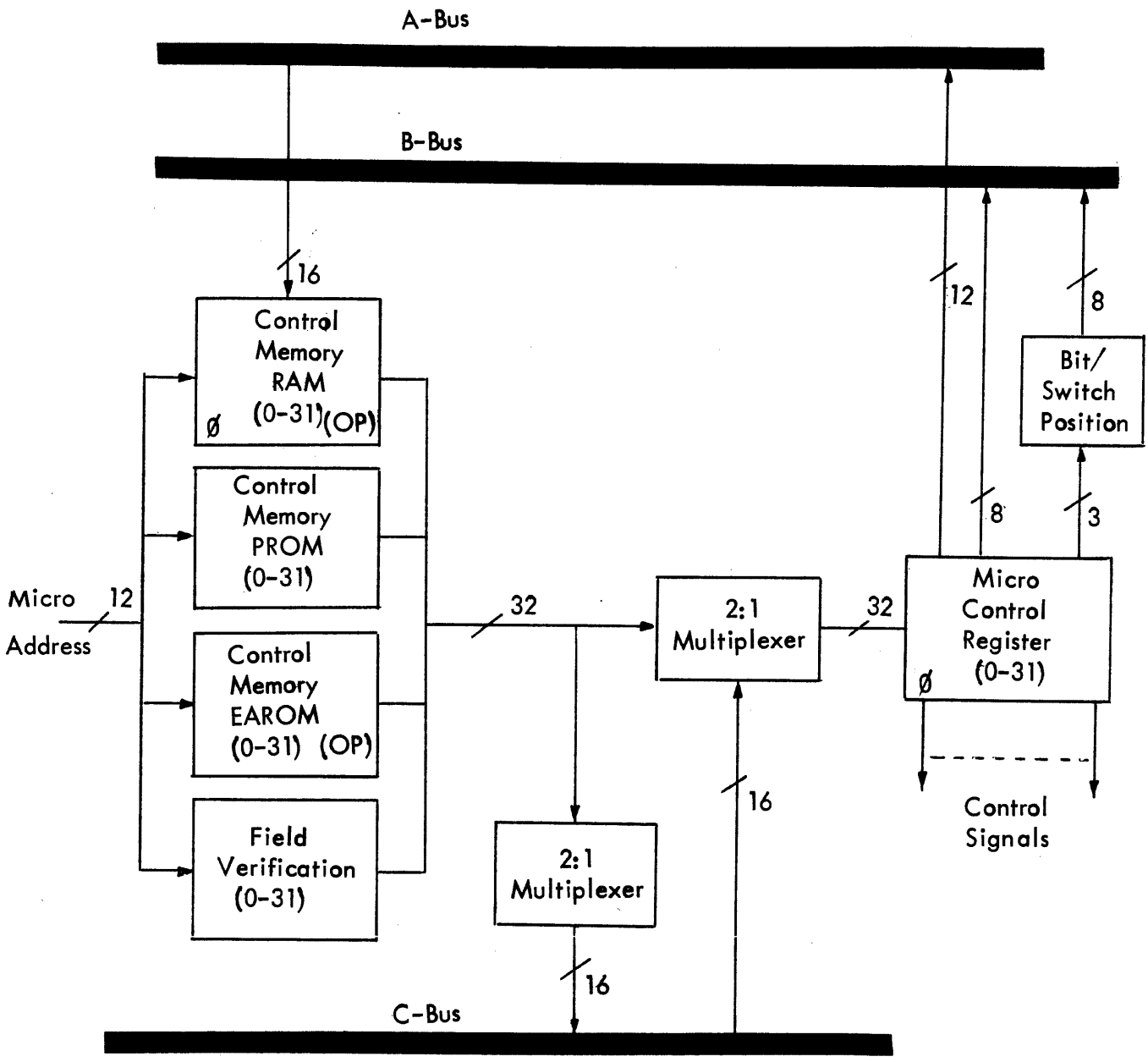


Figure 2-1. Block Diagram, Micro Address Generation



Note: Ø indicates clocked registers

Figure 2-2. Block Diagram, Micro Instruction Processing

From the micro control register the micro instruction is decoded to provide control signals that direct the flow of data through the system, as detailed in Section 3. The last 12 bits of the micro instruction, when used as an immediate micro address, are placed on the A-Bus. Similarly, the last eight bits of the micro instruction, when used as the Emit Field, are placed on the B-Bus.

2-6 DATA FLOW

Data flows through a number of paths and registers: the arithmetic logic unit, the general registers, input/output modules of various kinds, scratch pad, main memory, and the instruction translator.

2-7 ARITHMETIC LOGIC UNIT

The arithmetic logic unit (ALU) accepts a byte or word from the A-Bus and the B-Bus, performs arithmetic or logic operations on the two operands, and outputs the results through shift/byte select logic to the C-Bus, as shown in Figure 2-3. Operand word or byte selection, operation selection, and output shift or byte select are all specified by control signals derived from the micro instruction.

Input byte select applies both bytes, left byte, or right byte, right justified, from the A-Bus to the A operand or from the B-Bus to the B operand.

Arithmetic or logic operation is specified by the Mode signal, M. One of 16 arithmetic or logic operations is selected by the four Select lines, S. Carry-in, Cn, adds one to the result.

The ALU result can be shifted left or right one bit position, or transferred directly to the C-Bus. Also the right byte of the result can be placed in the left byte of the C-Bus.

2-8 GENERAL REGISTERS

The eight general registers provide a source for the A- and B-Bus and a destination for the C-Bus. See Figure 2-3. For example, the A and B operands for the ALU can be taken from general registers and the ALU result can be stored in a general register.

Moreover, the value on the B-Bus can be either incremented or decremented by one and the result stored in one of the general registers.

2-9 STATUS BITS

Six Status bits simplify the checking of results by automatically providing information on current operations without requiring extra instruction execution. Indicators on the maintenance control panel showing the status bits simplify maintenance and debugging operations. Four of the six status bits are derived from arithmetic operations. See Figure 2-3. The status bits are: Carry, Overflow, C-Bus=0, Sign, Enable I/O Interrupt, and Control Mode.

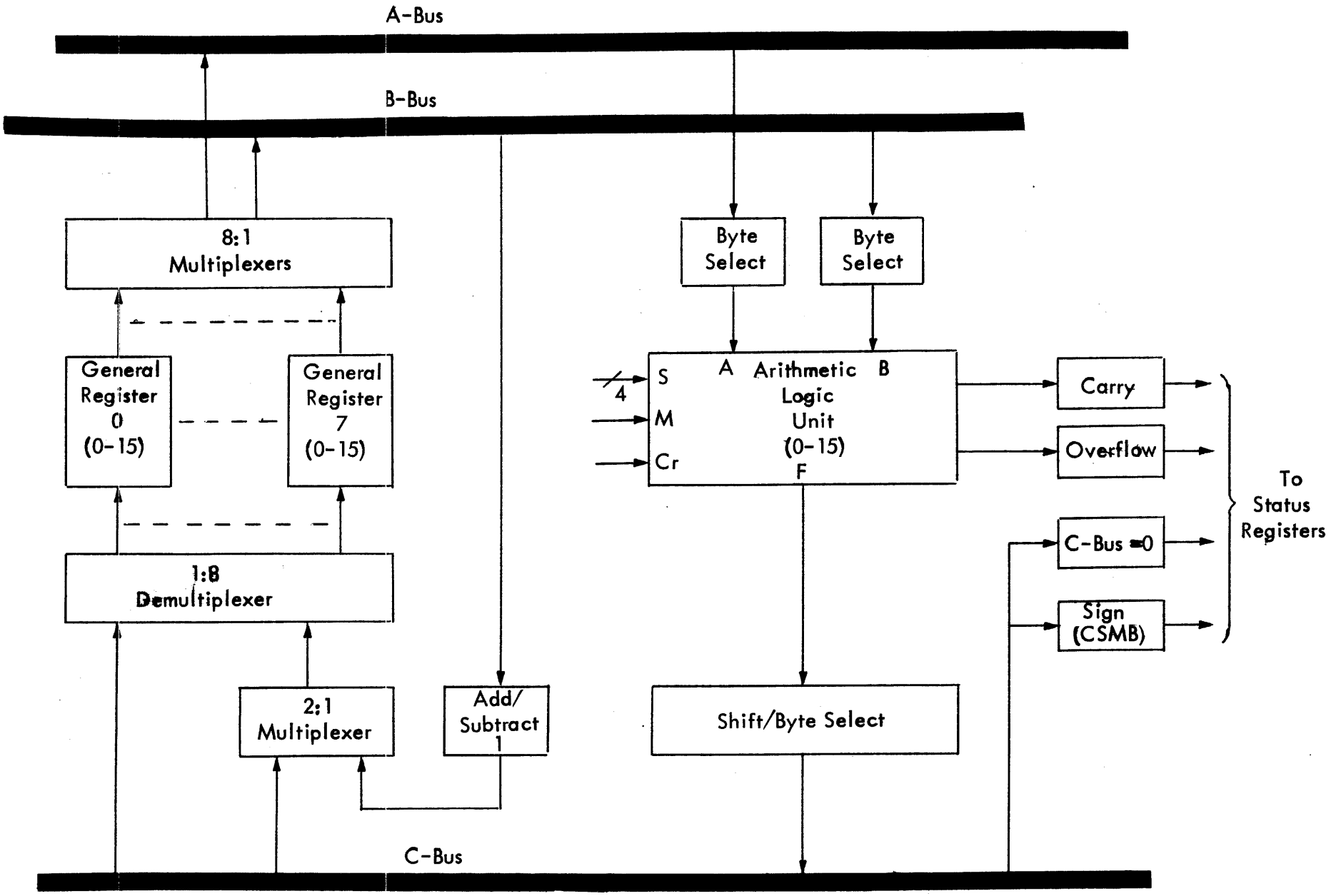


Figure 2-3. Block Diagram, ALU and General Registers

Carry. In word operation, the carry-out from the most significant bit of the ALU (bit 0), or in byte operation, the carry-out from the most significant bit of the byte (bit 8).

Overflow. An arithmetic operation on A and B operands results in a number greater than the largest number than can be processed in the space specified; word or byte. For example, in two's complement format overflow occurs if the sum of two positive numbers is negative or if the sum of two negative numbers is positive. Conversely, overflow results if the subtraction of a negative number from a positive number gives a negative result, or if the subtraction of a positive number from a negative number gives a positive result.

C-Bus = 0. All bits of the C-Bus are zero.

Sign. In word operation, the most significant bit of the word (bit 0) or, in byte operation, the most significant bit of the byte (bit 8).

Enable I/O Interrupt. A command decoded from the micro instruction sets and resets this bit.

Control Mode. A command decoded from the micro instruction sets and resets this bit. It is used for additional control of an optional function, such as the macro instruction translator.

2-10 INPUT/OUTPUT

Input/output modules may be of various types; for example, the standard input/output module or the teletype paper-tape module. In general, these modules perform such functions as:

1. Accepting incoming data and placing it on the A-Bus.
2. Transferring outgoing data from the C-Bus to the output.
3. Accepting multiplexed I/O interrupts and placing their identification on the A-Bus.

Input/output operations and capabilities are covered in Section 4.

Figure 2-4 is a complete block diagram, including a simplified form of input/output.

2-11 SCRATCH PAD/MAIN MEMORY

Data is written from the A-Bus into the scratch pad/main memory at the location addressed by the B-Bus. Data is transferred to the C-Bus from the scratch pad/main memory location addressed by the B-Bus.

2-12 TRANSLATOR

The translator enables the SCU to emulate a computer instruction set. This option provides the means for translating a macro instruction (taken from the C-Bus) into a micro address vector

(placed on the A-Bus) that accesses the first of a series of micro instructions in control memory necessary to implement the macro instruction. Moreover, in the case of a macro instruction containing an argument field, such as the address field used in calculating direct or indirect memory locations, the translator transfers the argument field to the B-Bus.

2-13 MAINTENANCE CONTROL PANEL

The optional maintenance control panel provides controls and indicators that display the current status of the machine. Change of status can also be made from the maintenance control panel. The controls permit 16 data bits to be placed on the C-Bus, in the micro control register, or 12 bits in the micro address register. Interrupt and status bits are displayed continuously. Register or bus contents can be displayed one at a time. The capabilities of the maintenance control panel are defined in more detail in Section 5.

2-14 INTERRUPT FLOW

Seven interrupt lines originate in various parts of the unit, as shown in Figure 2-5, and terminate at the interrupt register. When one or more of the interrupt flip-flops is set, the micro address of the interrupt that has the highest priority is applied via the memory address multiplexer to the control memory. The interrupts are shown in order of priority by micro addresses, x'4' to x'B'. Each micro address points to a micro instruction specified by the system programmer initiating a procedure that responds to the interrupt.

The contents of the interrupt register (except Autoload) are displayed on the maintenance control panel.

The interrupts perform the following functions:

Power Off. A logic signal from the power supply that indicates loss of input power (50 percent or below) 4 milliseconds or less after input power has dropped. There is sufficient storage in the +5 volt supply to maintain voltage regulation until two milliseconds after the power off signal has gone true. At this time, main memory goes into battery backup, locking out further access.

Power On. A logic signal from the power supply that indicates that power is on and the +5 volt logic supply is available. This signal is delayed at least 100 milliseconds after the supply is turned on.

Fast Interrupt 1. A logic signal from the input/output interface directly to the interrupt register used primarily for high-speed data transfers to or from user logic via the I/O interface to or from scratch pad/main memory. The interrupt register returns a hardware response to the I/O interface.

Fast Interrupt 2. Similar to Fast Interrupt 1, except with a lower priority. One of the fast interrupts may be used to input data, while the other is used to output data. The fast interrupts are described in Section 4.

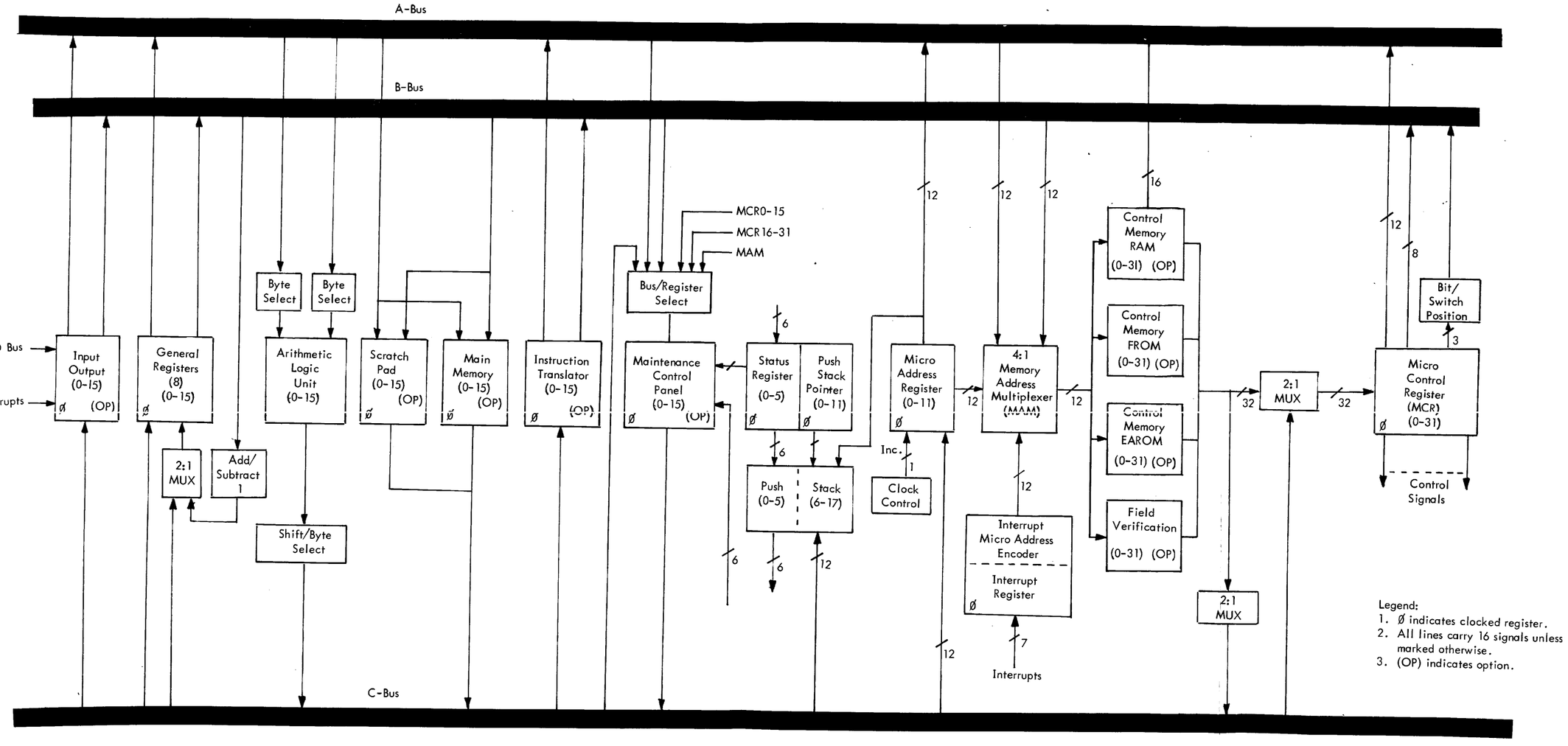


Figure 2-4. Block Diagram System Control Unit

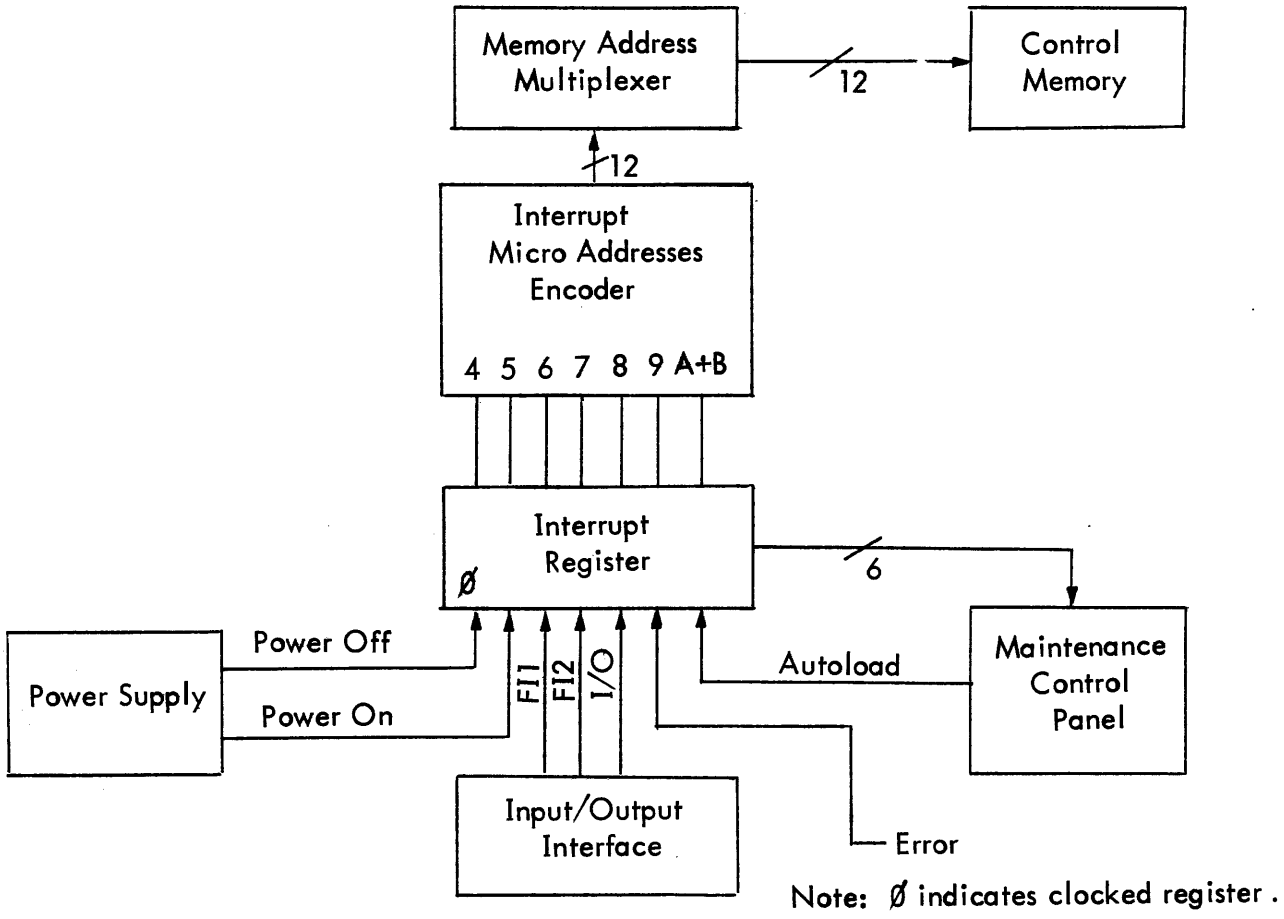


Figure 2-5. Block Diagram, Interrupt Flow

Input/Output. A logic signal that indicates that a multiplexed input/output interrupt is pending on one or more of the input/output interfaces. In response to this signal, the system programmer requests the highest priority I/O module with an interrupt pending to put its address on the A-Bus.

Error. A logic signal that indicates a parity error has occurred in control memory, scratch pad, or in main memory.

Autoload. A switch signal from the maintenance control panel used to initiate an automatic loading routine. This signal effects a trap to control memory location X'A' if the Run switch is off, or to location X'B' if the Run switch is on. This second use of Run switch serves as a Console Interrupt.

SECTION III
MICRO INSTRUCTION REPERTOIRE

3-1 INTRODUCTION

This section describes the actions that result from each micro opcode of each field of the micro instruction.

The micro instruction consists of 32 bits, divided into a number of single-purpose and multiple-purpose fields, as illustrated in Table 3-1. These fields are decoded to provide control signals for all other elements of the machine. The micro program is stored in control memory; however, during the execution of a particular micro instruction, it is held in the micro control register.

3-2 CONFIGURATION

The configuration shown in the block diagram, Figure 2-4, is summarized in the following listing.

Bus (No. of bits)

A(0-15)	A-Bus
A(0-7)	A-Bus, Left Byte
A(8-15)	A-Bus, Right Byte
B(0-15)	B-Bus
B(0-7)	B-Bus, Left Byte
B(8-15)	B-Bus, Right Byte
C(0-15)	C-Bus
C(0-7)	C-Bus, Left Byte
C(8-15)	C-Bus, Right Byte
IO(0-15)	Input/Output Bus
IO(0-7)	IO Bus, Left Byte
IO(8-15)	IO Bus, Right Byte

Registers (No. of bits)

IORn(0-15)	Input/Output Register (n=0-127)
GRn(0-15)	General Register (n=0-7)
MAR(0-11)	Micro Address Register
MCR(0-31)	Micro Control Register
IR(0-6)	Interrupt Register
SR(0-6)	Status Register
PSn(0-17)	Push Stack (n=0-15)
T(0-15)	Translator

Table 3-1. Micro Instruction Field

0	1——3	4	5——7	8 9	10 11	12———15	16———19	20—23	24————27	28—31
IOM	RA SELECT	ALUM	RB SELECT	A-BUS CONTROL	B-BUS C ONTROL	C-BUS/REG CONTROL	BRANCH CONTROL	ALU SELECT	MCR CONTROL	DEVICE SELECT
			BIT SWITCH					OTHER FUNCTIONS	MCR=X'A'	DEVICE SELECT
								ALU SELECT	INPUT/OUTPUT CONTROL	DEVICE SELECT
								ALU SELECT	EMIT FIELD	
								IMMEDIATE ADDRESS FIELD		

Memory

Control memory	CM-RAM(MAM) = CM(0-1024, 0-31) CM-ROM(MAM) = CM(0-4096, 0-31) CM-EAROM(MAM) = CM(0-4096, 0-31)
Field Verification	FV(MAM) = CM(x'F00'-x'FFF', 0-31)
Scratch Pad	SP(B) = SP(0-1024, 0-15)
Main Memory	MM(B) = MM(0-65,536, 0-15)
PHI (or ϕ)	One-phase clock (350 nanosecond period) (One clock dropped for main memory access)

In addition to the terms listed above, the following symbols and mnemonics are used in this section in the description of the micro instruction operation.

ABC	A-Bus Control
ALU	Arithmetic Logic Unit
ALU-A	A Operand inputs to ALU
ALU-B	B Operand inputs to ALU
ALUM	ALU Mode
ALUS	ALU Select
ARG	Argument from translator
BBC	B-Bus Control
BC	Branch Control
CBRC	C-Bus/Register Control
CMSB	C-Bus Most Significant Bit
Cn	Carry-in to ALU
DA	Device Address
DS	Device Select
EF	Emit Field
I	Interrupt
IAF	Immediate Address Field
IOC	Input/Output Control
IOI	Input/Output Interrupt
IOM	Input/Output Mode
MAR	Micro Address Register
MAM	Memory Address Multiplexer
MCP	Maintenance Control Panel
MCR	Micro Control Register
MCRC	Micro Control Register Control
OF	Other Function
RAS	Register to A-Bus Select
RBS	Register to B-Bus Select

S	Status Bit
SP	Scratch Pad/Main Memory
SR	Status Register
Vn	Vector n from Translator (a micro address)
X'n'	Hexadecimal number
NX'n'	Not hexadecimal n
plus	Add to
minus	Subtract from
+	OR Function (bit n+1 means 1 greater than n)
.	AND Function
⊻	Exclusive OR Function
$\overline{n}$	1's complement of number
⇐	Is replaced with

3-3 SINGLE-PURPOSE MICRO FIELDS

The specific function of each single-field micro opcode is defined. Unless otherwise stated, execution of each micro instruction occupies one clock period (350 nanoseconds), and each micro instruction updates the status register.

3-4 INPUT/OUTPUT MODE

Input/Output Mode (IOM) selects either general registers or input/output data.

IF IOM = 0, general registers are selected;
If IOM = 1, input/output modules are selected.

3-5 RA SELECT

Unless specifically inhibited, this field sources a general register or input/output module to the A-Bus. When the IO Mode is false (IOM=0), RA Select (RAS) contains the address of a general register selected to be the source of data for the A-Bus. When the IO Mode is true (IOM=1), RA Select (bits 1-3 of the micro instruction) and Device Select (bits 28-31 of the micro instruction) contain the address of an input/output module that is to be the source of data for the A-Bus.

3-6 ALU MODE

ALU Mode (ALUM) causes the arithmetic logic units to perform arithmetic functions when false, and logic functions when true. This mode is covered in greater detail in connection with ALU Select.

3-7 RB SELECT

RB Select (RBS) contains the address of a general register that is to be the source of data for the B-Bus. In addition, when IOM=1 and B-Bus Control=3, the RB Select field assigns a bit to one of the eight least-significant positions positions of the B-Bus. This operation is covered in more detail in subsection 3-19, Bit Switch/Position.

3-8 A-BUS CONTROL

A-Bus Control (ABC) in concerned with a) the source of data placed on the A-Bus, b) the byte applied to the A operand inputs of the ALU, and c) the bytes read from or written into main memory.

ABC=0

The contents of the general register or input/output module specified by IO Mode, RA Select, and Device Select are placed on the A-Bus. Both bytes of the A-Bus are applied to A operand of the ALU. In addition, both bytes of the A-Bus may be read from or written into main memory under control of Branch Control (BC).

ABC=	IOM=	BC=	
0	0		A(0-15) ⇐ GRn(0-15); ALU-A(0-15)⇐ A(0-15)
0	1		A(0-15) ⇐ IOn(0-15); ALU-A(0-15) ⇐ A(0-15)
0	X	2	C(0-15) ⇐ MM(0-15)(B)
0	X	3	MM(0-15)(B) ⇐ A(0-15)

The functions performed by BC=2 or 3 are in addition to the functions performed by ABC=0 and IOM=0 or 1.

ABC=1

The contents of the general register or input/output module specified by IO Mode, RA Select and Device Select are placed on the A-Bus. The left byte of the A-Bus is applied to A operand of the ALU, right justified. In addition, the left byte of the A-Bus may be read from or written into the left byte of main memory under control of Branch Control (BC).

ABC=	IOM=	BC=	
1	0		A(0-15) ⇐ GRn(0-15); ALU-A(8-15) ⇐ A(0-7);ALU-A(0-7) ⇐ 0
1	1		A(0-15) ⇐ IOn(0-15); ALU-A(8-15) ⇐ A(0-7); ALU-A(0-7)⇐ 0
1	X	2	C(0-7) ⇐ MM(0-7)(B)
1	X	3	MM(0-7)(B) ⇐ A(0-7)

The functions performed by BC=2 or 3 are in addition to the functions performed by ABC=1 and IOM=0 or 1.

ABC=2

The contents of the general register or input/output module specified by IO Mode, RA Select and Device Select are placed on the A-Bus. The right byte of the A-Bus is applied to A operand of the ALU, right justified. In addition, the right byte of the A-Bus may be read from or written into the right byte of main memory.

ABC=	IOM=	BC=	
2	0		A(0-15) ⇐ GRn(0-15);ALU-A(8-15) <=A(8-15);ALU-A(0-7) <=0
2	1		A(0-15) ⇐ IOn(0-15); ALU-A(8-15) ⇐ A(8-15); ALU-A(0-7) ⇐ 0
2	X	2	C(8-15) ⇐ MM(8-15)(B)
2	X	3	MM(8-15)(B) ⇐ A(8-15)

The functions performed by BC=2 or 3 are in addition to the functions performed by ABC=2 and IOM =0 or 1.

ABC=3

The contents of the micro address register are placed on the A-Bus, right justified. Both bytes of the A-Bus are applied to the A operand of the ALU. Transferring the contents of a register to the A-Bus is inhibited.

ABC=	
3	A(4-15) ⇐ MAR(0-11); ALU-A(0-15) ⇐ A(0-15)

A micro opcode in one of the C-Bus/Register Control micro opcodes is needed to complete A-Bus Control. A-Bus Control can be overridden by MCR Control=3, Read Vector 1 to A-Bus from Translator (optional); MCR Control=5, Read Vector 2 to A-Bus from Translator (optional); MCR Control=E, Read Interrupt Address to A-Bus; and C-Bus/Register Control=B, Immediate Addressing Invoked to MAR.

3-9 B-BUS CONTROL

B-Bus Control (BBC) is concerned with the source of data placed on the B-Bus and the bytes applied to the B operand of the ALU.

BBC=0

The contents of the general register specified by RB Select are placed on the B-Bus. Both bytes of the B-Bus transfer to the ALU.

BBC=	IOM=	
0	X	B(0-15) ⇐ GRn(0-15); ALU-B(0-15) ⇐ B(0-15)

BBC=1

The contents of the general register specified by RB Select are placed on the B-Bus. The left byte of the B-Bus transfers to the ALU, right justified.

BBC=	IOM=	
1	X	B(0-15) ⇐ GRn(0-15); ALU-B(8-15) ⇐ B(0-7); ALU-B(0-7) ⇐ 0

BBC=2

The contents of the general register specified by RB Select are placed on the B-Bus. The right byte of the B-Bus transfers to the ALU, right justified.

BBC=	IOM=	
2	X	B(0-15) ⇐ GRn(0-15); ALU-B(8-15) ⇐ B(8-15); ALU-B(0-7) ⇐ 0

BBC=3

If IOM=0, transfers Emit Field to the B-Bus, right justified. If IOM=1, transfers a one to the position in the right byte of the B-Bus defined by RB Select. (See subsection 3-19, Bit Switch/Position, for greater details.) In either case, the right byte of the B-Bus transfers to the ALU, right justified.

BBC=	IOM=	
3	0	B(8-15) ⇐ MCR(24-31); ALU-B(8-15) ⇐ B(8-15); ALU-B(0-7) ⇐ 0
	1	B(8-15) ⇐ 1(RB Select); ALU-B(8-15) ⇐ B(8-15); ALU-B(0-7) ⇐ 0

The B-Bus Control can be overridden by MCR=7, Read Argument to B-Bus from Translator (optional).

3-10 C-BUS/REGISTER CONTROL

The C-Bus/Register Control (CBRC) field (bits 12-15 of the micro instruction) is concerned primarily with routing data from the C-Bus to the general registers or to input/output modules. This field is concerned also with shifting left or right one position the ALU output to the C-Bus; with applying the value on the C-Bus to the micro address register; with incrementing/decrementing the value on the B-Bus and returning it to a general register; and with using the last twelve bits of the micro instruction as an immediate address.

In the case of transfers from the C-Bus to the general registers or input/output modules defined by IO Mode, RA Select, and Device Select, the bytes transferred are controlled by A-Bus Control. In the case of transfers from the C-Bus to general registers selected by RB Select, the bytes transferred are controlled by B-Bus Control.

Transfers from the general registers or input/output modules to the A-Bus are not affected by the C-Bus/Register Control field with one exception: C-Bus/Register Control=x'B' inhibits transfers to the A-Bus, since the A-Bus is being used by the immediate address. For other values of the C-Bus/Register Control field, transfers from the general registers or input/output modules to the A-Bus are always enabled.

Moreover, transfers from the general registers to the B-Bus are not affected by the C-Bus/Register Control field. In other words, these transfers are always enabled, so far as this field is concerned.

CBRC=X'O'

No register is strobed. Data from C-Bus is not transferred into a general register, input/output module or any other register. However, transfers from a general register or an input/output module to the A-Bus and from a general register to the B-Bus can take place. (Transfers from a general register to the A-Bus can be inhibited by the C-Bus/Register Control field, but not enabled.)

CBRC=

0	C(0-15) ⇐ ALU(0-15)

CBRC=X'1'

C-Bus to MAR. Transfers the least significant 12 bits on the C-Bus to the micro address register on the clock following the micro opcode. The status register is not updated, but its contents are maintained. Then the micro instruction corresponding to the micro address transferred into the MAR is transferred into the micro control register on the second clock following the micro opcode and is available for execution during the third clock period. Thus, execution of this micro opcode occupies two clock periods (700 nanoseconds).

CBRC=

1	C(0-15) ⇐ ALU(0-15); MAR(0-11) ⇐ C(4-15)·PHIn; MAM(0-11) ⇐ MAR(0-11); MCR(0-31) ⇐ CM(0-31)(MAMO-11)·PHIn+1.

Status Register

The foregoing micro opcode is the first one encountered in which the status register was not updated. Normally, the status register is updated by the clock concluding each micro instruction period. Under those circumstances described below, however, the status register is not updated and the existing contents are maintained. In brief, the contents of the status register are unchanged when branching or jumping to a new micro address. In detail:

a. The C-Bus/Register Control field of the current micro instruction equals x'1' or x'9', involving C-Bus to MAR, or

b. The Branch Control field of the current micro instruction specified a Test/Branch micro opcode or Push, or

c. A fast interrupt micro instruction in control memory location x'006' or x'007' is being executed.

CBRC=X'2'

RA Select, Device Select, IO Mode and A-Bus Control are used as routing address.

If A-Bus Control=0, transfers both bytes of C-Bus to general register or input/output module selected by IO Mode, RA Select and Device Select. If A-Bus Control=1, transfers the right byte of ALU output to the left byte of the C-Bus, and transfers the left byte of the C-Bus to registers or input/output modules. For other values of A-Bus Control, transfers both bytes of ALU output to the C-Bus. For A-Bus Control=2, transfers right byte of C-Bus to registors or input/output modules. If A-Bus Control=3, transfers both bytes of C-Bus to general register.

CBRC=	ABC=	IOM=	
2	0	0	GRn(0-15) ⇐ C(0-15); C(0-15) ⇐ ALU(0-15)
2	0	1	IOn(0-15) ⇐ C(0-15); C(0-15) ⇐ ALU(0-15)
2	1	0	GRn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(8-15);C(8-15)<=0
2	1	1	IOn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(8-15);C(8-15)<=0
2	2	0	GRn(8-15) ⇐ C(8-15); C(0-15) ⇐ ALU(0-15)
2	2	1	IOn(8-15) ⇐ C(8-15); C(0-15) ⇐ ALU(0-15)
2	3	X	GRn(0-15) ⇐ C(0-15);C(0-15)<= ALU(0-15)

CBRC=X'3'

RB Select and B-Bus Control are used as routing address. If B-Bus Control=0 or 3, both bytes of the C-Bus are transferred to the general register addressed by RB Select and both bytes of ALU output are transferred to the C-Bus.

If B-Bus Control=1, the right byte of ALU output is transferred to the left byte of the C-Bus and the left byte of the C-Bus is transferred to a general register; for B-Bus Control=2, the right byte of the C-Bus is transferred to a general register.

CBRC=	BBC=	
3	0	GRn(0-15) ⇐ C(0-15); C(0-15) ⇐ ALU(0-15)
3	1	GRn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(8-15); C(8-15) <= 0
3	2	GRn(8-15) ⇐ C(8-15); C(0-15) ⇐ ALU(0-15)
3	3	GRn(0-15) ⇐ C(0-15); C(0-15) ⇐ ALU(0-15)

CBRC=X'4'

Shifts ALU output (to C-Bus) one bit to the left, and uses RA Select, Device Select, IO Mode, and A-Bus Control as routing address. If A-Bus Control = 1, shifts right byte of ALU output one bit to the left, fills least significant bit with zero, and transfers to left byte of C-Bus. Transfers left byte of C-Bus to left byte of general register or input/output module addressed by IO Mode, RA Select, and Device Select. For other values of A-Bus Control, shifts both bytes of ALU output one bit to the left, fills least significant bit with zero, and transfers both bytes to C-Bus. If A-Bus Control = 0, transfers both bytes of C-Bus to general register or input/output module. For A-Bus Control = 2, transfers right byte of C-Bus to general register or input/output module. If A-Bus Control =3, transfers both bytes of C-Bus to a general register.

CBRC=	ABC=	IOM=	
4	0	X	C(0-14)<= ALU(1-15); C15 <= 0
4	0	0	GRn(0-14) <= C(0-14); GRn15 <= C15(=0)
4	0	1	IOn(0-14) <= C(0-14); IOn15 <= C15(=0)
4	1	X	C(0-6) <= ALU(9-15), C7 <= 0 C(8-14) <= ALU(9-15), C15 <= 0
4	1	0	GRn(0-6) <= C(0-6); GR7 <= C7(=0)
4	1	1	IOn(0-6) <= C(0-6); IOn <= C7(=0)
4	2	X	C(0-14) <= ALU(1-15); C15 <= 0
4	2	0	GRn(8-14) <= C(8-14); GRn15 <= C15(=0)
4	2	1	IOn(8-14) <= C(8-14); IOn15 <= C15(=0)
4	3	X	C(0-14) <= ALU(1-15); C15 <= 0
4	3	X	GRn(0-15) <= C(0-15)

CBRC=X'5'

Shifts ALU output (to C-Bus) one bit to the left, and uses RB Select and B-Bus Control as routing address. If B-Bus Control = 1, shifts right byte of ALU output one bit to the left, fills least significant bit with zero, and transfers to left byte of C-Bus; for other values of B-Bus Control, shifts both bytes of ALU output one bit to the left, fills least significant bit with zero, and transfers both bytes to C-Bus. For B-Bus Control = 0 or 3, transfers both bytes of C-Bus to general register addressed by RB Select; for B-Bus Control = 1, transfers left byte of C-Bus to general register; for B-Bus Control = 2, transfers right byte of C-Bus to general register.

CBRC=	BBC=	
5	0+3	C(0-14) <= ALU(1-15); C15 <= 0; GRn(0-14) = C(0-14); GRn15 <= C15(=0)
5	1	C(0-6)<= ALU(9-15); C7 <= 0; C(8-14) = ALU(9-15); C15 <= 0; GRn(0-6) <= C(0-6); GRn7 <= C7(=0)
5	2	C(0-14) <= ALU(1-15); C15 <= 0; GRn(8-14) <= C(8-14); GRn15 <= C15(=0)

CBRC=X'6'

Shifts ALU output (to C-Bus) one bit to the right and uses RA Select, Device Select, IO Mode, and A-Bus Control as routing address. If A-Bus Control=0, shifts both bytes of ALU output one bit to the right, fills most significant bit with zero, and transfers both bytes to C-Bus; transfers both bytes of C-Bus to general register or input/output module addressed by IO Mode, RA Select, and Device Select. If A-Bus Control=1, shifts ALU output one bit to the right and transfers right byte of output to left byte of C-Bus; transfers left byte of C-Bus into register. In byte operation, since the inputs to the ALU A and B operands are limited to the right byte, output bit 7 will usually be zero, but may be a one if the operation performed results in a carry. For A-Bus Control=2, shifts right byte of ALU output one bit to the right, fills most significant bit with zero, and transfers to right byte of C-Bus; transfers right byte of C-Bus into register. In this case, the carry to bit 7, if any, from the ALU operation, is lost. If A-Bus Control=3, shifts both bytes of ALU output one bit to the right, fills most significant bit with zero, transfers both bytes to the C-Bus, and transfers both bytes of the C-Bus to a general register.

CBRC=	ABC=	IOM=	BBC=	
6	0	0	X	GRn(0-15) ⇐ C(0-15); C(1-15) ⇐ ALU(0-14); CO ⇐ 0
6	0	1	X	IOn(0-15) ⇐ C(0-15); C(1-15) ⇐ ALU(0-14); CO ⇐ 0
6	1	0	0	GRn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(7-14); C(8-15) ⇐ ALU(7-14)
6	1	0	1	GRn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(7-14); C8 ⇐ 0; C(9-15) ⇐ ALU(8-14)
6	1	1	0	IOn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(7-14); C(8-15) ⇐ ALU(7-14)
6	1	1	1	IOn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(7-14); C8 ⇐ 0; C(9-15) ⇐ ALU(8-14)
6	2	0	X	GRn(8-15) ⇐ C(8-15); C(1-7) ⇐ ALU (0-6); C8 ⇐ 0; C0 ⇐ 0; C(9-15) ⇐ ALU(8-14)
6	2	1	X	IOn(0-15) ⇐ C(8-15); C(1-7) ⇐ ALU(0-6); C8 ⇐ 0; C0 ⇐ 0; C(9-15) ⇐ ALU(8-14)
6	3	X	X	GRn(0-15) ⇐ C(0-15); C(1-15) ⇐ ALU(0-14); C0 ⇐ 0

CBRC=X'7'

Shifts ALU output (to C-Bus) one bit to the right, and uses RB Select and B-Bus Control as routing address.

If B-Bus Control=0, shifts both bytes of ALU output one bit to the right, fills most significant bit with zero, and transfers both bytes to C-Bus; transfers both bytes of C-Bus to general register addressed by RB Select.

If B-Bus Control=1, shifts ALU output one bit to the right and transfers right byte of output to left byte of C-Bus; transfers left byte of C-Bus into general register addressed by RB Select. In byte operation, since the inputs to the ALU A and B operands are limited to the right byte, output bit 7 will usually be zero, but may be a one if the operation performed results in a carry.

If B-Bus Control=2, shifts right byte of ALU output one bit to the right, fills most significant bit with zero, and transfers right byte to C-Bus; transfers right byte of C-Bus into general register addressed by RB Select.

If B-Bus Control=3, shifts each byte one bit to the right, fills in most significant bit of each byte with zero, and transfers both bytes to the C-Bus; transfers both bytes from C-Bus to general register addressed by RB Select.

CBRC=	BBC=	ABC=	
7	0	X	GRn(0-15) ⇐ C(0-15); C0 ⇐ 0; C(1-15) ⇐ ALU(0-14)
7	1	0+3	GRn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(7-14); C(8-15) ⇐ ALU(7-14)
7	1	1+2	GRn(0-7) ⇐ C(0-7); C(0-7) ⇐ ALU(7-14); C8 ⇐ 0; C(9-15) ⇐ ALU(8-14)
7	2	X	GRn(8-15) ⇐ C(8-15); C0 ⇐ 0; C(1-7) ⇐ ALU(0-6); C8 ⇐ 0; C(9-15) ⇐ ALU(8-14)
7	3	X	GRn(0-15) ⇐ C(0-15); C0 ⇐ 0; C(1-7) ⇐ ALU(0-6); C8 ⇐ 0; C(9-15) ⇐ ALU(8-14)

CBRC = X'8'

Increments B-Bus and places result in RB Selected register. Adds one to the bit value on the B-Bus and places the result in the general register addressed by RB Select. Transfers both bytes of ALU output direct to the C-Bus.

GRn(0-15)⇐B(0-15) plus 1;
C(0-15)⇐ALU(0-15)

CBRC = X'9'

Increments B-Bus and places result in RB Selected general register. C-Bus to MAR. See CBRC= X'1' and X'8' for detailed description. This micro opcode occupies two clock periods.

CBRC=X'A'

Increments B-Bus and places result in general register addressed by RB Select. Uses IO Mode, RA Select, Device Select, and A-Bus Control as routing address.

CBRC=	GRn(0-15)⇐B(0-15)+1
A	(Remaining operations are identical to CBRC=X'2'.)

CBRC=X'B'

Immediate Address. The Immediate Address placed on the A-bus can originate from one of four sources:

(1) If IO Mode is false, the Immediate Address field of the micro instruction.

(2) If IO Mode is true, the Device Address for the highest priority input/output device controller that has a multiplexed I/O interrupt pending.

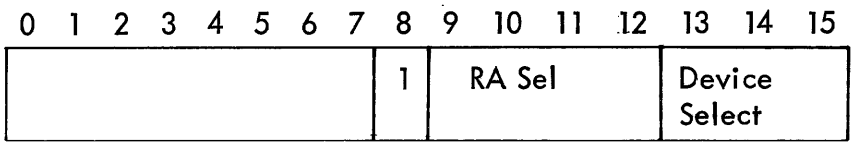
(3) If IO Mode is false, Branch Control is X'1', and MCR Control is X'3', Vector 1 micro address from the (optional) Translator.

(4) If IO Mode is false, Branch Control is X'1', and MCR Control is X'5', Vector 2 micro address from the (optional) Translator.

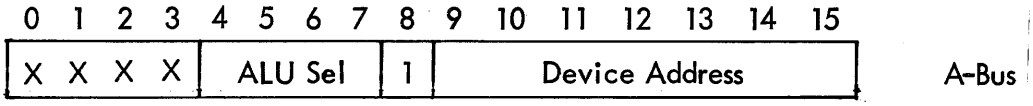
The remaining operations are common. The operations are described in detail below.

If IOM = 0, transfers bits 20-31 of the micro instruction, called the immediate address, from the micro control register to the A-Bus.

If IOM = 1, the input/output module with a multiplexed I/O interrupt pending that has the highest priority location places its Device Address (consisting of the RA Select and Device Select values) on the A-Bus, as follows:



The input/output module provides a signal forcing bit 8 true, in effect adding 128 to the Device Address. At the same time, the contents of the ALU Select field, bits 20-23 of the micro instruction are appended to the above to generate a 12-bit micro address, as follows:



If IOM = 0, Branch Control = X'1', and MCR = X'3' or X'5', the Translator places the Vector 1 or Vector 2 micro addresses on the A-Bus.

In each case, a 12-bit micro address is now present on the A-bus. This micro address via the memory address multiplexer addresses a location in control memory during the clock period of the micro opcode. At the same time, the 12 bits are applied to the A operand inputs of the ALU, incremented by one, transferred direct to the C-Bus, and applied to the micro address register at the clock following the micro opcode. Thus, the incremented micro address serves as the control memory address during the following clock cycle.

In the execution of this micro opcode, although the ALU Select field constitutes part of the Immediate Address or expanded Device Address, the ALU is forced to execute the function A plus 1 by the hardware.

The A-bus sources are summarized as follows:

If IOM = 0 then A(4-15) <= MCR(20-31)
If IOM = 1, then A(9-15) <= DA;
A8 <= 1;
A(4-7) <= MCR(20-23).

If IOM = 0, BRC = X'1', and MCR = X'3', then A(4-15) <= V1(0-11)
If IOM = 0, BRC = X'1', and MCR = X'5', then A(4-15) <= V2(0-11).

The remaining operations are common to all sources:

MAM(0-11) <= A(4-15);
ALU(0-15) <= A(4-15) plus 1;
C(0-15) <= ALU(0-15);
MAR(0-11) <= C(4-15). PHIn.

Both the unique and common operations are executed in one clock cycle. See also subsection 3-28, Immediate Addressing.

CBRC=X'C'

Increments B-Bus and places result in general register addressed by RB Select; shifts ALU output (to C-Bus) one bit to the left; uses RA Select, Device Select, IO Mode, and A-Bus Control as routing address.

GRn(0-15) <= B(0-15) plus 1
(Remaining operations same as CBRC = X'4'.)

CBRC = X'D'

Not assigned.

CBRC = X'E'

Increments B-Bus and places result in general register addressed by RB Select; shifts ALU output (to C-Bus) one bit to the right; uses RA Select, Device Select, IO Mode, and A-Bus Control as routing address.

GRn(0-15) <= B(0-15) plus 1
(Remaining operations same as CBRC = X'6'.)

CBRC = X'F'

Decrements B-Bus and places result in general register addressed by RB Select; uses RA Select, Device Select, IO Mode, and A-Bus Control as routing address.

GRn(0-15) <= B(0-15) minus 1
(Remaining operations same as CBRC = X'2'.)

3-11 BRANCH CONTROL

The Branch Control field (BC) is concerned primarily with Test/Branch micro opcodes. In general, a Test/Branch micro opcode tests a condition, external or internatl, and branches to an immediate address or a relative address as the next control memory location if the condition is found. Otherwise, the next micro address in sequence is taken as the control memory location.

See also Section 3-26.

In addition, the Branch Control field is concerned with arithmetic operations, reading or writing scratch pad/main memory, push-stack operations, and status bits.

As pointed out in subsection 3-10, CBRC = 1, certain Branch Control micro opcodes cause the status bits to be loaded into the status register. In the case of other Branch Control micro opcodes, the current status bits are not loaded, but those in the status register are retained. The effect of each Branch Control micro opcode on the status register is included in the following statements of function.

BC = X'0'

Normal Mode Carry False. In arithmetic mode (ALU Mode = 0), the ALU performs the operations listed under Cn = 0, ALU Select. Status bits are loaded into the status register at the next clock, providing C-Bus/Register Control is not X'1' or X'9'.

C <= ALU;
Cn = 0;
SR <= S. PHIn.

BC = X'1'

Normal Mode Carry True. In arithmetic mode (ALU Mode = 0), the ALU performs the operations listed under Cn = 1, ALU Select. Status bits are loaded into the status register at the next clock, providing C-Bus/Register Control is not X'1' or X'9'.

C <= ALU;
Cn = 1;
SR <= S. PHIn;

BC = X'2'

Read Scratch Pad or Main Memory. The contents of the SP/MM location addressed by B-Bus are read out to the C-Bus. Status bits are loaded into the status register at the next clock,

providing C-Bus/Register Control is not X'1' or X'9'. In the case of main memory the bytes read out are specified by A-Bus Control (ABC).

C(0-15)<= SP(0-15)(B)
SR<= S·· PHIn

ABC	
0	C(0-15)<= MM(0-15)(B)
1	C(0-7) <= MM(0-7)(B)
2	C(8-15)<= MM(8-15)(B)
3	C(0-15)<= MM(0-15)(B)

This micro opcode requires one clock period for scratch pad and a double-length clock period (700 nanoseconds) for main memory.

BC = X'3'

Write Scratch Pad or Main Memory. The bits on the A-Bus are written into the Scratch Pad/ Main Memory location addressed by the B-Bus. Because the C-Bus is not used, the values of the status bits derived from the C-Bus (though loaded into the status/register) are indeterminate. In the case of main memory the bytes written are specified by A-Bus Control(ABC).

SP(0-15)(B)<= A(0-15);
SR<= S · PHIn

ABC	
0	MM(0-15)(B)<= A(0-15)
1	MM(0-7)(B) <= A(0-7)
2	MM(8-15)(B)<= A(8-15)
3	MM(0-15)(B)<= A(0-15)

This micro opcode requires one clock period for scratch pad and a double-length clock period (700 nanoseconds) for main memory.

BC = X'4'

Test/Branch on External Condition Set. Addresses external line (or lines) on I/O module defined by RA Select and Device Select, if optional input/output modules with this function are installed.

If the test is successful, the least significant 12 bits of the C-Bus are loaded into the MAR on the clock following the micro opcode. The status register is not modified. The micro instruction corresponding to the micro address transferred into the MAR is transferred into the micro control register on the second clock following the micro opcode and is available for execution during the third clock period. Execution time is two clock cycles.

If the test is unsuccessful, the micro address register increments to the next micro address in sequence at the first clock. Under this circumstance, execution time is one clock cycle.

If EXT = 1, then MAR(0-11)⇐ C(4-15) · PHIn;
MCR(0-31)⇐ 0 · PHIn;
MAM(0-11)⇐ MAR(0-11);
MCR(0-31)⇐ CM(0-31)(MAM0-11) · PHIn + 1

If EXT = 0, then MAR plus 1 at PHIn

Because the Device Select field is employed in addressing the external lines (requiring that C-Bus/Register Control be other than X'B'), the immediate address field can not be used to supply the new address.

BC = X'5'

Test/Branch on External Condition Reset. Same as BC = X'4' except that the test state is reversed.

BC = X'6'

Push. Stores 12-bit micro address from micro address register and six status bits from status register in 18-bit push stack. Transfers contents of C-Bus to MAR to serve as the next micro address. Status register is not modified. Other operations depend upon the C-Bus/Register Control field.

If C-Bus/Register Control is X'B', the immediate address obtained from the last 12 bits of the mciro control register or the device address supplied by an input/output module requesting service provides the micro address that addresses control memory. Thus, Push with Immediate Address occupies one clock period.

If C-Bus/Register Control is not X'B', the micro control register resets at the first clock, so that no operations are executed during the following clock period. The micro address transferred into the MAR from the C-Bus addresses the micro instruction transferred into the micro control register on the second clock. This micro instruction is available for execution during the third clock period. Thus, Push without Immediate Address occupies two clock periods.

PS(0-11)⇐ MAR(0-11);
PS(12-17)⇐ SR(0-5)

SR unchanged

If CB/RC = X'B', then A(4-15) ⇐ MCR(20-31) + DA;
MAM(0-11) ⇐ A(4-15);
C(0-15) ⇐ A(4-15) plus 1;
MAR(0-11) ⇐ C(4-15) : PHIn

If CB/RC ≠ X'B', then MAR(0-11) ⇐ C(4-15) · PHIn;
MCR(0-31) ⇐ 0 · PHIn;
MAM(0-11) ⇐ MAR(0-11);
MCR(0-31) ⇐ CM(0-31)(MAM0-11) · PHIn + 1

Interrupts: The use of a Push function (and also a Pull function) in processing an interrupt is shown in Figure 3-1. It is assumed that an Interrupt Request goes true during the clock cycle in which micro instruction X, for example, is being executed. The next clock sets the Interrupt flip-flop, synchronizing the Interrupt Request. The Interrupt flip-flop causes a jump to a control memory location (in this example, X'008') in which the I/O Interrupt micro instruction is stored. This micro instruction is assumed to contain Push and Immediate Address commands, the effect of which is to place micro address R on the memory address multiplexer.

On the next clock, micro address R moves to the micro control register and is executed. At the same time, the I/O Interrupts are disabled.

Next, the interrupt routine, R, R+1, R+2, ... R+n, R+n+1, is executed. The R+n micro instruction contains an IO Control micro opcode commanding the I/O module to drop the Interrupt Request. The R+n+1 micro instruction contains a Pull command (see following paragraph). At this time the Enable IO Interrupts status bit from the push stack restores the original IO Interrupt Enable status.

BC = X'7'

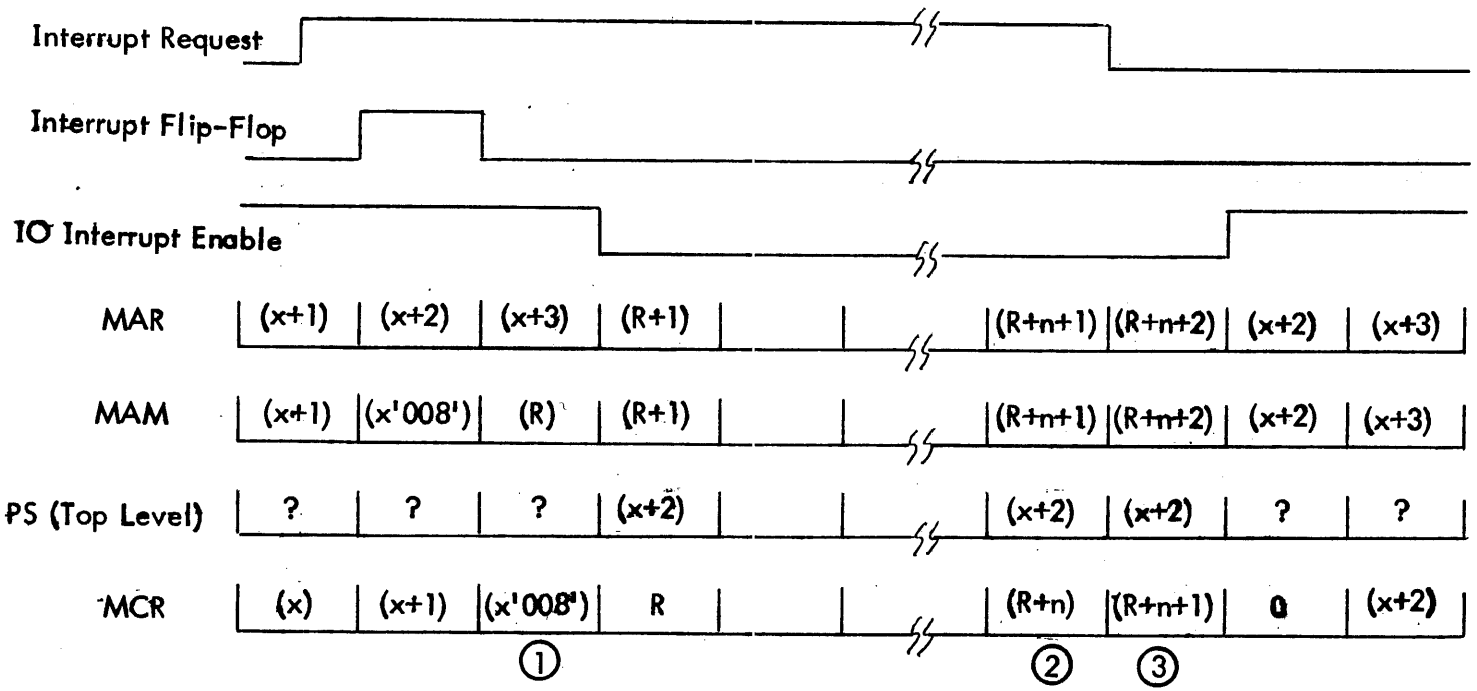
Pull. Transfers micro address from the last-used push-stack location via the C-Bus to the micro address register on first clock. Transfers status bits from last-used push-stack location into status register at same time. The micro instruction corresponding to the micro address placed into the MAR is transferred into the micro control register on the second clock and is available for execution during the third clock period. The micro address register increments to the next micro address in sequence on the second clock. This micro opcode occupies two clock periods.

C(4-15) ⇐ PS(0-11);
SR(0-5) ⇐ PS(12-17) · PHIn;
MAR(0-11) ⇐ C(4-15) · PHIn;
MCR(0-31) ⇐ 0 · PHIn;
MAM(0-11) ⇐ MAR(0-11);
MCR(0-31) ⇐ CM(0-31)(MAM0-11) · PHIn+1;
MAR plus 1 at PHIn+1

BC = X'8'

Test/Branch Overflow = 1. Examines the state of the Overflow bit in the status register. Basically, if this test is successful, the value on the C-Bus is placed in the MAR to serve as the micro address. If unsuccessful, the MAR increments to the next micro address in sequence. The value on the C-Bus depends upon the C-Bus/Register Control function.

C-Bus/Register Control is not X'B': If the test is successful, the least significant 12 bits of the C-Bus are loaded into the MAR on the clock following the micro opcode. The status register is not modified. The micro instruction corresponding to the micro address placed into the MAR is transferred into the micro control register on the second clock following the micro opcode and is available for execution during the third clock period. This procedure occupies two clock periods.



① Push Immediate to micro address R (stores Enable IO Interrupt status bit in Push Stack.)

② Commands Interrupt Request to be dropped.

③ Pull (restores original Enable IO Interrupt Status.

Figure 3-1. Interrupt Timing Diagram

If the test is unsuccessful, the micro address register increments to the next micro address in sequence at the first clock. Under this circumstance, the micro opcode occupies one clock period.

If OV = 1 and CB/RC ≠ X'B', then

MAR(0-11 ⇐ C(4-15) · PHIn;
MCR(0-31) ⇐ 0 · PHIn;
MAM(0-11) ⇐ MAR(0-11);
MCR(0-31) ⇐ CM(0-31)(MAM0-11) · PHIn+1

otherwise MAR plus 1 at PHIn

C-Bus/Register Control is X'B': If the test is successful, the immediate address obtained from the last 12 bits of the micro control register provides the micro address that addresses control memory during the clock period and the incremented micro address for the second clock period. Status register is not modified. If the test is not successful, the micro address register increments to the next micro address in sequence. This procedure occupies one clock period.

If OV = 1 and CB/RC = X'B', then

A(4-15) ⇐ MCR(20-31);
MAM(0-11) ⇐ A(4-15);
C(0-15) ⇐ A(4-15) plus 1;
MAR(0-11) ⇐ C(4-15) · PHIn.

otherwise MAR plus 1 at PHIn

BC = X'9' through X'F'

Test/Branch functions X'9' through X'F' follow the pattern set forth under X'8' above. Only the condition tested is different.

BC = X'9'

Test/Branch C-Bus = 0. Examines the state of the C-Bus = 0 in the status register.

BC = X'A'

Test/Branch Carry = 1. Examines the state of the Carry bit in the status register.

BC = X'B'

Test/Branch CMSB = 1. Examines the state of the most significant bit on the C-Bus, which is the sign bit in arithmetic operations. If the sign bit is 1 (negative), the test is successful.

BC = X'C'

Test/Branch Overflow ≠ 1. Same as BC = X'8' except that the test is reversed.

BC = X'D'

Test/Branch C-Bus ≠ 0. Same as BC = X'9' except that the test is reversed.

BC = X'E'

Test/Branch Carry ≠ 1. Same as BC = X'A' except that the test is reversed.

BC = X'F'

Test/Branch CMSB ≠ 1. Same as BC = X'B' except that the test is reversed.

3-12 MULTIPLE PURPOSE FIELDS

Micro instruction bits 20-31 are divided into three 4-bit fields which are used for multiple purposes. The particular use of each field is determined by the micro opcodes assigned to IO Mode, B-Bus Control, C-Bus/Register Control, and Branch Control, as tabulated in Table 3-2. In brief, there are four distinctions:

a. IO Mode distinguishes between the use of the second multiple-purpose field, bits 24-27, for input/output control, or other purposes.

b. B-Bus Control distinguishes between the use of the second and third multiple-purpose fields, bits 24-31, for Emit Field, or other purposes.

c. C-Bus/Register Control distinguishes between the use of all three fields, bits 20-31, for Immediate Address, or other purposes.

d. Branch Control field distinguishes the source of the Immediate Address.

The specific function, taken alone, of each micro opcode for each purpose of the multiple-purpose fields is defined below.

3-13 ALU SELECT

The ALU Select field is used to select the function the ALU is to perform, as described in Table 3-3. In addition, bits 20-23 may be used for other functions, as defined in subsection 3-14, MCR Control.

The only difference between the two sets of arithmetic functions listed in Table 3-3 is that an incoming carry adds a one to the result. For example, ALU Select = X'6' generates A minus B minus 1 (one's complement subtract) with carry-in false and A minus B (two's complement subtract) with carry-in true.

Arithmetic operations may produce changes in the status bits Carry, C-Bus = 0, C-Bus MSB, and Overflow. The status register is updated at the next clock following each arithmetic operation (as well as at other times) to reflect these changes. The status bits are defined briefly in Section 2, but Overflow is explained more fully below.

Table 3-2. Multiple Purpose Micro Fields

0	1——3	4	5——7	8 9	10 11	12 —— 15	16 —— 19	20 ———— 23	24 —— 27	28 —— 31
IOM	RA SELECT	ALUM	RB SELECT	A-BUS CONTROL	B-BUS CONTROL	C-BUS/REG CONTROL	BRANCH CONTROL			
0					N3	N X 'B'		ALU SELECT	MCR CONTROL	DEVICE SELECT
0					N3	N X 'B'		OTHER FUNCTIONS	MCR = A	DEVICE SELECT
0					3	N X 'B'		ALU SELECT	EMIT FIELD	
0						X 'B'	N X '1'	IMMEDIATE ADDRESS FIELD		
0						X'B'	X '1'		MCR = x'3' + x'5'	
1					N3	N X 'B'		ALU SELECT	IO CONTROL	DEVICE SELECT
1		BIT POSITION			3					

Key: NX'n' = Not Hexadecimal n. ($\overline{n}$).

\+ = OR Function

Table 3-3. Functions of ALU Select

ALU Select Bits 20-23 (Hexidecimal)	ALU Mode Reset		ALU Mode Set
	Arithmetic Function Carry In False (Branch Control = 0)	Arithmetic Function Carry In True (Branch Control = 1)	Logic Function
0	C ⇐ A	C ⇐ A plus 1	C ⇐ $\overline{A}$
1	C ⇐ A + B	C ⇐ (A + B) plus 1	C ⇐ $\overline{A + B}$
2	C ⇐ A + $\overline{B}$	C ⇐ (A + $\overline{B}$) plus 1	C ⇐ $\overline{A}$ · B
3	C ⇐ Minus 1	C ⇐ 0	C ⇐ 0
4	C ⇐ A plus A · $\overline{B}$	C ⇐ A plus A · $\overline{B}$) plus 1	C ⇐ $\overline{A \cdot B}$
5	C ⇐ (A + B) plus A · $\overline{B}$	C ⇐ (A + B) plus A · B plus 1	C ⇐ $\overline{B}$
6	C ⇐ A minus B minus 1	C ⇐ A minus B	C ⇐ A ∀ B
7	C ⇐ A · $\overline{B}$ minus 1	C ⇐ A · $\overline{B}$	C ⇐ A · $\overline{B}$
8	C ⇐ A plus A · B	C ⇐ A plus A · B plus 1	C ⇐ $\overline{A}$ + B
9	C ⇐ A plus B	C ⇐ A plus B plus 1	C ⇐ $\overline{A \forall B}$
A	C ⇐ (A + $\overline{B}$) plus A · B	C ⇐ (A + $\overline{B}$) plus A · B plus 1	C ⇐ B
B	C ⇐ A · B minus 1	C ⇐ A · B	C ⇐ A · B
C	C ⇐ A plus A	C ⇐ A plus A plus 1	C ⇐ 1's
D	C ⇐ (A + B) plus A	C ⇐ (A + B) plus A plus 1	C ⇐ A + $\overline{B}$
E	C ⇐ (A + $\overline{B}$) plus A	C ⇐ (A + $\overline{B}$) plus A plus 1	C ⇐ A + B
F	C ⇐ A minus 1	C ⇐ A	C ⇐ A

Key: $\overline{n}$ = 1's complement of n; + = OR function;
. = AND function; plus = add; minus = subtract;
∀ = Exclusive OR function; ⇐ = Replaced with.

The overflow status bit is set whenever the result of the arithmetic functions, A plus B (ALU Mode = 0, ALU Select = X'9'), and A minus B (ALU Mode = 0, ALU Select = X'6') are too large to be contained in a 16-bit word (or 8-bit byte). Overflow can be set or reset unconditionally.

The following conditions are required to set Overflow, assuming A and B are two's complement numbers:

1. If A plus B and
   A positive, B positive, and ALU output negative, or
   A negative, B negative, and ALU output positive
2. If A minus B and
   A positive, B negative, and ALU output negative, or
   A negative, B positive, and ALU output positive

The above algorithm is applied to the right byte of the A operand, B operand, and ALU output if A-Bus Control and B-Bus Control both specify a byte operation. Otherwise, the algorithm is applied to the entire 16-bit word. See also subsection 3-21, Word/Byte Operation.

Overflow, once set, remains set until turned off by a micro instruction command (See MCR Control field).

3-14 MCR CONTROL

If IO Mode = 0, B Bus Control is not 3 (Emit Field to B-Bus), and C-Bus/Register Control is not X'B' (immediate Address), bits 24-27 become the Micro Control Register (MCRC) Control field (see Table 3-2). This field is concerned with translator operation, enabling/disabling and reading I/O interrupts, write/read control memory, and enabling Other Functions.

MCRC = X'0'

No special function.

MCRC = X'1'

C-Bus to Translator (optional). If a translator function is installed, a 16-bit macro instruction on the C-Bus, normally obtained from scratch pad/main memory, is transferred into the translator.

T ⇐ C

MCRC = X'2'

Resets MAR. Resets micro address register to micro address 0. Resets micro control register to all zeros at PHI following this micro opcode, so that a NOP is executed during the following clock period. Inhibits incrementing of micro address register at this PHI, so that micro address register continues to hold micro address zero during the following clock period. The

micro instruction corresponding to micro address zero reaches the micro control register at the second PHI following the micro opcode and is available for execution during the third clock period. Execution of this micro opcode occupies two clock periods.

MAR(0-11) ⇐ 0
MAR unchanged at PHIn
MCR(0-31) ⇐ 0 · PHIn
MCR(0-31) ⇐ CM(0-31)(MA=0) · PHIn+1

MCRC = X'3'

Read Vector 1 to A-Bus from Translator (optional). If a translator function is installed, reads vector 1 (a 12-bit micro address) to A-Bus from translator.

A(4-15) ⇐ V1(0-11)

In addition, if IOM = 0, C Bus/Register Control = X'B', and Branch Control = X'1', the Immediate Address procedure becomes effective. The Vector 1 micro address, now on the A-Bus, via the memory address multiplexer addresses a location in control memory. At the same time, the 12 bits are applied to the A operand inputs of the ALU, incremented by one, transferred direct to the C-Bus, and applied to the micro address register at the next clock. This combination of micro opcodes achieves one-clock-cycle vector micro addressing.

MCRC = X'4'

Enable IO Interrupts. Enables IO Interrupt register, permitting a multiplexed input/output interrupt to be accepted and acted upon. Following this micro opcode, the IO Interrupt register remains enabled until disabled by MCR = X'6', or by the SCU hardware when an interrupt occurs, as discussed in Section 4. Also see Figure 3-1.

IR ⇐ IOI

MCRC= X'5'

Read Vector 2 to A-Bus from Translator (optional). If a translator function is installed, reads vector 2 (a 12-bit mciro address) to A-Bus from translator. Refer to MCR = X'3' for description of one-clock-cycle micro addressing.

A(4-15) ⇐ V2(0-11)

MCRC = X'6'

Disable IO Interrupts. Disables IO Interrupt register, inhibiting the acceptance of a multiplexed IO Interrupt and, consequently, preventing action on the interrupt. Following this micro opcode, the IO Interrupt register remains disabled until enabled by MCR = X'4'.

IR ⇐ IOI

MCRC = X'7'

Read Argument to B-Bus from Translator (optional). If a translator function is installed, read argument field of a macro instruction, such as the address field used in calculating direct or indirect memory locations, to B-Bus from translator.

B(0-15) ⇐ ARG(0-15)

MCRC = X'8'

Read MCP Data Entry switches to C Bus (optional).

C(0-15) ⇐ MCP(0-15)

MCRC = X'9'

Not assigned.

MCRC = X'A'

Other Functions. Inhibits output of ALU to C-Bus. Enables bits 20-23 to be used to code Other Functions, as follows:

Bits 20-23

X'8'	Reset Overflow Status Bit.	OV ⇐ 0
X'9'	Set Overflow Status Bit.	OV ⇐ 1
X'A'	Reset Control Mode Status Bit.	CON ⇐ 0
X'B'	Set Control Mode Status Bit.	CON ⇐ 1

MCRC = X'B'

Not assigned.

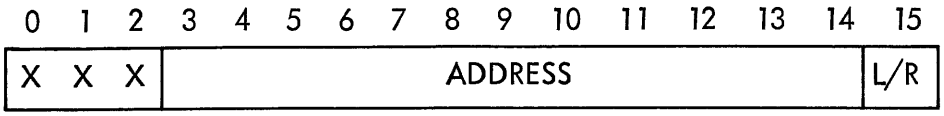
MCRC = X'C'

Write Micro Control Memory (optional). If a read/write random-access control memory is installed, transfers 16 bits from A-Bus into left half of control memory location addressed by bits 3-14 of B-Bus when B15 is false, or into right half when B15 is true. The micro instruction corresponding to the micro address for the micro opcode period reaches the micro control register at the second clock following the micro opcode and is available for execution during the third clock period. Execution of this micro opcode occupies two clock periods.

If B15 = 0, then CM-RAM(0-15)(B03-B14) ⇐ A(0-15)

If B15 = 1, then CM-RAM(16-31)(B03-B14) ⇐ A(0-15)

MCR(0-31 ⇐ 0 · PHIn
MCR(0-31) ⇐ CM(0-31)(MAM0-11) · PHIn+1
MAR plus 1 at PHIn+1



MCRC = X'D'

Read Micro Control Memory. Transfers left half of micro instruction addressed by B03-B14 of B-Bus from control memory to C-Bus when B15 is false, or transfers right half when B15 is true. This micro opcode occupies two clock periods.

If B15 = 0, then C(0-15) ⇐ CM(0-15)(B03-B14)

If B15 = 1, then C(0-15) ⇐ CM(16-31)(B03-B14)

MCR(0-31) ⇐ 0 · PHIn

MCR(0-31) ⇐ CM(0-31)(MAM0-11) · PHIn+1
MAR plus 1 at PHIn+1

MCRC = X'E'

Read IO Interrupts to A-Bus (optional). Reads the Device Address of highest priority IO controller with a multiplexed IO interrupt pending to the A-Bus.

A ⇐ DA

MCRC = X'F'

Not assigned.

3-15 IO CONTROL

When IO Mode is true (see Table 3-2), bits 24-27 become the IO Control field. This field is used to operate IO Modules. Refer to Section 4.

3-16 DEVICE SELECT

When B-Bus Control is not 3 and C-Bus/Register Control is not X'B' (see Table 3-2), bits 28-31 become the Device Select field. This field, in conjunction with RA Select, acts as a 7-bit IO device address.

3-17 EMIT FIELD

When IO Mode is false, B-Bus Control is 3, and C Bus/Register Control is not X'B' (see Table 3-2), the last eight bits of the micro instruction are transferred from the micro control register to bits 8-15 of the B-Bus.

B(8-15) ⇐ MCR(24-31)

B(0-7) ⇐ 0

3-18 IMMEDIATE ADDRESS

Wneh IO Mode is false and C-Bus/Register Control is X'B' (see Table 3-2), the last twelve bits of the micro instruction, called the Immediate Address, serve as the micro address for control memory, as described under C-Bus/Register Control = X'B'.

3-19 BIT SWITCH/POSITION

When IO Mode is true and B-Bus Control is 3, the Bit Switch is considered to be on, and RB Select becomes the Bit Position assignment field. This field assigns a one-state to any one of the bit positions 8 to 15 on the B-Bus, as listed in Table 3-4.

3-20 BUS USAGE

Some combinations of fields attempt to place data on a bus from more than one source, but the A, B, or C-Busses can be used by the various fields of any one micro instruction for only one purpose – that is, only one source should be assigned to any bus. If more than one source is assigned, the processor hardware applies some arbitrary priorities.

Table 3-5 lists all the standard sources and five optional sources from which data can be placed on the three buses. The five optional sources are the input/output device controller modules, instruction translator, scratch pad, main memory and maintenance control panel. For each source the micro programming prerequisites are tabulated. For some bus sources, the programming prerequisites prevent the simultaneous programming of other sources.

Table 3-4. B-Bus Bit Position Assignments

BIT POSITION FIELD			OCTAL	B-BUS CONTENTS
5	6	7		0 ——— 15
0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0	0	1	1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0	1	0	2	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0	1	1	3	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
1	0	0	4	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
1	0	1	5	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
1	1	0	6	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
1	1	1	7	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

Table 3-5. Micro Programming of Bus Sources

0	1 — 3	4	5 — 7	8 9	10 11	12 —— 15	16 —— 19	20 – 23	24 — 27	28 —— 31	
I O M	RA SELECT	M	RB SELECT	A-BUS CONT.	B-BUS CONT.	C-BUS/REG CONTROL	BRANCH CONTROL	ALU SELECT			FUNCTION
0	0-7			N3		NX'B'					A ⇐ GRn
1	0-7			N3		NX'B'					A ⇐ IOn
				3							A(4-15) ⇐ MAR(0-11)
						X'B'					A(4-15) ⇐ MCR(20-31)
0					N3	NX'B'			MCR=3+5		A(4-15) ⇐ V1+V2
0					N3	NX'B'			MCR=E		A ⇐ DA
1	0-7				3						B(8-15) ⇐ 1
	0-7				N3						B ⇐ GRn
0					3	NX'B'					B(8-15) ⇐ MCR(24-31)
0					N3	NX'B'			MCR=7		B ⇐ ARG
						NX'B'	0+1		MCR≠A		C ⇐ ALU
							2				C ⇐ SP/MM
							7				C(4-15) ⇐ PS(0-11)
						NX'B'			MCR=0		C ⇐ CM
0						NX'B'			MCR=8		C ⇐ MCP
						B	(via A, ALU)				C(4-15) ⇐ MCR(20-31)

Key: + = OR function
⇐ = Replaced with
X'N' = HEXADECIMAL NUMBER

3-21 WORD/BYTE OPERATION

Word or byte operation is specified by means of micro opcodes in the A-Bus Control and B-Bus Control fields, as previously defined and summarized for reference in Table 3-6.

There are two aspects of word or byte operation: first, the application of a word or a byte from the A-Bus to the A operand of the ALU and from the B-Bus to the B operand of the ALU; and second, the transfer of a word or a byte from the C-Bus to a general register or input/output bus or register. The second aspect is effective only for the C-Bus/Register micro opcodes listed in the table, while the first aspect depends only on A-Bus or B-Bus Control.

In arithmetic mode, the status bits Carry-out, C-Bus Most Significant Bit, and Overflow are modified to reflect byte or word operation.

3-22 CARRY

The ALU generates Carry (Byte) from ALU08, and Carry (Word) from ALU00. Carry (Byte) becomes Carry if both the A-Bus and the B-Bus are in byte mode; that is both buses are applying either the left byte or the right byte, both right justified, to the ALU operands, providing C-Bus/Register Control is not X'B', Immediate Addressing. Otherwise, Carry (Word) becomes Carry.

Specifically,

$$\begin{aligned}\text{Carry} = {} & \text{Carry (Byte)}(\text{A-Bus Control}=1+2)(\text{B-Bus Control}=1+2) \\ & (\text{C Bus/Register Control} \neq \text{B})(\text{Branch Control} \neq 7) \\ & + \text{Carry (Word)}(\text{Branch Control} \neq 7) \\ & + (\text{Push Stack})(\text{Branch Control} = 7)\end{aligned}$$

3-23 CMSB

C-Bus Most Significant Bit (sign bit) is derived from C08 if A-Bus Control is 2 and C-Bus/Register Control specifies Use RA Select, Device Select, IO Mode, and A-Bus Control as routing address, or if B-Bus Control is 2 and C-Bus/Register Control specifies Use RB Select and B-Bus Control as routing address, as shown by the underlining in Table 3-6.

Specifically,

$$\begin{aligned}\text{CMSB} = {} & \text{C08}(\text{BC} \neq 7)\,[(\text{ABC} = 2)\,(\text{CBRC} = 2+4+6+\text{A}+\text{C}+\text{E}+\text{F}) + (\text{BBC} = 2)\,(\text{CBRC} = 3+5+7)] \\ & + \text{C00}(\text{BC} \neq 7) \cdot \text{N}[(\text{ABC} = 2)\,(\text{CBRC} = 2+4+6+\text{A}+\text{C}+\text{E}+\text{F}) + (\text{BBC} = 2)\,(\text{CBRC} = 3+5+7)] \\ & + (\text{Push Stack})\,(\text{BC} = 7)\end{aligned}$$

Table 3-6. Word/Byte Operations

0 IOM	1 2 3 RA SELECT	4	5 6 7 RB SELECT	8 9 A-BUS	10 11 B-BUS	12 13 14 15 C-BUS/REG CONTROL	FUNCTION
				0			ALU-A(0-15) ⇐ A(0-15)
				1			ALU-A(8-15) ⇐ A(0-7)
				2			ALU-A(8-15) ⇐ A(8-15)
				3			ALU-A(4-15) ⇐ A(4-15)
					0		ALU-B(0-15) ⇐ B(0-15)
					1		ALU-B(8-15) ⇐ B(0-7)
					2		ALU-B(8-15) ⇐ B(8-15)
					3		ALU-B(8-15) ⇐ B(8-15)
1+0	0-7			0		2,4,6,A,C,E,F	IOn(0-15) + GRn(0-15) ⇐ C($\underline{0}$-15)
1+0	0-7			1		2,4,6,A,C,E,F	IOn(0-7) + GRn(0-7) ⇐ C($\underline{0}$-7)
1+0	0-7			2		2,4,6,A,C,E,F	IOn(8-15) + GRn(8-15) ⇐ C($\underline{8}$-15)
1+0	0-7			3		2,4,6,A,C,E,F	IOn(0-15) + GRn(0-15) ⇐ C($\underline{0}$-15)
			0-7		0	3,5,7	GRn(0-15) ⇐ C($\underline{0}$-15)
			0-7		1	3,5,7	GRn(0-7) ⇐ C($\underline{0}$-7)
			0-7		2	3,5,7	GRn(8-15) ⇐ C($\underline{8}$-15)
			0-7		3	3,5,7	GRn(0-15) ⇐ C($\underline{0}$-15)

Key: + = OR function,
⇐ = Replaced with
- = CMSB

3-24 OVERFLOW

Overflow is generated on a byte basis when both the A operand and the B operand of the ALU are on a byte basis. On the other hand, Overflow is generated on a word basis if either the A operand or the B operand is on a word basis. For further information on Overflow, refer to Subsection 3-13, ALU Select.

3-25 SET/RESET/TEST A BIT

The Bit Switch and the Position field can be used to set, reset, or test one bit of a data function. These functions are accomplished by comparing the data on the A-Bus with a known bit on the B-Bus. These comparisons are performed by the ALU in an appropriate select function. The known bit on the B-Bus is supplied by the Bit Position field. The procedure is as follows.

1. Insert data function on the A-Bus. Since the Bit Position field assigns bits only to positions B08-B15, the portion of the A-Bus function that is to be compared must be routed to positions A08-A15 at the A operand of the ALU. This routing is accomplished by the use of A-Bus Control, as shown in Table 3-6.
2. Insert desired bit-position mask on B-Bus, using the Bit Position Field.
3. With ALU Mode true (logic operation), use ALU Select:
   To set a bit: ALU Select = X'E' (A+B)

   For example:
   A Bus = 0100 0001
   B Bus = 0000 1000
   C = A+B = 0100 1001

   To reset a bit: ALU Select = X'7' (A· NB)

   A Bus = 0100 1001
   B Bus = 0000 1000
   C = A· NB = 0100 0001

   To test a bit: ALU Select = X'B' (A· B)

   A Bus = 0000 0001
   B Bus = 0000 1000
   C = A· B = 0000 0000

   In this case, Test/Branch if C-Bus = 0 is used as the next micro instruction. Then, if C-Bus = 0, the bit on the A-Bus that is tested is 0. On the other hand, if C-Bus is not equal to 0, then the bit tested is 1.

3-26 ADDRESSING AND BRANCHING

There are three forms of addressing: sequential (increment micro address register), immediate addressing, and relative addressing. Two types of branching are available: unconditional and conditional.

3-27 SEQUENTIAL ADDRESSING

The micro address register increments to the next micro address, in the absence of the following:

a. C-Bus/Register Control = X'1' or X'9' (C-Bus Data to MAR) or X'B' (Immediate or Device Addressing), and

b. Branch Control = X'4' or X'5' or X'8' through X'F' (Test/Branch functions), or X'6' (Push), or X'7' (Pull), and

c. Certain function switches pushed on the maintenance control panel.

3-28 IMMEDIATE ADDRESSING

If IOM = 0 and C-Bus/Register Control = X'B', the immediate address field from the last 12 bits of the micro instruction supplies the next micro address. If IOM = 1 and C-Bus/Register Control = X'B', the Device Address plus 128 of the highest priority device controller with a multiplexed IO interrupt pending supplies the next micro address. If IOM = 0, C-Bus/Register Control = X'B', Branch Control = X'1', and MCR Control = X'3' or X'5', the Translator (optional) places the Vector 1 or 2 micro address on the A-Bus.

Unconditional immediate addressing is implemented by the foregoing fields as well as, (in the case of immediate or device addressing) Branch Control = X'0', Normal Mode Carry False. Under these circumstances the actions listed in detail under C-Bus/Register Control = X'B' are carried out.

Conditional immediate or device addressing is implemented by C-Bus/Register Control = X'B' and Branch Control = X'8' through X'F', the status bit Test/Branch micro opcodes. If the result of the test specified by the Branch Control field is true, the immediate address or the device address is passed by the memory address multiplexer to address a location in control memory and, incremented by one, are entered, via the C-Bus, into the micro address register to be available as the micro address for the next clock period. If the test result is false, the memory address multiplexer passes a micro address from the micro address register during the current clock period and the micro address register increments for the next clock period.

If IOM = 0, then A(4-15) ⇐ MCR(20-31)
If IOM = 1, then A(4-15) ⇐ DA

If Test = 1, then MAM(0-11) ⇐ A(4-15);
C(0-15) ⇐ A(4-15) plus 1;
MAR(0-11) ⇐ C(4-15)· PHIn;
If Test = 0, then MAM(0-11) ⇐ MAR(0-11);

MAR plus 1 at PHIn

Immediate or device address micro opcodes require one clock period.

3-29 RELATIVE ADDRESSING

Relative addressing performs an arithmetic operation on the current micro address, taken from the micro address register, and the value on the B-Bus. The value on the B-Bus can be supplied by the Emit Field of the micro instruction, by a general register, or any other B-Bus source.

Relative addressing may be unconditional or conditional.

Unconditional relative addressing may be implemented by use of the fields shown on the first line of Table 3-7 (assuming that the B-Bus value is being obtained from the Emit Field). A-Bus Control = 3 puts the contents of the micro address register on the A-Bus. B-Bus Control = 3 puts the Emit Field on the B-Bus. (The 8-bit Emit Field can have any value up to 255.) ALU Mode = 0, and ALU Select = X'9' or X'6', adds or subtracts the A-Bus and the B-Bus and puts the result on the C-Bus. C-Bus/Register Control = X'1' enters the result into the micro address register at the first clock. The micro instruction corresponding to the micro address transferred into the MAR is transferred into the micro control register on the second clock and is available for execution during the third clock period. Unconditional relative addressing requires two clock periods.

A(4-15) ⇐ MAR(0-11);
ALU-A(4-15) ⇐ A(4-15)
B(8-15) ⇐ MCR(24-31);
ALU-B(8-15) ⇐ B(8-15)

If ALU Sel = X'9', then C(4-15) ⇐ ALU-A(4-15) plus ALU-B(8-15)
If ALU Sel = X'6', then C(4-15) ⇐ ALU-A(4-15) minus ALU-B(8-15)

MAR(0-11) ⇐ C(4-15)· PHIn;
MCR(0-31) ⇐ 0· PHIn;
MAM(0-11) ⇐ MAR(0-11);
MCR(0-31) ⇐ CM(0-31)(MAM0-11)· PHIn+1

Conditional relative addressing may be implemented as shown on the second line of Table 3-7. Cn is forced to 0 by the hardware; consequently, only the Cn = 0 column of ALU Select is usable. If the test is successful, the operations are the same as for unconditional relative addressing. If the test is unsuccessful, the micro address register increments at the first clock. Thus, if the test is successful, conditional relative addressing occupies two clock periods; if unsuccessful, one clock period.

3-30 SAMPLE MICRO PROGRAM

The micro program listed below assumes a three-part table in main memory, beginning at location 1000_{10}. The first part of the table contains 66_{10} data values. The second part contains the

Table 3-7. Relative Addressing, Conditional and Unconditional

	0 IOM	1 — 3 RA SELECT	4 ALUM	5 — 7 RB SELECT	8 — 9 A-BUS CONTROL	10 — 11 B-BUS CONTROL	12 — 15 C-BUS/REG CONTROL	16 — 19 BRANCH CONTROL	20 — 23 ALU SELECT	24 — 31
Uncon.	0				3	3	1	0	9+6	EMIT
Con.	0				3	3	0	8-F	9+6	EMIT

lower limits and the third part, the upper limits. The purpose of the micro program is to compare a data value with the lower and upper limits.

1.	1872	0A03	Load GR1 with location of data
2.	18B2	0AE8	Complete loading GR1 with location of data
3.	1433	0942	Set limit for loop test
4.	2902	2F00	Data from table
5.	1032	0942	Set pointer to lower limit table
6.	3902	2F00	Lower limit from table
7.	2300	1600	Is data greater than lower limit?
8.	000B	F0D0	Yes
9.	1032	0942	Set pointer to upper limit table
10.	390A	2F00	Upper limit from table and increment pointer
11.	3200	1600	Is data less than upper limit?
12.	000B	F0D4	Yes
13.	1032	1684	Reset pointer to data table
14.	1400	1600	Is loop complete?
15.	000B	B0CB	Sample complete (program would continue)

The effects of each field of the first micro instruction are detailed in Table 3-8. In summary, this micro instruction accomplishes the following:

a. Transfers the 8-bit Emit Field to the B-Bus (right justified). (The Emit Field to B-Bus micro opcode overrides the General Register to B-Bus micro opcode.)

b. Transfers contents of B-Bus through ALU to C-Bus (left justified).

c. Stores 8-bit C-Bus value (left justified) in left half of General Register 1.

The second micro instruction, 18B2 0AE8, is similar, except that it stores the contents of the Emit Field in the right half of General Register1. At the completion of this micro instruction, there is stored in General Register 1 the hexadecimal number 03E8, equivalent to decimal 1000, the number specified to be the initial location of the table in main memory.

The third micro instruction, 1433 0942, transfers $03E8_{16}$ to the A-Bus, 42_{16} from the Emit Field to the B-Bus, adds them together in the ALU, and puts the result, $042A_{16}$, in general register 4 via the C-Bus.

The fourth micro instruction, 2902 2F00, puts the contents of general register 1, $03E8_{16}$, on the B-Bus and uses this number to address main memory. The contents of this memory location are

Table 3-8. Analysis of First Micro Instruction

FIELD VALUE	FUNCTION
1	IOM = 0; RA Select = GR1
8	ALUM = 1; RB Select = GR0
7	A-BUS CONTROL = 1; A ⇐ GR1; ALU-A(8-15) ⇐ A(0-7) B-BUS CONTROL = 3 B(8-15) ⇐ MCR(24-31) ALU-B(8-15) ⇐ B(8-15)
2	C-BUS/REGISTER CONTROL = 2; C(0-7) ⇐ ALU(8-15), C(8-15) ⇐ 0 GR1(0-7) ⇐ C(0-7)· PHIn
0	BRANCH CONTROL = 0 C ⇐ ALU, Cn = 0 SR ⇐ S· PHIn
A	ALUSEL = X'A' C ⇐ B
0	EMIT
0	EMIT

placed on the C-Bus and transferred into general register 2. (When main memory is enabled, the output of the ALU to the C-Bus is inhibited.)

The fifth micro instruction, 1032 0942, transfers the 8-bit Emit Field, 42_{16}, from the micro control register via the B-Bus to the B operand input of the ALU. It applies $03E8_{16}$ from general register 1 via the A-Bus to the A operand. The ALU adds these two numbers and outputs the sum, $042A_{16}$, to the C-Bus. From the C-Bus this number is stored in general register 1.

The sixth micro instruction, 3902 2F00, puts the contents of general register 1, $042A_{16}$, on the B-Bus and uses this number to address main memory. This location contains the first lower limit. This lower limit is placed on the C-Bus and transferred into general register 3.

The seventh micro instruction, 2300 1600, puts the contents of general register 2 (the first data) on the A-Bus and the contents of general register 3 (the first lower limit) on the B-Bus. The ALU is set to subtract the lower limit from the data and place the result on the C-Bus. The C-Bus data is not strobed to any destination. On the same instruction the C-Bus Most Significant Bit is stored in the status register.

The eighth micro instruction, 000B F0D0, examines the most significant bit of the C-Bus from the previous micro instruction, now stored in the status register. If the bit is 0 (positive), indicating that the first data value is greater than the lower limit, the next micro address, $0D0_{16}$, is

taken from the immediate address field of the micro instruction via the A-Bus and the memory address multiplexer. At the same time, the contents of the immediate address field are applied to the ALU, incremented by one, and transferred via the C-Bus to the micro address register at the next clock. This incremented micro address then serves as the control memory address during the next clock period. On the other hand, if the C-Bus Most Significant Bit is 1 (negative), the micro address in the micro address register is incremented. This next micro instruction is the one that deals with a data value less than the lower limit. This micro opcode provides an example of conditional immediate addressing.

The remaining micro instructions of this micro program perform similar functions.

SECTION IV
SYSTEM INPUT/OUTPUT

4-1 INTRODUCTION

This section describes the design specifications that apply to any input/output module, and presents an example of the standard input/output module as it conforms to the design specifications. In addition, other available input/output modules are described.

In general, input/output modules enagage the three-bus architecture of the System Control Unit as shown in Figure 4-1. Under the direction of control signals derived from the micro instruction, an input/output module can:

a. Transfer data, status, and interrupt information from the external interface to the A-Bus

b. Transfer data and control information from the C-Bus to the external interface

c. Transfer data from the external interface to the B-Bus to provide a direct memory address, when using the fast interrupt capability

d. Sense the state of one or more external conditions.

4-2 INPUT/OUTPUT AND INTERRUPT SYSTEM SPECIFICATION

This section describes the criteria for designing input/output modules to interface with the three-bus structure of the Systems Control Unit. The design is specified in terms of the signals provided by or used by the SCU. In general, the SCU can be interfaced to all types of external equipment. Any kind of line protocol can be simulated by means of micro instructions, thus reducing the special control logic otherwise required on the input/output module.

4-3 INPUT/OUTPUT SYSTEM

The input/output system permits users to connect input/output functions directly into the bus structure of the data processor. Each one of a large number of I/O module connector locations is wired identically, except for priority determination, and each location can be used for effective data transfers in either a multiplexed fashion or in a faster Direct-To-Memory fashion. In the multiplexed arrangement, multiple I/O devices share a common I/O interrupt.

Special function-generation or algorithm-processing logic, which may not necessarily be related to I/O data transfers, can also be mechanized using the provisions of this structure.

A standardized interface, both hardware and protocol, between the user's unique logic elements and the signals on the I/O module connectors is mandatory and is described in detail in this section.

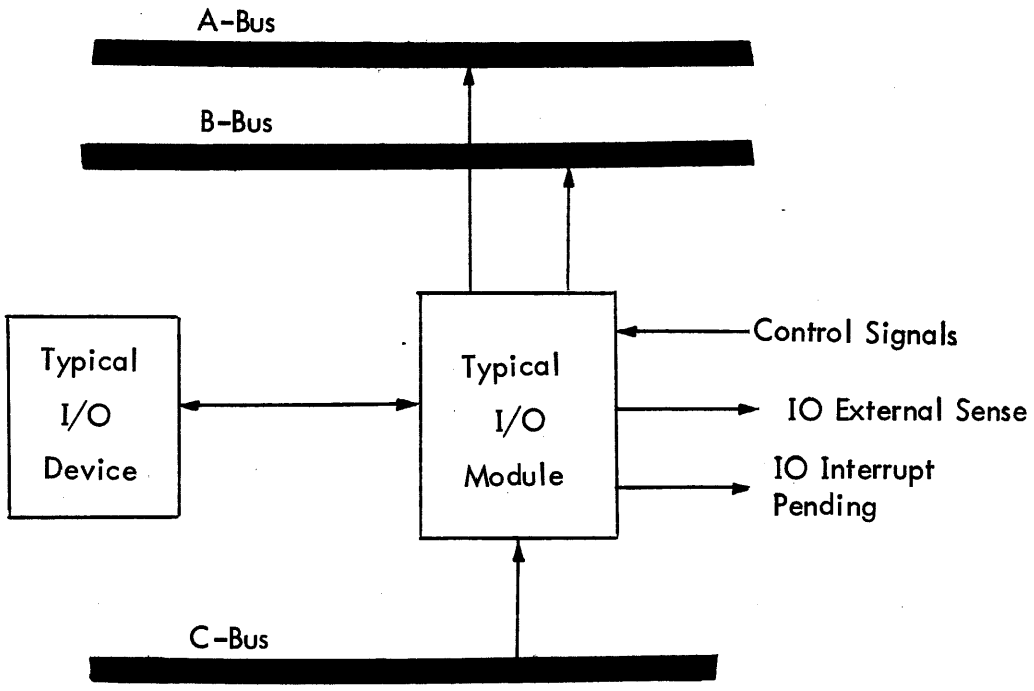


Figure 4-1. Block Diagram, Generalized Input/Output Module

An input/output module connects to the three-bus structure of the SCU and supplies interrupt identification to the SCU under the control of certain fields of the SCU micro instruction.

4-4 Data Paths

A typical I/O interface (controller) module interfaces to the A-Bus in order to move data, status, and interrupt identification information to the SCU. It interfaces to the C-Bus in order to move data and control information from the SCU, and to the B-Bus to provide a direct memory address when the interface (controller) uses the fast interrupt capability. The I/O interface module also interfaces to interrupt control logic in the SCU.

The basic SCU is able to move data on the A-, B-, and C-Buses in either 16-bit word or 8-bit byte increments. This capability allows easy interface to byte-oriented devices and also permits the 16-bit interfaces to be split between data and status information, or data and control information.

4-5 Interrupts

Two types of I/O interrupts are available in the SCU: multiplexed interrupts and fast interrupts.

The multiplexed I/O interrupt is available to every I/O interface (controller) module and may be requested at any time by a module. In response to this interrupt, the SCU accesses a fixed location in control memory which links it with an interrupt handler routine. The interrupt handler then requests the address of the highest priority device which has an interrupt pending.

This address is gated to the A-Bus by the highest priority device. The interrupt handler can then use the device address as an indirect address to the handler for the particular I/O controller.

The fast interrupt accesses a fixed location in control memory which contains a single instruction. This instruction typically moves data between the I/O controller and bulk data storage memory. The SCU has two fast interrupts. One can be used, for example, for moving data into the SCU, and the other for moving data out of the SCU.

4-6 Micro Instruction

The use of the micro instruction fields has been described in Section 3, but it is useful at this point to review the fields used to control input/output functions.

When IO Mode = 1, the RA Select and Device Select fields make up a 7-bit address field capable of selecting up to 128 I/O device controllers. RA Select and Device Select also select one of 128 external condition signals to be tested when one of the Branch Control functions, Test/Branch on External Condition Set or Reset, is employed.

A-Bus Control = 0+1+2 causes data from the I/O device controller addressed by IO Mode, RA Select, and Device Select to be placed on the A-Bus (unless C-Bus/Register Control is x'B' or MCR Control is x'3' or x'5').

A-Bus Control also provides byte-selection control for data written to an I/O device controller from the C-Bus, provided that C-Bus/Register Control specifies a function that includes the use of IO Mode, RA Select, and Device Select to define the destination.

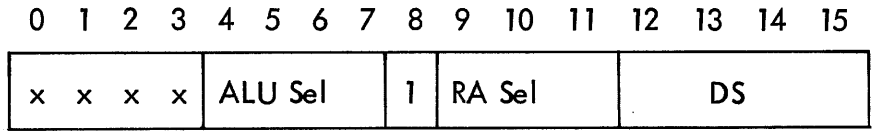
A-Bus Control	Definition
0	Full 16-bit word to destination
1	To left byte of destination
2	To right byte of destination
3	No transfer

C-Bus/Register Control specifies the destination of data on the C-Bus, as well as control over data-byte routing between the ALU and the C-Bus and right and left shifting of data transferred from the ALU to the C-Bus. The table below defines the destination for each decoded value of the C-Bus/Register Control field.

C-Bus/Register Control (Hexadecimal)	Destination
0,8	None
1,9,B	MAR

C-Bus/Register Control (Hexadecimal)	Destination (Continued)
2,4,6,A,C,E,F	IO Mode, RA Select, and Device Select define
3,5,7	RB Select defines
D	Not assigned

In addition, C-Bus/Register Control = x'B' has some other results significant to input/output operations. First, it overrides A-Bus Control. Second, if IO Mode = 0, the contents of the Immediate Address field are gated to the A-Bus. But, if IO Mode = 1, the Device Address (consisting of the RA Select and Device Select values) for the highest priority input/output device controller having a multiplexed I/O interrupt pending is placed on the right byte of the A-Bus, while the contents of the ALU Select field are placed in bits 4-7. This action yields the following format on the A-Bus.



MCR Control = x'E' is significant to input/output operations. This micro command causes the highest priority device controller with a multiplexed I/O interrupt pending to place its device address, as defined in the preceding paragraph, on the A-Bus.

When IO Mode = 1 and C-Bus/Register Control is not x'B', bits 24-27 become the Input/Output Control field. This field is not defined by the basic SCU and may be used by each unique device controller as a device command field. It should be recognized by a particular device controller only when IO Mode 1 and RA Select and Device Select contain the address of the controller. Thus, each device controller may interpret this field in any way the input/output module designer chooses.

4-7 INTERFACE SIGNALS AND REQUIREMENTS

The SCU operates synchronously on a single-phase clock with the exception of main memory accesses which require a double-length clock period. Thus, all operations within the unit happen within one or two clock times, simplifying interface timing requirements.

Important design characteristics include:

a. Unloading all C-Bus signals not actively in use.

b. Buffering all control signals supplied by the SCU (one TTL unit load of 2.0 ma per signal, maximum).

c. Using tri-state logic devices to drive all data bus connections to the SCU.

Table 4-1 contains the signal names, pin numbers, and functions of the interface signals. A brief description of the function and any unique design requirement of each signal follows:

A00 - A15: A-Bus. Used by IO interface modules to transfer data, status, and interrupt identification to the SCU. This bus must be driven by a tri-state integrated circuit, such as Xerox parts 200916, 200920, 200936, or other compatible devices. The A-Bus should not be loaded by any I/O module.

B00 - B15: B-Bus. Used by I/O interface modules which use the fast interrupts to provide a main memory address to the SCU. Bus connection rules are identical to those given above for the A-Bus.

C00 - C15: C-Bus. Used by the SCU to transfer data and control information to I/O interface modules. This bus is never driven by I/O interface modules. It must be unloaded from the module unless the module has been addressed by the SCU for data or control information transfer. Tri-state quad buffers, such as Xerox 200916, or equivalent, must be used as bus receivers for this purpose.

IOM: Input/Output Mode. Used by all I/O modules in qualifying data, control, and status transfers or from the SCU.

RSA01 - RSA03: A Register Select; EMIT28 - EMIT31: Device Select. Used to address the input/output module to be activated by the SCU. These seven signals are applied to each I/O module location. The design of an I/O module must permit the address of a module to be selectable as an installation parameter. See sub Section 4-9 for addressing information.

Note that IOM must be used with the Device Address compare to select an I/O module. In the case of certain types of module interfaces, it may be desirable to assign more than one 7-bit address to one module.

BYTESEL-L, BYTESEL-R: Byte Select, Left or Right. Used to control the transfer of the left byte or the right byte from the C-Bus to the output buffer register on an I/O module. When the micro instruction specifies word operation, both BYTESEL signals are true.

BCA08, BCA09: A-Bus Control. May be used with Device Address decode to address a single byte of data.

EMIT24 - EMIT27. When IO Mode = 1, and C-Bus/Register Control is not x'B', EMIT 24 - EMIT27 become the Input/Output Control field. These four signals are available to the designer to create special micro commands for the interface modules.

AXDASTROBE. Strobe for device address transfer to the A-Bus. Gates Device Address of the highest priority input/output interface having a multiplexed input/output interrupt pending to the A-Bus. See Section 4-13 for discussion of interrupt system.

TABLE 4-1

SCU Interface Signals

SIGNAL	PIN	FUNCTION
A00	H07	
A01	H13	
A02	H19	
A03	H25	
A04	H31	
A05	H37	
A06	H43	
A07	H50	
A08	J07	A-Bus Bits 0 Through 15
A09	J13	
A10	J19	
A11	J25	
A12	J31	
A13	J37	
A14	J43	
A15	J50	
B00	H09	
B01	H15	B-Bus Bits 0 Through 3
B02	H21	
B03	H27	

TABLE 4-1

SCU Interface Signals (Continued)

SIGNAL	PIN	FUNCTION
B04	H33	
B05	H39	
B06	H45	
B07	H46	
B08	J09	
B09	J15	B-Bus Bits 4 Through 15
B10	J21	
B11	J27	
B12	J33	
B13	J39	
B14	J45	
B15	J46	
C00	H08	
C01	H11	
C02	H17	
C03	H23	C-Bus Bits 0 Through 7
C04	H29	
C05	H35	
C06	H41	
C07	H47	

TABLE 4-1

SCU Interface Signals (Continued)

SIGNAL	PIN	FUNCTION
C08	J08	C-Bus Bits 8 Through 15
C09	J11	
C10	J17	
C11	J23	
C12	J29	
C13	J35	
C14	J41	
C15	J47	
IOM	J40	Input/Output Mode
NRSA01	J24	A Register Select Field Bit 01, inverted
NRSA02	J30	A Register Select Field Bit 02, inverted
NRSA03	J42	A Register Select Field Bit 03, inverted
NEMIT28	J10	Device Select Field Bit 28, inverted
NEMIT29	J12	Device Select Field Bit 29, inverted
NEMIT30	J14	Device Select Field Bit 30, inverted
NEMIT31	J18	Device Select Field Bit 31, inverted
EMIT24	J20	I/O Control Field Bit 24
EMIT25	J38	I/O Control Field Bit 25
EMIT26	J26	I/O Control Field Bit 26
EMIT27	J44	I/O Control Field Bit 27

TABLE 4-1

SCU Interface Signals (Continued)

SIGNAL	PIN	FUNCTION
NIMHERE	J28	Address Recognition Signal
NAXIOCONT	J22	Qualifier for A-Bus
BYTESEL-R(A)	J34	Byte Select (Right)
BYTESEL-L(A)	J06	" " (Left)
NINHB	J03	B-Bus override control signal
BCA08	H40	A Bus Control field (MSB)
BCA09	H44	A Bus Control field (LSB)
NIOINT	H10	Multiplexed I/O Interrupt Request
PRIn	H22	Priority String Input (Multiplexed I/O Interrupt)
PRIn+1	H24	Priority String Output (Multiplexed I/O Interrupt)
NINTLOCK	H14	Multiplexed I/O Interrupt Request Inhibit Signal
AXDASTROBE	H42	Device Address to A Bus Strobe Signal
NFIRQ1	H04	Fast Interrupt #1 Request
NFIRQ2	H06	" " 2 "
FI-1	H03	" " 1 Response
FI-2	H02	" " 2 "
NEXT	H38	External Condition Line
NRESET	H28	Power On and MCP Reset
PHI	J01	SCU System Clock
CLOCK	J05	Free Running 350ns Clock

TABLE 4-1

SCU Interface Signals (Continued)

SIGNAL	PIN	FUNCTION
RTCC	H12	60Hz Clock
ZP15 Volts	H01	+15 Volts
ZN15 Volts	H05	-15 Volts
+5v	H49, J49	Logic Power
GND	H00, 16, 32, 48 J00, 16, 32, 48	Logic Ground
FI1PRIn	H18	Fast Interrupt Priority Input
FI2PRIn	H34	Fast Interrupt Priority Input
FI1PRIn+1	H20	Fast Interrupt Priority Output
FI2PRIn+1	H36	Fast Interrupt Priority Output
Spare Pins	H26, 30, 51, J02, 04, 36, 51	Unused

NAXIOCONT. A-Bus transfer from IO Control. When low, inhibits input/output module from inserting any data onto the A-bus.

NRESET. May be used to initialize conditions on an I/O module. Driven to zero volts by the power supply at Turn On or by the Clear switch on the maintenance control panel. NRESET must be buffered on the I/O module.

CLOCK. This signal runs continuously, as indicated in Figure 4-2, with a period of 350 nanoseconds and a pulse width (low) between 50 and 80 nanoseconds.

PHI. Phi is the system clock. All clocking within the SCU occurs on the rising edge of PHI. Timing characteristics are shown in Figure 4-2. The period is 350 nanoseconds, except during a main memory access, when it is 700 nanoseconds. When the SCU is in the Idle mode (Run switch down), PHI stops. Single pulses are generated by the C-Bus, MAR, MCR1, and MCR2 Alter switches and the Step switch.

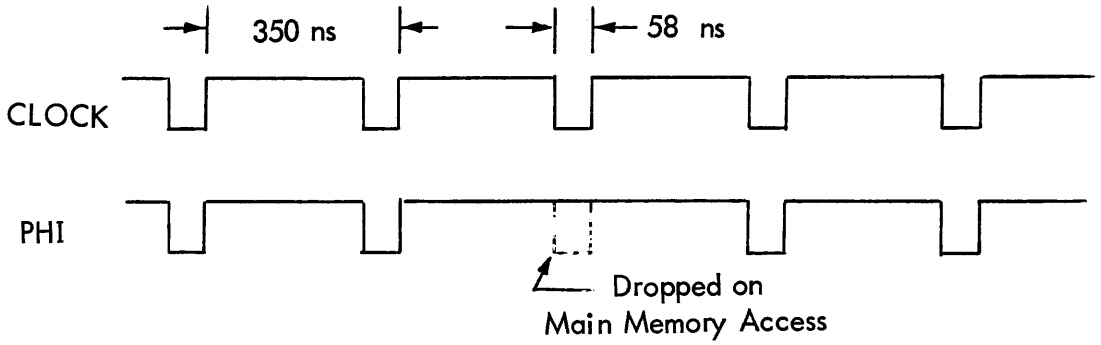


Figure 4-2. Timing Signals

PHI must be buffered on each I/O Module, using a Schottky AND gate located as close to module pin J01 as possible. The maximum stage delay for this gate should not exceed 7.5 nanoseconds. If more than 10 unit loads must be driven by this buffer, a second, parallel buffer may be used. The total load presented by these buffers to the PHI line must not exceed 4.0 ma.

NIMHERE. Must be driven low by each I/O module when IO Mode = 1 and RSA01-RSA03 and EMIT28-31 contain the module's device address. NIMHERE must be driven by an open collector TTL gate, such as Xerox TN510. Used to control the C-Bus buffers in an I/O expansion chassis.

NFIRQ1, NFIRQ2. Fast Interrupt Request 1 and 2. Driven low to request that a fast interrupt be generated. Must be driven by an open collector TTL gate, such as Xerox TN510. Fast interrupts are discussed in Subsection 4-18.

FI-1, FI-2. Fast Interrupt response. Generated by the SCU in response to the corresponding Fast Interrupt Request. The highest priority device with a fast interrupt pending may use this signal to remove its request signal.

FI1PRIn, FI2PRIn, FI1PRIn+1, FI2PRIn+1. Fast Interrupt Priority strings. See Section 4-18 for details. I/O modules not using the fast interrupts must short FI1PRIn to FI1PRIn+1 and FI2PRIn to FI2PRIn+1.

NIOINT. Multiplexed Input/Output Interrupt request line. Driven low to request an interrupt. Must employ an open collector TTL gate, such as Xerox TN510.

NINTLOCK. Interrupt Lock. Signal generated by the SCU to freeze interrupt requests. No new requests can be initiated while NINTLOCK is low. Used to allow the multiplexed I/O interrupt priority string to settle before device address is read to the A-Bus by the processor.

PRIn, PRIn+1. Multiplexed I/O interrupt priority string. PRIn is the priority input line; PRIn+1 is the priority output line. Input/output modules not using the multiplexed I/O interrupt must short PRIn to PRIn+1.

NEXT. External condition common line. When an I/O module recognizes its devices address on RSA01 - RSA03 and EMIT28-31, it should gate its external condition line or lines, if any, into NEXT, using an open collector TTL gate equivalent to Xerox TN510 or TN509.

NINHB. Inhibit B. I/O modules using the fast interrupts are required to drive NINHB low, using an open collector TTL gate, such as Xerox TN510 when placing the memory address on the B-bus. INHB inhibits the general register selected by RB Select from being gated to the B-bus.

POWER. +5V ±5 percent, Logic Ground, +15v ±2 percent, -15v ±2 percent are available at the backplane connector.

4-8 DATA TRANSFER

This subsection describes a method for addressing an I/O module and rules for transferring data into and out of the SCU.

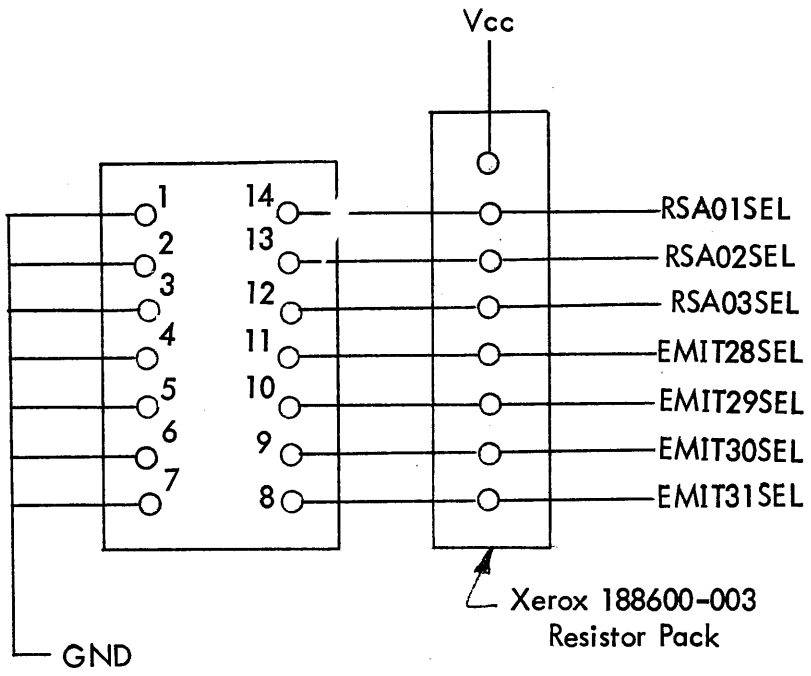
4-9 Addressing

Each I/O module must respond to one or more device addresses. The device address is made up of the RA Select field (RSA01 - RSA03) and the Device Select field (EMIT28-EMIT31), where RSA01 is the most significant bit and EMIT31 is the least significant. The particular address or addresses to which the module is to respond must be assignable at the time the module is installed by means of push-on jumpers, switches, or some other method which can be quickly accomplished.

For the sake of consistency all I/O modules should use the same method for address selection. Therefore, the following is recommended. The selection mechanism is based on a 14-pin dip pattern etched on the printed circuit board as shown in Figure 4-3. A low-profile 14-pin DIP socket, Xerox 151228-001, is stuffed into the pattern. The module address is defined by placing preformed jumpers across the socket, as shown, for each bit of the device address which is to be false. The socket may be located anywhere on the I/O module, but must be identified as "ADDR SEL" on the module silk screen.

The logic recommended for address decoding is shown in Figure 4-4. Timing is very critical in address decoding, as will be seen in the discussion of timing for multiplexed I/O interrupts. Therefore, propagation delay through the address decoding logic should be kept as short as possible, not to exceed 30 nanoseconds.

The signal IOMME (IOM Module Enable), generated as shown on Figure 4-4, is used to enable the I/O module to respond to any command from the IO Control field, including the transfer of data to the A-Bus and the enabling of the C-Bus input buffers. ME (Module Enable) is used primarily to enable the external condition line or lines onto the NEXT line. NIMHERE is used to control the C-Bus buffers in the I/O expansion chassis.



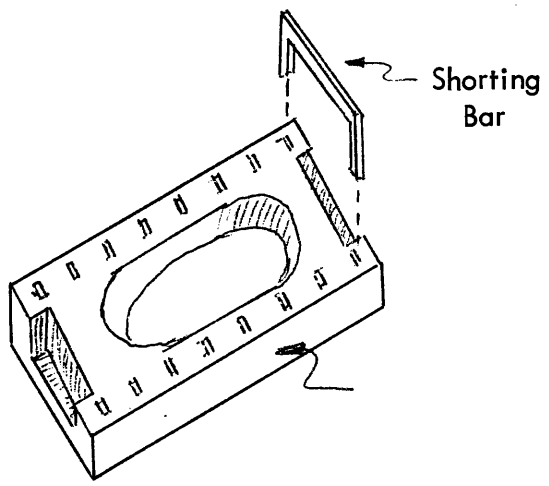


Figure 4-3. Recommended Address Selection Method

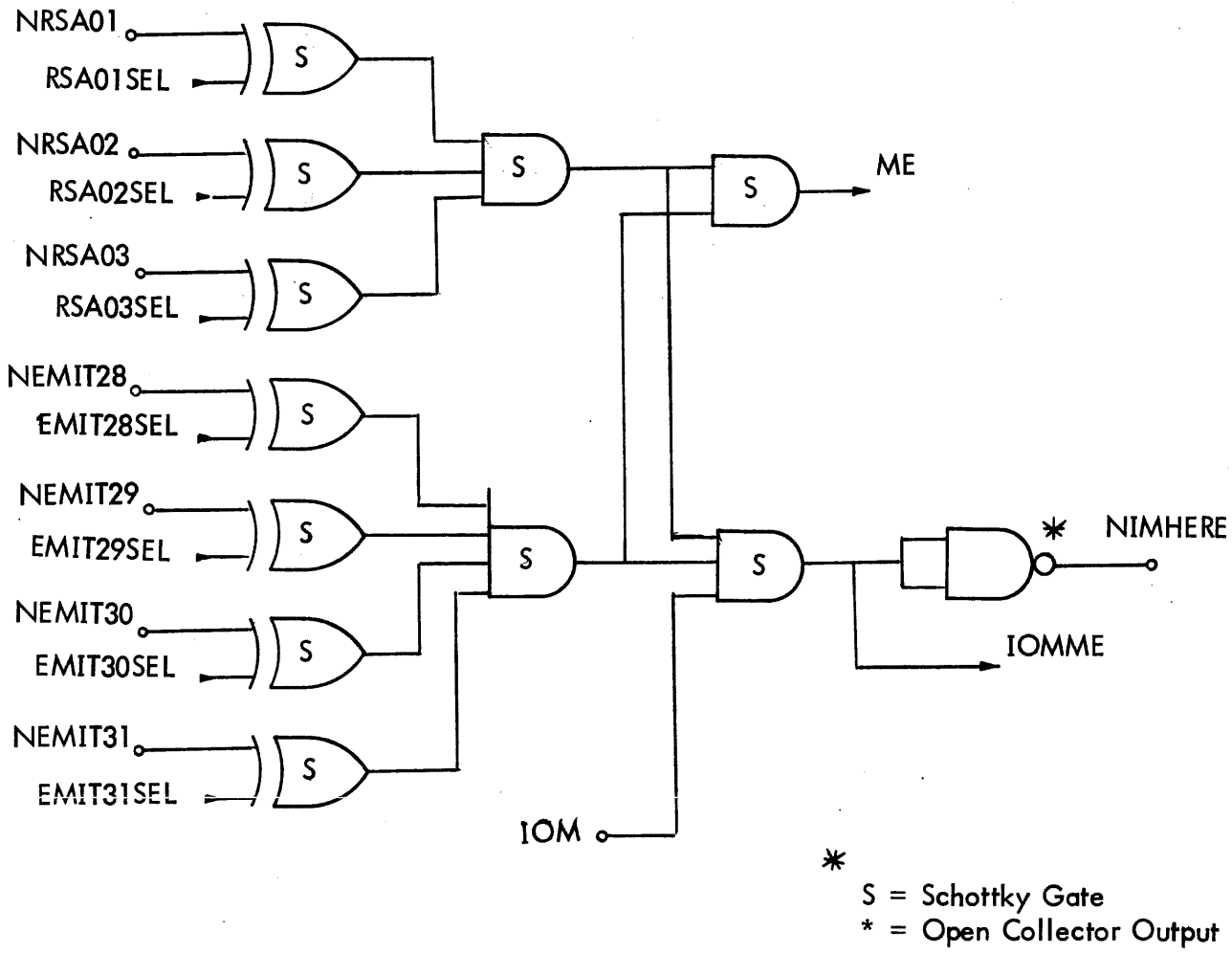


Figure 4-4. Recommended Address Decode Logic

4-10 A-Bus Connect Rules

The A-Bus provides the path users must take when inputting data to the SCU. This path allows the user to enter data on one side of the Arithmetic Logic Unit, perform some function on that data, and strobe the result back to the user from the C-Bus - all during one SCU cycle time.

Transfers from the input module to the A-Bus must be disabled and disconnected except when the module is selected and NAXIOCONT is high. This type of gating is accomplished by the use of tri-state logic, such as Xerox 200916, 200920, 200936, or equivalent.

The logic for transferring input data to the A-bus can be as follows:

Enable Data to A-Bus = IOMME. INPUT. NAXIOCONT where

IOMME = IO module selected by micro instruction, as indicated in Figure 4-4

INPUT = Designer-assigned configuration of the IO Control field, EMIT24 - 27 specifying either data or status information input to the SCU

NAXIOCONT = SCU IO to A-Bus enable signal

Figure 4-5 describes the A-Bus logic rules. In addition to the tri-state buffers illustrated, other tri-state devices such as multiplexers may be used to drive the A-Bus. Timing requirements are shown in Figure 4-6.

4-11 B-Bus Connect Rules

The B-bus provides the path for users to take when supplying a direct memory address to the SCU when using the fast interrupts, as explained in subsection 4-18. The logic rules and timing for connection to the B-bus are identical to those for connection to the A-bus, except that the control equation is modified as follows:

Enable Data to B-Bus = IOMME. DMAINPUT where

DMAINPUT = designer-assigned configuration of the IO Control field specifying direct memory access

4-12 C-Bus Connect Rules

The C-Bus is the path for transferring data and control information from the SCU to an I/O module. Data on the C-Bus is available for any I/O module to acquire. In order to minimize loading on the C-Bus, I/O modules not selected and actively receiving data from the C-Bus must be in a disabled mode. This type of gating is provided by tri-state logic which, in the disabled state, imposes very little load on the C-Bus.

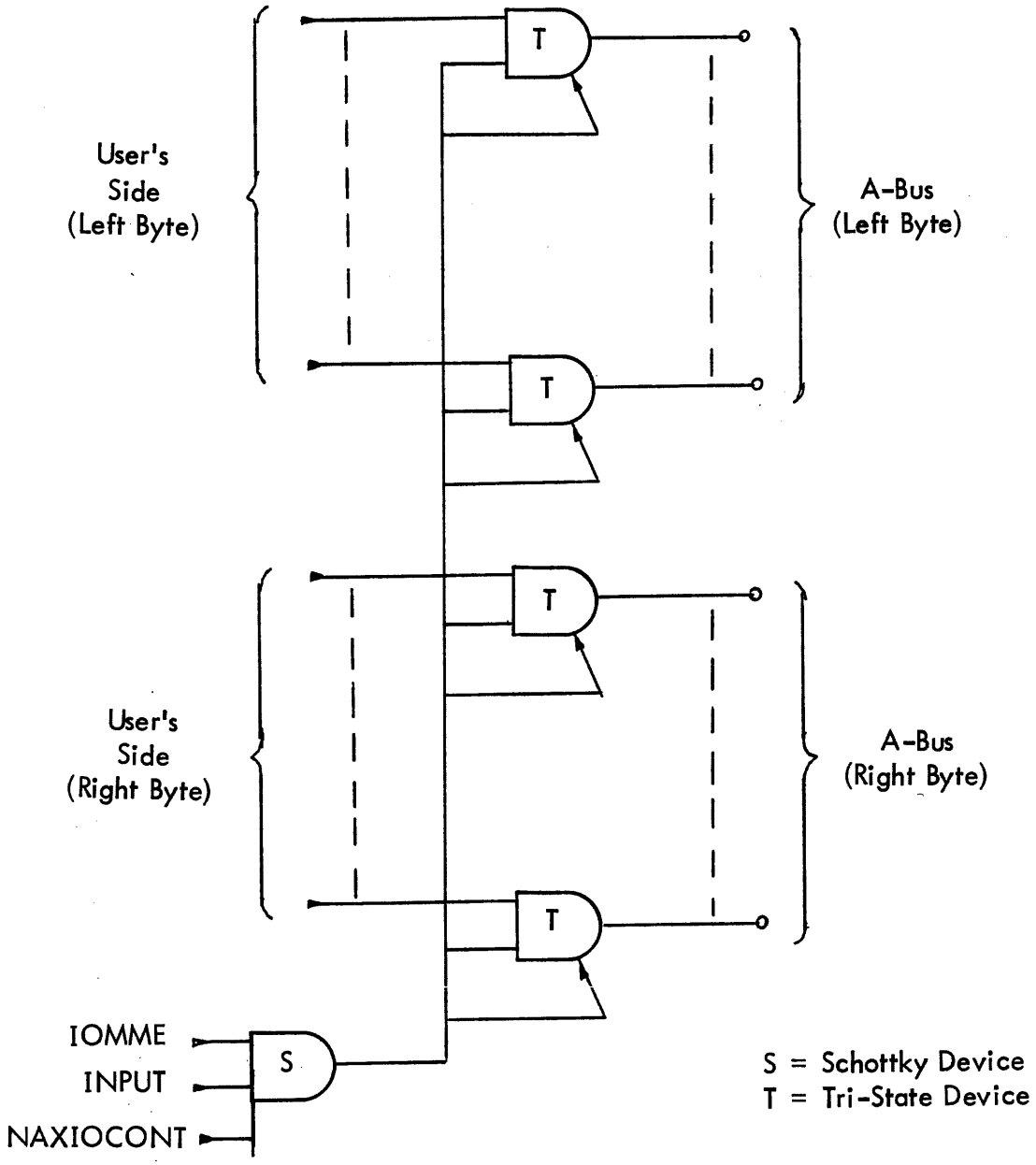


Figure 4-5. A-Bus Connect

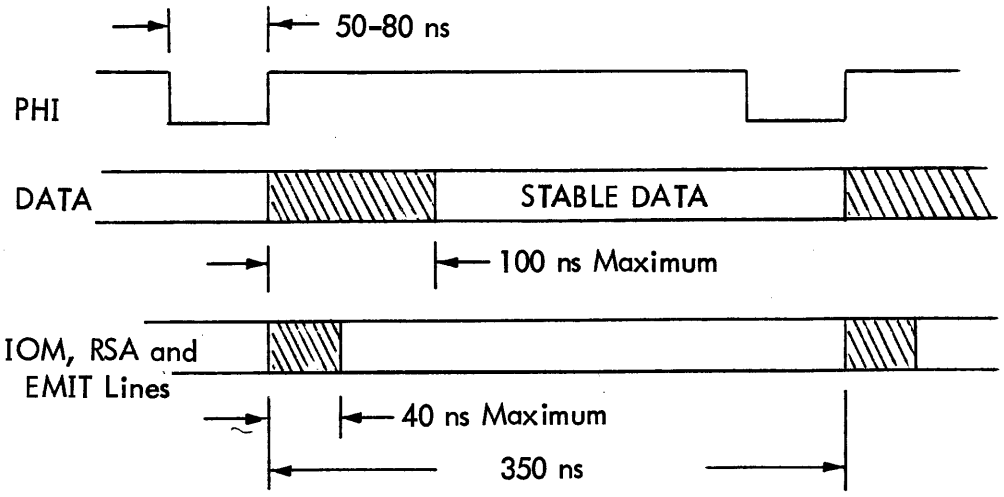


Figure 4-6. A-Bus or B-Bus Timing Diagram

The logic for transferring output data from the C-Bus is as follows:

Enable C-Bus to I/O Module = IOMME where

IOMME = I/O module selected by micro instruction, as shown in Figure 4-4.

The C-Bus logic rules and timing requirements are shown in Figures 4-7 and 4-8.

4-13 I/O INTERRUPT

The SCU accommodates two types of interrupt systems: fast interrupts for high-speed data transfers in or out, and multiplexed input/output interrupts. The fast interrupt system returns a hardware response as soon as the interrupt has been accepted, while the I/O interrupt system does not provide a hardware acknowledge signal. The I/O interrupt system is discussed in this subsection and the fast interrupt system, in subsection 4-18.

The SCU accepts up to 128 inputs to the multiplexed input/output interrupt, corresponding to the 128 device addresses. This interrupt can be enabled or disabled by micro opcodes in the MCR Control field. Requests can be asynchronous with respect to the system clock and do not interrupt the execution of a micro instruction during its clock period. An interrupt does take control at the end of the cycle, providing it is not disabled.

The request for an I/O interrupt is transmitted to the SCU by means of the signal NIOINT. Then the SCU transfers the device address of the I/O module with the highest priority which has an I/O interrupt pending to the A-Bus and responds accordingly.

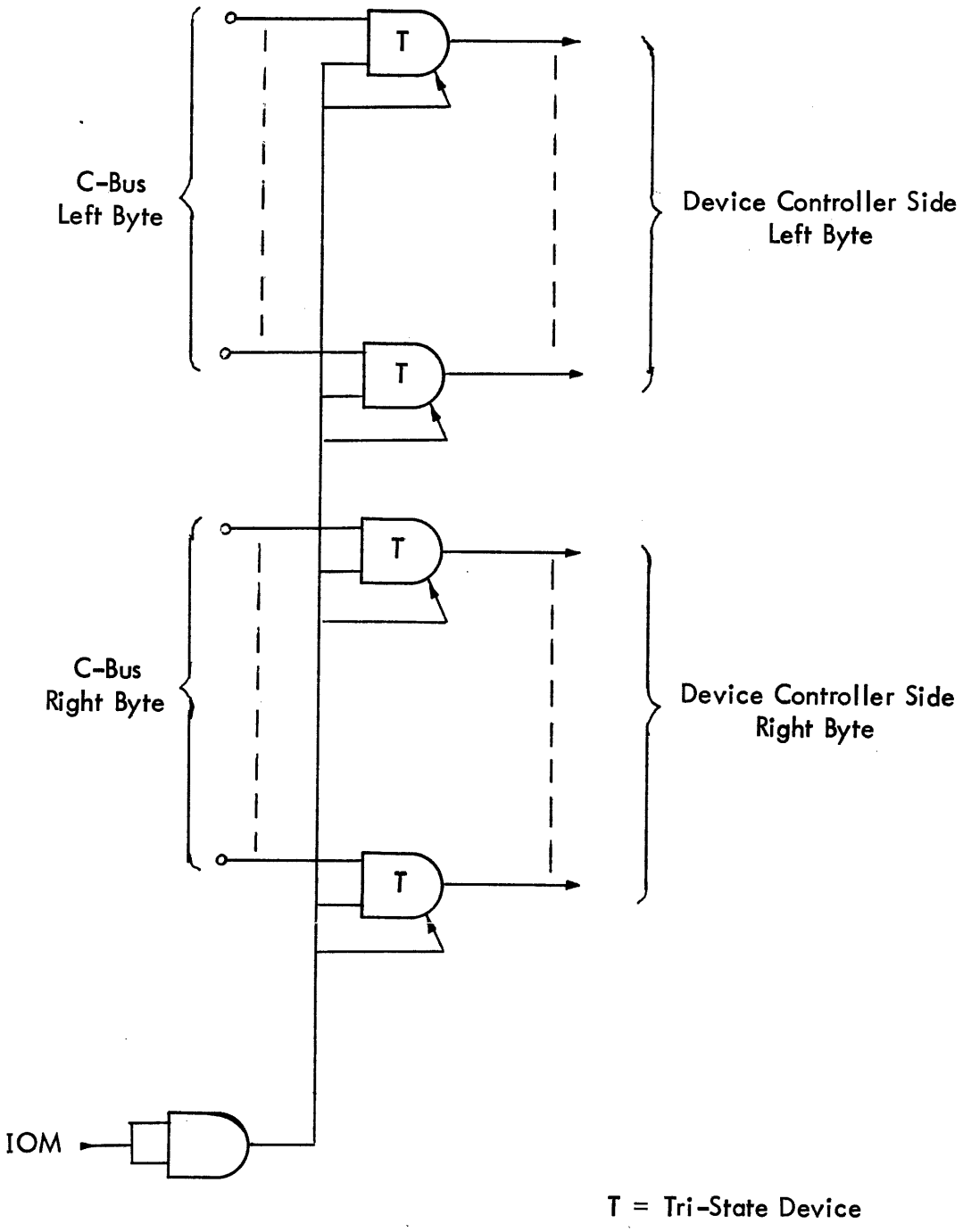


Figure 4-7. C-Bus Connect

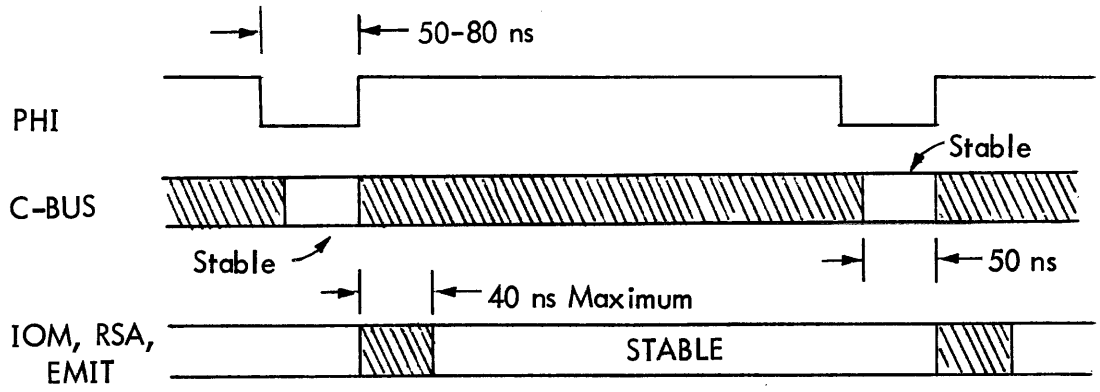


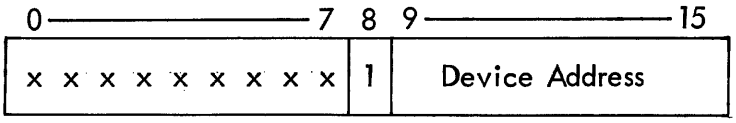
Figure 4-8. C-Bus Timing

4-14 I/O Interrupt Operation

The SCU provides a single input line, NIOINT, which can be pulled low by any I/O module to signify that the module has an I/O interrupt pending. In the SCU, NIOINT low sets the I/O Interrupt flip-flop on the next system clock. The flip-flop output is applied to the Interrupt Priority logic and then to logic that generates control memory address x'8'. As soon as this address becomes available, it is used to reset the flip-flop. Thus, the I/O Interrupt flip-flop remains high until the I/O interrupt is reached in priority order.

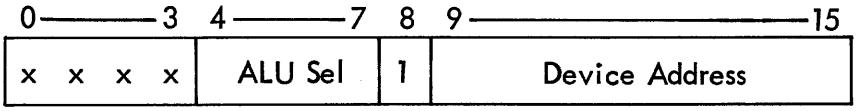
The next PHI, after control memory address x'8' becomes available, loads the micro instruction at this location into the micro control register for execution. Thus if no higher priority interrupt interferes, the micro instruction is loaded into the micro control register by the second PHI after NIOINT goes low. This micro instruction should contain the Push micro opcode in the Branch Control field and the Immediate Address micro opcode in the C-Bus/Register Control field. The Immediate Address micro opcode, it will be recalled (see subsection 3-10) is interpreted by the SCU in two ways, depending on whether I/O Mode is true or false.

If IO Mode = 1, the device address of the highest priority I/O module with an interrupt request pending is gated to the A-Bus in the following format:



Bits 0-7 should not be driven by the I/O module, and bit 8 must be forced to 1 by the I/O module.

The SCU then appends to the A-Bus the contents of the ALU Select field from the micro instruction to the above Device Address to generate an address as shown below:



This 12-bit address, now on the A-Bus, is used by the SCU in the same way as an Immediate Address (which is gated from the micro control register to the A-Bus). In either case, the control memory jumps to the 12-bit address. Therefore, the Push micro instruction branches to a table in control memory between x'080' and x'0FF', assuming ALU Select = 0. The size of this table depends on the number of device addresses used in the system.

The contents of this table are a series of unconditional branch instructions with Immediate Addresses which cause the SCU to branch to the first micro instruction in the device handler interrupt routine. Thus, the total time from NIOINT going low (assuming no interference from a higher priority interrupt) to the start of execution of the first micro instruction of the interrupt handler is 1.05 microseconds minimum, to 1.40 microseconds maximum. The processor execution time required is 0.70 microseconds.

Alternatively, if I/O Mode = 0 in the original Push micro instruction, the Immediate Address bits from the micro instruction are used as the branch destination. In this case the device address of the interrupting I/O controller can be recovered later by executing a micro instruction with MCR = x'E'.

This micro opcode causes the I/O device controller to gate its device address to the A-Bus in the format shown above.

4-15 Interrupt Disable

The multiplexed I/O interrupt is automatically disabled by the SCU when any type of interrupt except the fast interrupts is requested. The disable occurs at the time of execution of the micro instruction found in the interrupt location in control memory. If the micro instruction contains a Push micro opcode, the current status of the multiplexed IO enable flip-flop is stored in the push stack before the flip-flop is reset. At the end of the interrupt handler subroutine, a Pull micro opcode in the final micro instruction restores the enable flip-flop to its original status.

Also, once an interrupt is requested, the signal NINTLOCK goes low. This signal must be used on the I/O module to prevent the initiation of an I/O interrupt request until it goes true. An I/O module that is already requesting an I/O interrupt must continue its request. The purpose of NINTLOCK is to freeze the priority chain so that the I/O device controller which had the highest priority at the time of interlock will still have the highest priority when the time comes to gate its device address to the A-Bus.

The I/O device controller maintains its interrupt request until an I/O command is received by the controller to remove the request. The I/O command can be accomplished either by use of the IO Control field or by means of a command word via a C-Bus transfer.

4-16 I/O Interrupt Logic

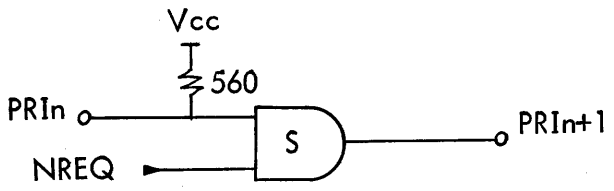
The logic functions necessary to implement I/O Interrupt are the priority chain, the interrupt request, and gating the device address to the A-Bus.

The priority chain, or string, consists of an input signal to each I/O module, PRIn, and an output signal from each module, PRIn+1. The equation is:

$$PRIn+1 = PRIn.NREQ$$

where NREQ = no interrupt request pending

This equation must be implemented by each I/O module that uses the multiplexed I/O interrupt, using a Schottky AND gate with a maximum propagation delay of 7.5 nanoseconds, as follows:



If an I/O module does not employ the I/O interrupt capability, the pin for PRIn must be shorted to the pin for PRIn+1 on the module.

The interrupt request, NIOINT, may be implemented as shown in Figure 4-9 although this approach is not the only possible one. The only criteria are:

a. NIOINT must be driven by an open collector TTL gate, Xerox TN510, TN511, or equivalent.

b. No new request may be initiated while NINTLOCK is low, but all requests initiated prior to NINTLOCK going low must be maintained until reset under program control.

It is not necessary that NREQ be synchronous with PHI.

The device address must be gated to the A-Bus using the rules for A-Bus connection set down in subsection 4-8. The signal controlling the tri-state bus drivers must adhere to the following equation:

$$AXDA = REQ.PRIn.AXDASTROBE$$

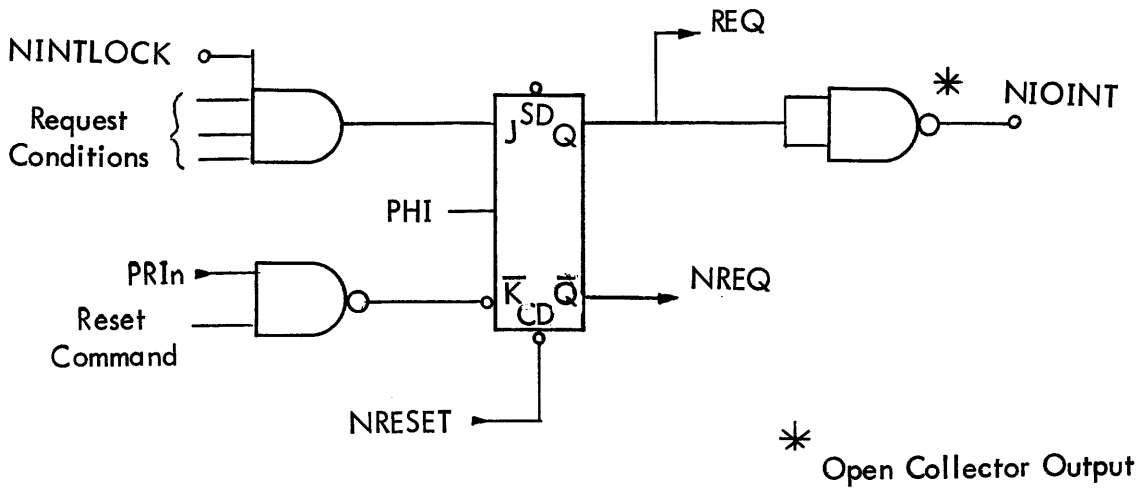


Figure 4-9. IO Interrupt Request

where AXDA = Device Address to A-Bus transfer control term

AXDASTROBE = Device Address to A-Bus Strobe signal from SCU.

Figure 4-10 shows a typical implementation. Schottky logic should be used for the control signal in order to meet timing requirements. Tri-state multiplexers could be used in place of the tri-state buffers.

The relationship between two I/O modules in an I/O interrupt priority chain is shown in figure 4-11. The priority pins are prewired in such a way that the ranking of an I/O module in the priority string is determined by the physical location into which it is plugged. The lowest ranking or highest priority devices are closest to the SCU.

Figure 4-12 is a timing diagram of the typical relationships between the I/O interrupt interface signals. Two considerations must be observed:

a. The maximum propagation delay between PRIn and PRIn+1 must not exceed 7.5 nanoseconds worst case.

b. The time requirements for gating the device address to the A-Bus must comply with the specifications of Figure 4-6. This means that the device address must be stable on the A-Bus not more than 100 nanoseconds after AXDASTROBE goes high and no longer than 45 nanoseconds after NRSA01-03 and NEMIT28-31 signals go low.

4-17 EXTERNAL CONDITION TEST

The Branch Control field of the micro instruction contains two micro opcodes used to test an external condition present on an I/O Module:

x'4' Test/Branch On External Condition Set
x'5' Test/Branch On External Condition Reset

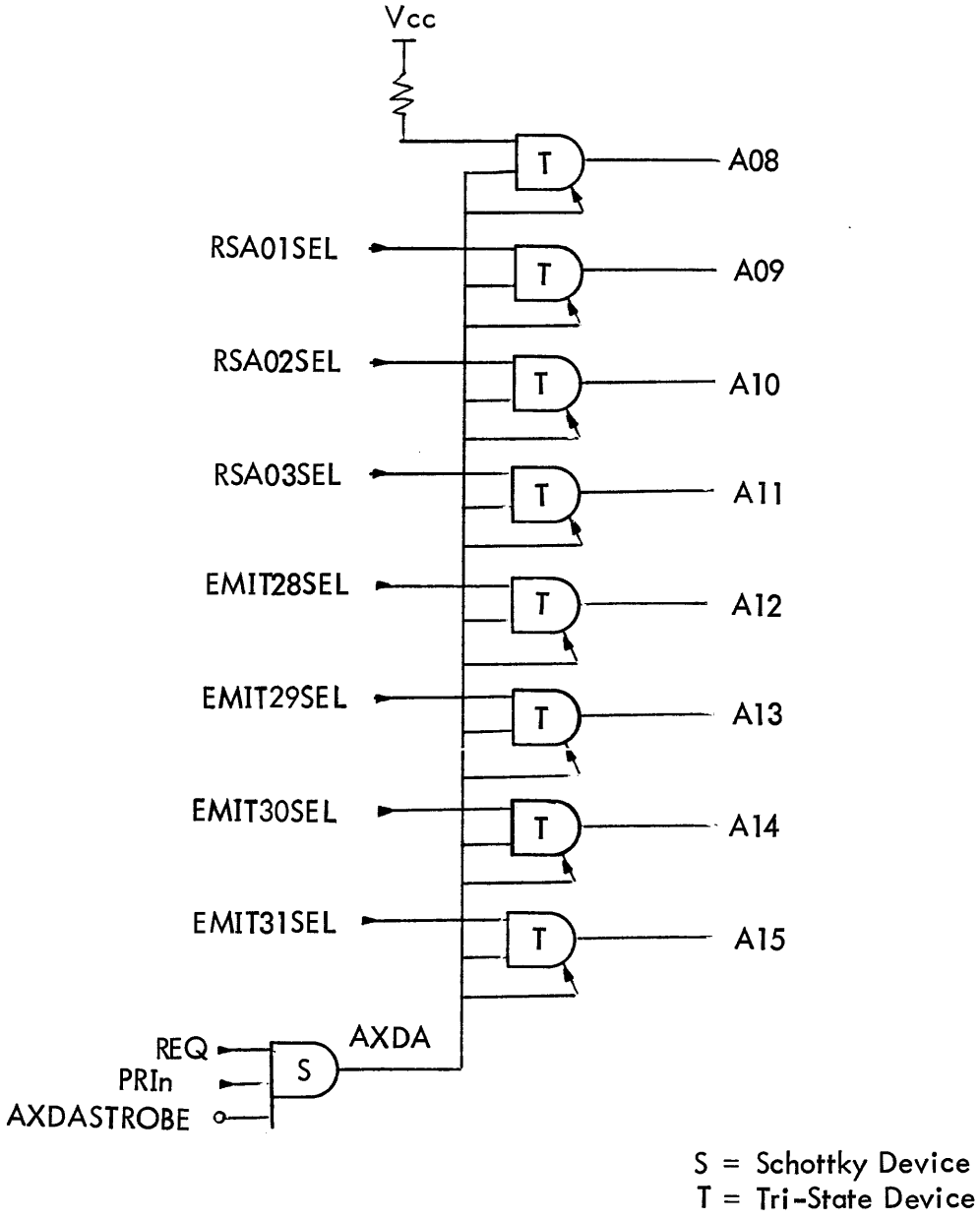


Figure 4-10. Device Address to A-Bus Logic

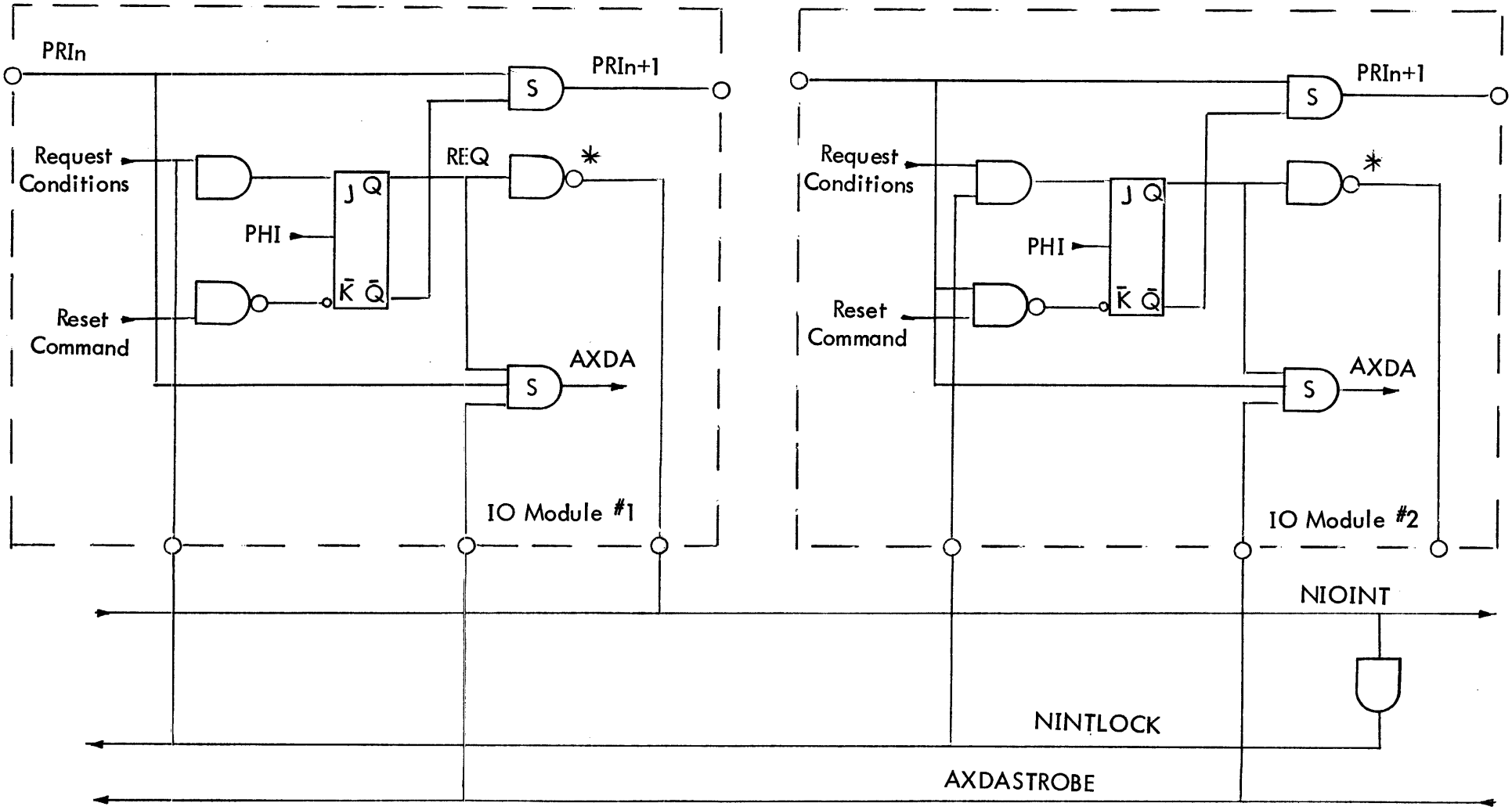


Figure 4-11. I/O Modules in Multiplexed Interrupt Priority String

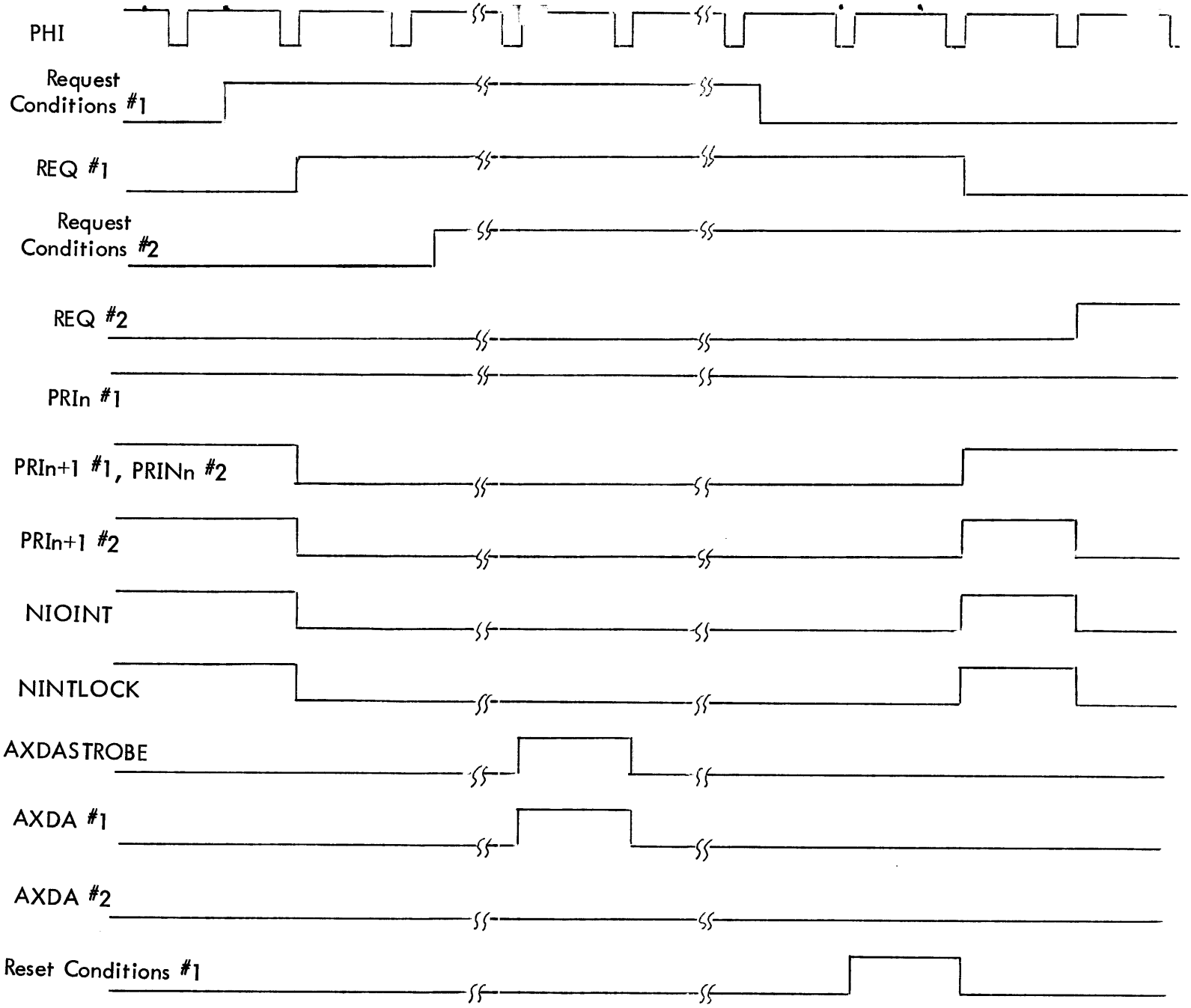


Figure 4-12. Typical Multiplexed I/O Interrupt Timing

Each I/O module may provide one or more external status conditions to a common signal, NEXT, when RA Select and Device Select fields contain an IO address assigned to the module. Moreover, more than one external condition for each I/O address may be selected and multiplexed to NEXT by use of additional signals such as designer-assigned micro opcodes in the IO Control field. On the other hand, the external condition test capability can be ignored if not required in a particular application.

The operations in the SCU resulting from these two Test/Branch micro opcodes are set forth in Section 3. However, if a test is successful, the least significant 12 bits of the C-Bus serve as the next micro address in a procedure requiring two clock periods. If the test is unsuccessful, the micro address increments in sequence, requiring one clock period. Moreover, because the Device Select field is employed in addressing the external conditions, the Immediate Address field can not be used to supply the new micro address.

Logic requirements for the external condition interface are:

a. The NEXT line must be driven by an open collector TTL gate.

b. The external condition lines must be synchronized with PHI so that they cannot change during the clock period in which they are being tested.

c. NEXT must be stable within 100 nanoseconds after the rising edge of PHI, as depicted in Figure 4-13.

Figure 4-14 shows a typical logic implementation for a single external condition status line. ME is the decoded device address, as derived in Figure 4-4.

4-18 FAST INTERRUPTS

The SCU is capable of accepting two fast interrupt requests from every I/O module location. The fast interrupts are used primarily for high-speed data transfers between I/O devices and scratch pad/main memory. The fast interrupts differ from the multiplexed I/O interrupts in two respects. First they automatically return a response signal and, second, they cycle-steal a single cycle from SCU operations.

During this cycle the SCU executes a single micro instruction that typically transfers a word or byte from an I/O bus to scratch pad/main memory, or from scratch pad/main memory to an I/O bus. The scratch pad/main memory address may come from a general register (which is incremented by the same micro instruction), or from the I/O module (device controller).

Cycle stealing refers to the insertion of an out-of-sequence micro instruction into the ordinary sequence of SCU operations without affecting the status bits or the next micro address. The state of the SCU is not stored in the push stack and, thus, does not require removal from the push stack by a later micro instruction in order to resume the interrupted sequence. Rather, after the clock cycle in which the out-of-sequence micro instruction is executed, the SCU can resume operation directly on the next micro instruction. The effect is that the fast interrupt capability can be employed to achieve direct access to scratch pad/main memory, or DMA.

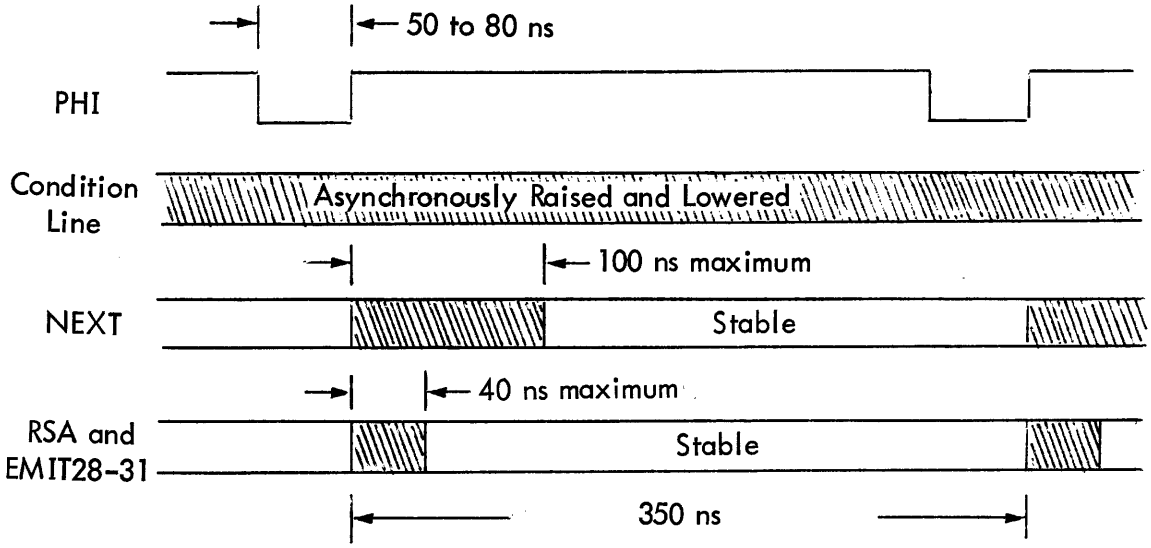


Figure 4-13. External Condition Timing

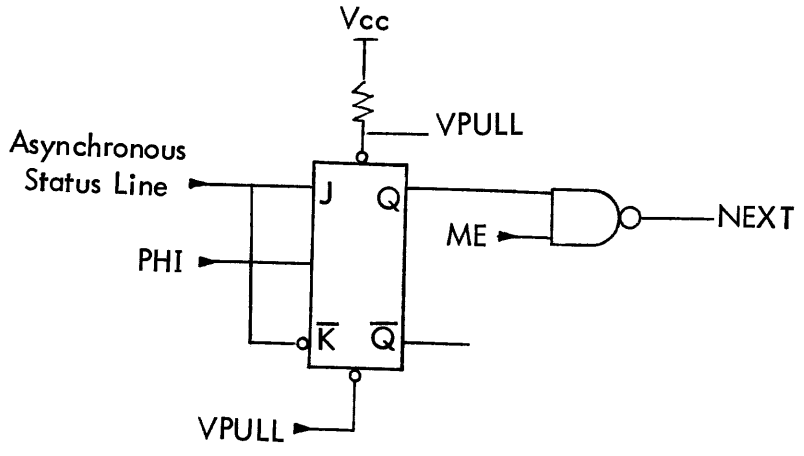


Figure 4-14. Typical External Condition Logic Circuit

Each fast interrupt is limited to stealing no more than one cycle out of every two. Thus, in making transfers to or from scratch pad with the clock period at 350 nanoseconds, each transfer takes 700 nanoseconds, equivalent to a maximum throughput rate of 1.428 megawords per second. Transfers to or from main memory can occur every 1.05 to 1.4 microseconds, depending on whether the micro instruction executed between two interrupt micro instructions is a one- or two-cycle instruction. This is equivalent to a maximum throughput rate of 714 kilowords per second. If both fast interrupts are running simultaneously at maximum throughput rate, any other SCU micro instruction execution is locked out; that is, one hundred percent of the SCU bandwidth is used by the fast interrupts.

Two approaches are possible in using the fast interrupt capability. The first requires less hardware, but limits the use of the fast interrupts to a single I/O device. The second approach requires a little more hardware, but shares the fast interrupt capability among many I/O devices.

4-19 Logic and Timing

Receipt of Fast Interrupt Request 1 or 2 from an I/O module causes Fast Interrupt flip-flop 1 or 2 in the SCU interrupt logic to set on the next PHI and to return Fast Interrupt response 1 or 2 to the I/O module. One possible logic arrangement is shown in Figure 4-15, and the timing in Figure 4-16.

The true output of the fast interrupt flip-flop is applied to interrupt priority logic and then to logic that generates control memory address x'6' for Fast Interrupt 1 or x'7' for Fast Interrupt 2. As soon as this address is available, signifying that the fast interrupt has been accepted by the SCU, it is used to reset the fast interrupt flip-flop.

Figure 4-16 shows the fast interrupt interface timing for a single data transfer or for multiple transfers at relatively slow throughput rates. Figure 4-17 shows timing at maximum throughput rates. Times shown on the figures are measured on the backplane at the I/O module. At maximum throughput, the interrupt request generated by the I/O module, NFIRQn, may be held low for the entire duration of a block transfer. At the end of the transfer, NFIRQn must be raised within 180 nanoseconds after the falling edge of the response pulse, FIn, for the last word to be transferred. Raising NFIRQn within this limit assures that an extra interrupt is not triggered.

Two connect rules are pertinent; first, NFIRQn need not be synchronized with PHI, and second, NFIRQn must be driven by an open collector TTL gate, Xerox TN510 or equivalent.

Additional implementation arrangements are described later in this section.

4-20 Single High-Speed Device

In this approach both fast interrupts are connected to a single high-speed device controller, or I/O module. One may be used for transfers-in and the other, for transfers-out. In addition, a multiplexed I/O interrupt capability is often required to enable the I/O module to initialize and terminate a series of transfers.

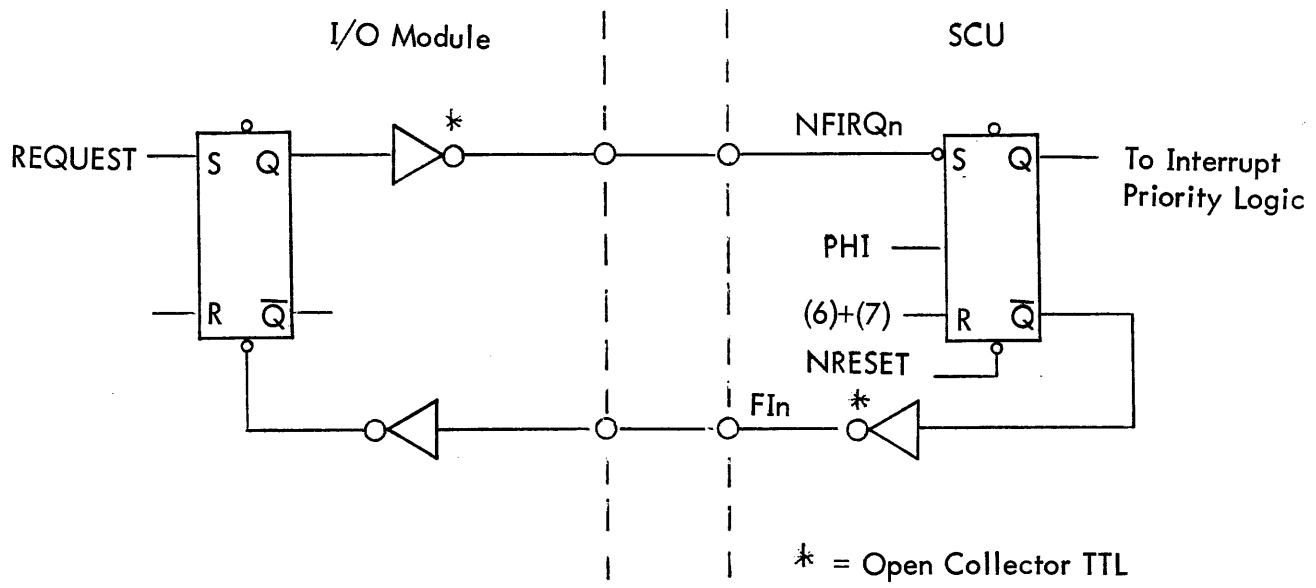
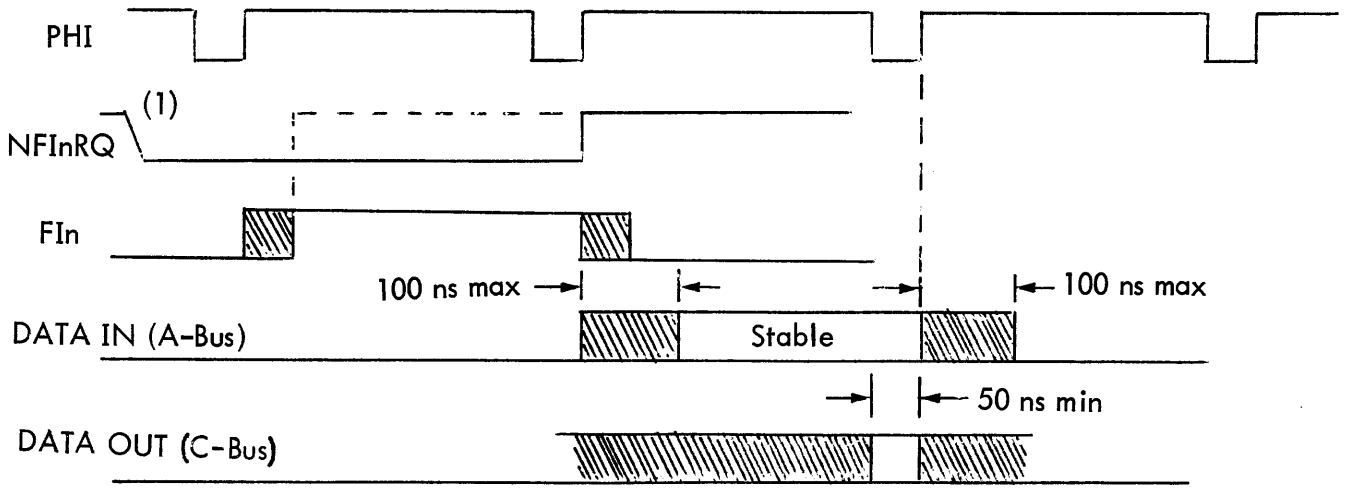
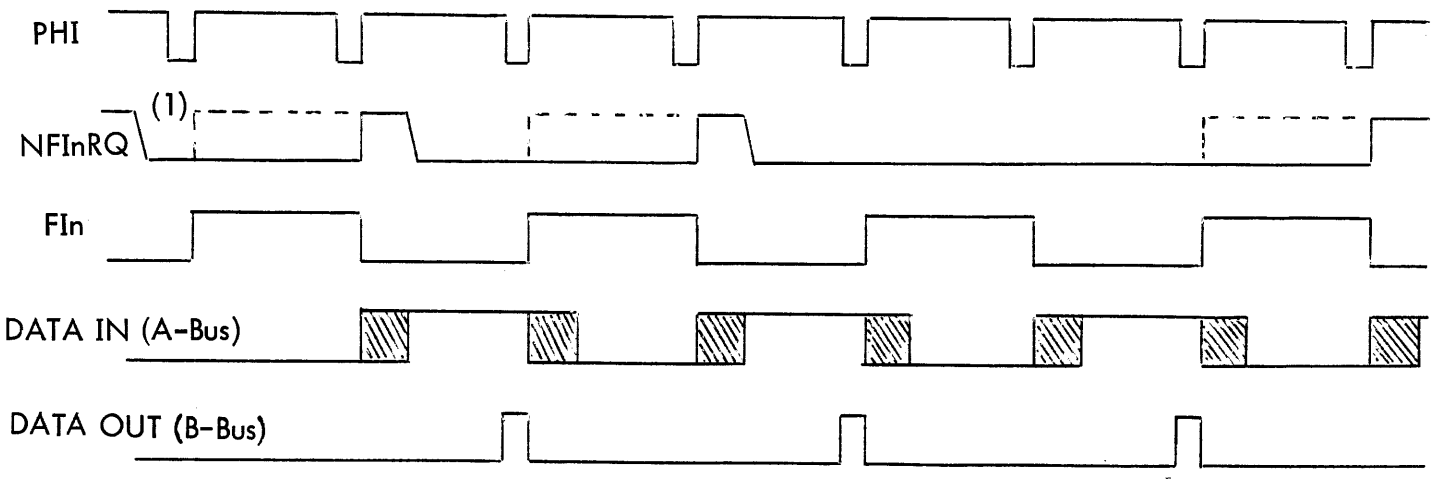


Figure 4-15. Fast Interrupt Request and Response.



(1) NOTE: NFInRQ may be switched high any time after rising edge of FIn.

Figure 4-16. Fast Interrupt Timing for Single Data Transfer



(1) NOTE: For maximum throughput, request line may either be pulsed or held low.

Figure 4-17. Fast Interrupt Timing for Multiple Data Transfer

Before starting fast interrupt data transfers it is necessary to have the address of the first scratch pad/main memory location to be used in a general register, where it is available to the B-Bus. The general register employed for this address is the one defined by the RB Select field of the micro instruction in Fast Interrupt location x'6' or x'7'. This general register, now serving the function of scratch pad/main memory address register, is incremented (or decremented) by the interrupt micro instruction on each transfer to supply the new scratch pad/main memory address.

Each time the user requests service, the I/O module generates NFIRQn, the SCU returns an automatic response, and the micro instruction in location x'6' or x'7' is executed. This single micro instruction typically transfers data to or from the I/O module into or out of the scratch pad/main memory location specified by the scratch pad/main memory address register, and increments or decrements this register.

A series of fast interrupt transfers is normally terminated by a multiplexed I/O interrupt. For example, before starting data transfers, the firmware may transmit a word count via the C-Bus to a word-count register on the I/O module. When this count is reached, the I/O module may initiate a multiplexed I/O interrupt to terminate the transfer.

4-21 Multiple Devices

In this approach both fast interrupts are connected to two or more I/O modules (device controllers), priority is determined by a priority chain between the I/O modules, and a scratch pad/main memory address register, located on each I/O module, supplies the memory location. The next four subsections present a standardized approach to this design, not because other implementations are not possible, but because compatibility among a number of I/O devices that may share the fast interrupts is desirable. With this approach, direct access to memory (DMA) for a number of I/O devices can be accomplished.

4-22 Direct Memory Access

Direct Memory Access (DMA) refers to the ability to transfer data from an I/O device into main memory or from main memory to an I/O device with a minimum of interference with the on-going SCU operations. Using the fast-interrupt approach, one fast interrupt is treated as a scratch pad/main memory write request, and the other as a memory read request. So far, this use of the fast interrupts is no different from the approach for a single high-speed device.

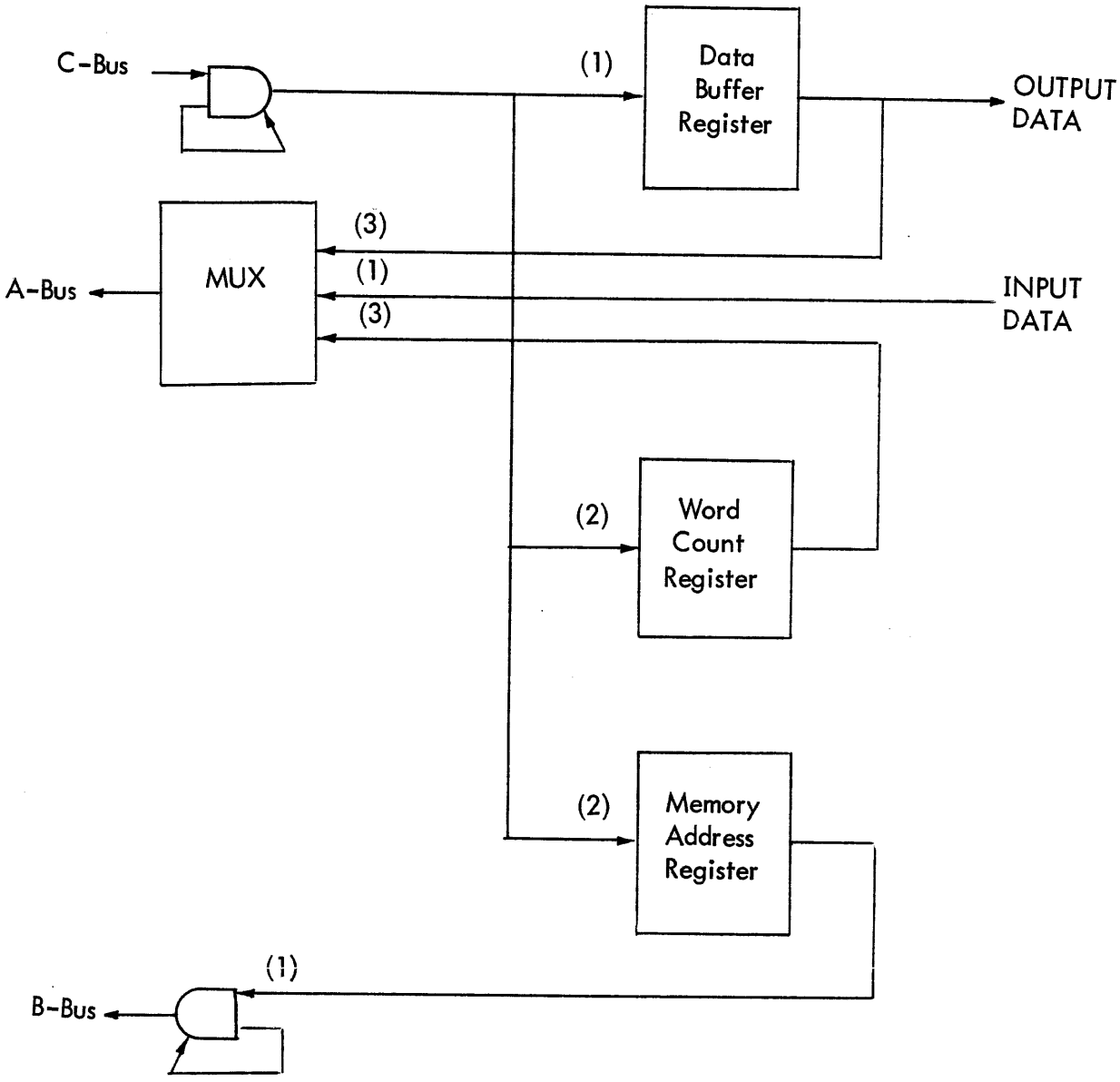
In the multiple-device approach, however, each I/O device controller has a scratch pad/main memory address register and a method of getting the register contents on to the B-Bus when the interrupt micro instruction is executed. Moreover, all the device controllers are assigned a common I/O address, called the DMA channel address. Priority among the device controllers on the DMA channel is resolved by a priority chain. However, each device controller also responds to a different and unique I/O address for other purposes, such as initializing and terminating a transfer operation.

In operation, then, the micro instructions in the fast-interrupt control-memory locations address the common DMA channel, but only the device controller with the highest priority assignment of those ready to transfer or receive data responds. The response consists of placing the current address from its scratch pad/main memory address register on the B-Bus and, (in the case of transfers-in) placing input data on the A-Bus or, (in the case of transfers-out) acquiring data from the C-Bus. During the same micro instruction period, the SCU uses the memory address on the B-Bus to write data from the A-Bus into scratch pad/main memory or to read data from scratch pad/main memory to the C-Bus. Thus, in DMA operation, a single micro instruction accomplishes a transfer-in or a transfer-out.

The I/O device controller in the next highest priority position, if one is ready, can make a transfer in the second-next micro instruction period. In this way, a number of I/O devices can interleave data transfers by means of the fast interrupt capability. Any number of I/O devices can use this DMA interface; however, a limit is set by the necessity to keep the combined bandwidth of the devices within the DMA channel bandwidth of 1.428 megawords per second, scratch pad, or 714 kilowords per second, main memory.

The essential elements of an I/O module for the DMA interface, shown in Figure 4-18, are a data buffer register, word count register, memory address register, and multiplexer to A-Bus. The common DMA channel address is used for data-transfer purposes, denoted by (1) on the diagram. The second address, unique to each I/O device controller, is used for passing control information such as device initialization, word count register setup, memory address register setup, to the device controller, and for passing status information to the SCU. These purposes are denoted by (2) on the diagram. In addition, the contents of the data buffer register and the word count register can be multiplexed to the A-Bus for test purposes.

Figure 4-19 is a block diagram showing a series of I/O devices and device controllers connected to the tri-bus structure of the SCU. For this example, device address 0 has been arbitrarily assigned to the common DMA channel.



(1) Controlled by DMA I/O command.
(2) Controlled by device controller I/O command.
(3) Same as (2) but for test procedures only.

Figure 4-18. Essentials of Direct Memory Access (DMA)

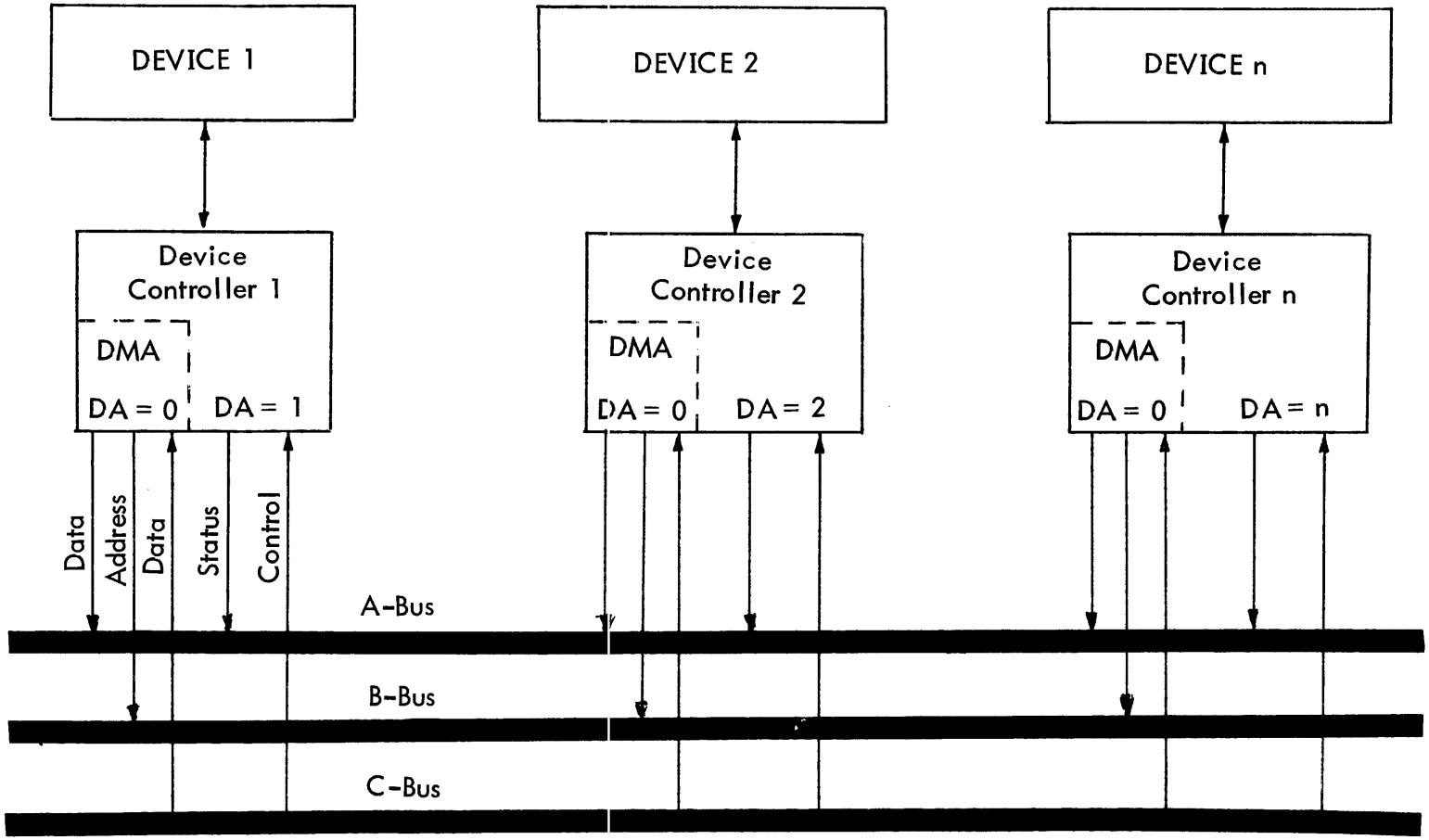


Figure 4-19. Block Diagram, Multidevice Mode

4-23 Programming Direct Memory Access

The two fast interrupts are each dedicated to data transfer in one direction. The higher priority interrupt (control memory location x'6') is assigned to data input to scratch pad/main memory and the lower priority interrupt (location x'7') is assigned to data output from scratch pad/main memory.

Programming the DMA makes use of the following command assignments in the IO Control field. No additions or alterations to this list are allowed.

IO Control					Function
24	25	26	27	Hexadecimal	
0	0	0	0	0	No action
0	0	0	1	1	Read input data from user to A-Bus; transfer memory address to B-Bus.
0	0	1	0	2	Write output data from C-Bus to user; transfer memory address to B-Bus.
1	0	0	1	9	Read data buffer register to A-Bus.
1	0	1	0	A	Write data from C-Bus to data buffer register.
x	1	0	0	4 or C	Disable both fast interrupts.

These commands are described below.

IO Control = x'1', along with IOM = 1 and Branch Control = x'3', is decoded by all I/O modules sharing the DMA channel. The I/O module (with the highest priority having an interrupt pending) supplies both the data to be stored and the memory address where it is to be stored. The data is placed on the A-Bus and the memory address on the B-Bus. The micro instruction for this function is placed in control memory location x'6' and appears as follows:

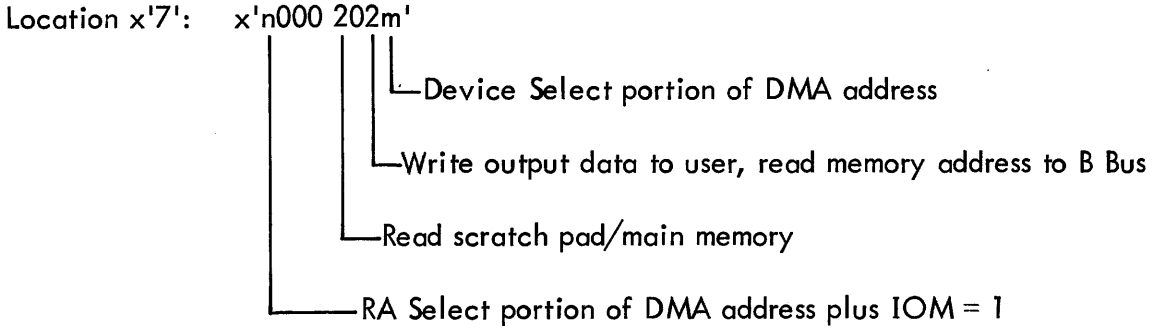
Read data and memory address from user:

Location x'6': x'n000 301m'

- `1` — Device Select portion of DMA address
- `0` — Read data and memory address from IO module
- `1` — Write scratch pad/main memory
- `n` — RA Select portion of DMA address plus IOM = 1

IO Control = x'2', along with IOM = 1 and Branch Control = x'2', is decoded by all I/O modules sharing the DMA channel. The I/O module (with the highest priority) supplies the memory address of the scratch pad/main memory location in which the data for the user is to be found. This memory address is placed on the B-Bus. Data from scratch pad/main memory is taken from the C-Bus by the I/O module for transfer to the I/O device. The micro instruction for this function is placed in control memory location x'7' and appears as follows:

Read scratch pad/main memory address, write data to user:



IO Control = x'4' or x'C', along with IOM = 1, is decoded by all I/O modules sharing the DMA channel. These codes disable the Fast Interrupt Request lines from the DMA channel to the SCU and, consequently, disable the user's ability to write into or read from scratch pad/main memory. These codes would be used in an executive routine or priority driver or in a power shut down sequence. The fast interrupts on each I/O module must be re-enabled separately, using the command/status device address.

IO Control = x'A', along with IOM = 1, is decoded by all I/O modules sharing the DMA channel. This code takes data from the C-Bus and stores it in the data buffer register. A particular data buffer register is selected for test by disabling all I/O modules on the DMA channel by means of IO Control = x'4' or x'C' and re-enabling one I/O module by means of the command/status device address. This operation tests the functioning of the DMA channel, DMA address decoding, and the data buffer register on the selected I/O module. This test is similar to writing data to an I/O module. This code is handled as an in line micro instruction, not one executed from the fast-interrupt control memory location and therefore is not dependent on whether an interrupt is pending.

IO Control = x'9', along with IOM = 1, is decoded by all I/O modules sharing the DMA channel. This code transfers data from the data buffer register to the A-Bus. It is then up to the rest of the micro instruction to determine what to do with the data on the A-Bus. This code is used in conjunction with x'A' to check the data buffer register and DMA address decoding.

Programming for the command/status interface (which is the one addressed by the second, unique address on each I/O module of the DMA type) is not restricted, with one exception. Provisions must be made in the command structure to enable each device controller individually to use the fast interrupts and to disable all device controllers together from using the fast interrupts.

4-24 Priority Chain

The priority chain, shown in Figure 4-20, determines which device controller may respond to the DMA address. Each device controller must provide this logic for each fast interrupt used (some controllers may not require bi-directional data transfer). Figure 4-21 shows a typical interrupt request flip-flop.

The equations for the priority chain are as follows:

FInPRIn+1 = FInPRIn. NPRICONT

(This equation must be implemented with a Schottky gate with a maximum propagation delay of 7.5 nanoseconds.)

Set PRJCONT = FIn PRIn. NFIn. SET. PHI

Reset PRICONT = FInPRIn. IOMME(DMA). READ. PHI + FIDISABLE

where SET = output of the interrupt request flip-flop if that flip-flop is not clocked by PHI, or the J input signal to the interrupt request flip-flop if the flip-flop is clocked by PHI.

FInPRIn+1 = the priority string output signal

FInPRIn = the priority string input signal

IOMME(DMA) = the decoded DMA channel address anded with IOM

READ = Read command, x'2', decoded from the IO Control field. Use Write command for Fast Interrupt 1 and Read command for Fast Interrupt 2.

FIDISABLE = Fast Interrupt Disable flip-flop, set by IO Control = x'4' or x'C', Disable Fast Interrupts

4-25 Memory Address to B-Bus

The B-Bus Control logic, shown in Figure 4-22, gates the output of the scratch pad/main memory address register to the B-Bus when IO Control = x'1' or '2'. The memory address is needed by the SCU whenever a transfer-in or a transfer-out is made.

The equation for the memory address transfer term is as follows:

BXMA = FI1PRIn. IOMME(DMA). WRITE + FI2PRIn. IOMME(DMA). READ

Schottky gates should be used where indicated on the diagram in order to guarantee that the memory address is stable on the B-Bus within 100 nanoseconds after PHI, as described in subsection 4-6.

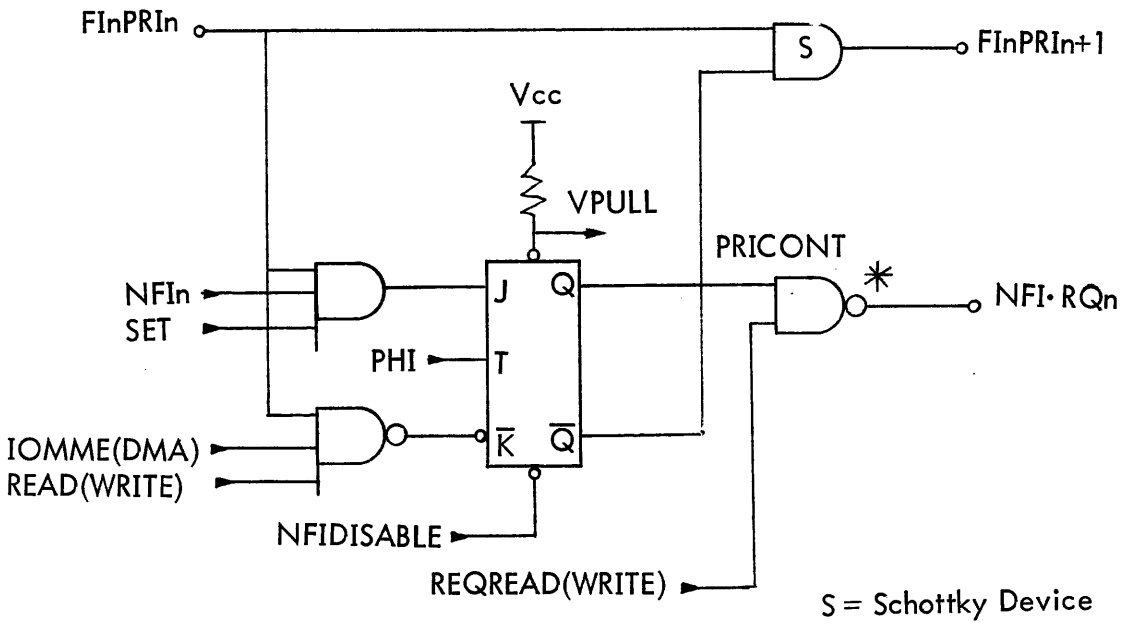


Figure 4-20. Priority Control Logic (Mandatory)

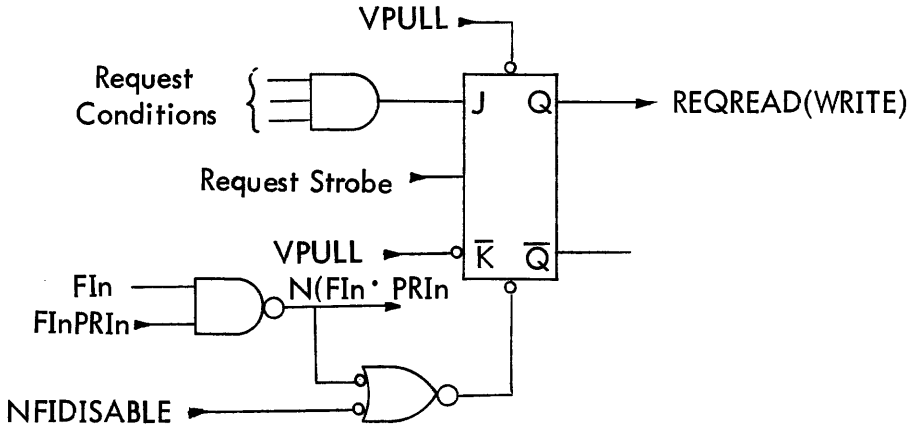


Figure 4-21. Request Control Logic (Typical)

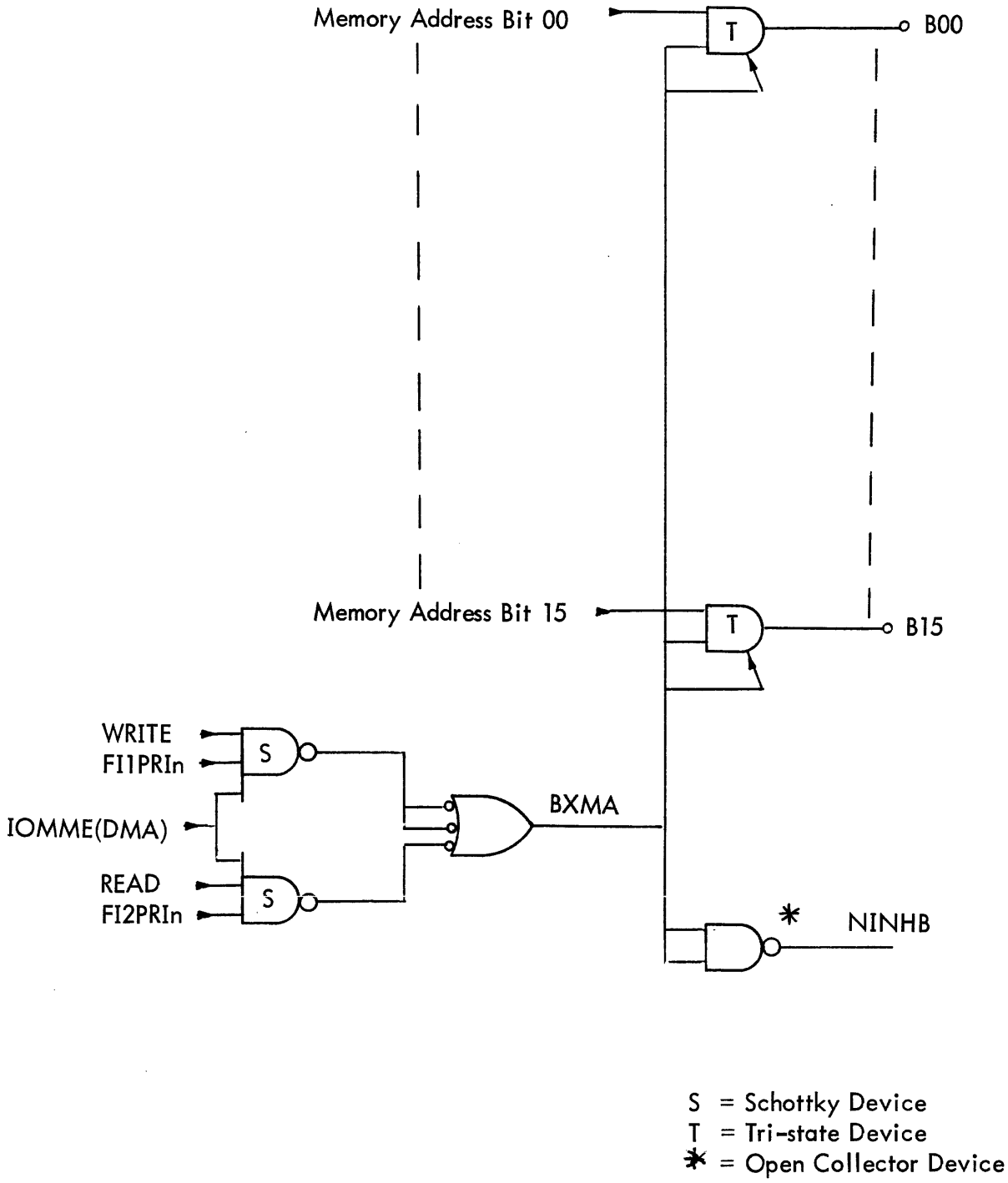


Figure 4-22. B-Bus Control Logic

The same terms are required as the input of an open collector TTL gate, Xerox TN510 or equivalent, to drive NINHB. This signal inhibits the SCU from placing other data on the B-Bus. Also, NINHB must be stable within 80 nanoseconds after PHI.

Figure 4-23 is a timing diagram showing the various interface signals when two device controllers of the DMA type request interrupts simultaneously. The timing assumes the interrupt request (REQREAD) is implemented as depicted in Figure 4-21.

4-26 MECHANICAL AND CABLING REQUIREMENTS

The SCU is made up of a group of TT-size modules housed in a double-high T-series chassis. Within this chassis module locations are provided for several I/O modules which interface directly to the SCU tri-bus structure. An optional I/O Expansion Chassis provides additional I/O module locations, which also interface directly to the SCU tri-bus structure. Signal assignments at all I/O module locations are identical, except for priority string assignment; consequently, any I/O module can be plugged into any I/O module location.

Figure 4-24 shows the outline and dimensions of a TT logic module. Figure 4-25 shows another TT module version with front-edge fingers for connection of ET11 cable connectors. As many as 56 I/O signals may be taken off the module through these connectors, using twisted pair, coaxial, or individually shielded multiple conductor cable. Either module type may be used for I/O module design.

When the module land area requires the use of more than one module per I/O controller, two modules may be sandwiched together with interconnecting pins and receptacles, Xerox 227130 and 201818 respectively, as shown on Xerox Specification, Module Assembly, 201050. Stiffeners must be used. Module marking requirements are contained in the foregoing specification. Environmental requirements are found in Xerox Specification, Standard Environmental Class B Equipment, 185631.

Sandwich or "piggyback" module assemblies can be accommodated only in certain prearranged I/O module locations in the SCU and the I/O Expansion Chassis.

4-27 STANDARD INPUT/OUTPUT MODULE

The Standard Input/Output Module (SIOM) provides a general interface between the SCU and external devices that can be used for many applications.

The backplane of the basic SCU chassis is wired to accept up to six standard or special input/output modules. Additional I/O locations are available in the I/O Expansion Chassis. The backplane interface signals are listed and defined in Section 4-7.

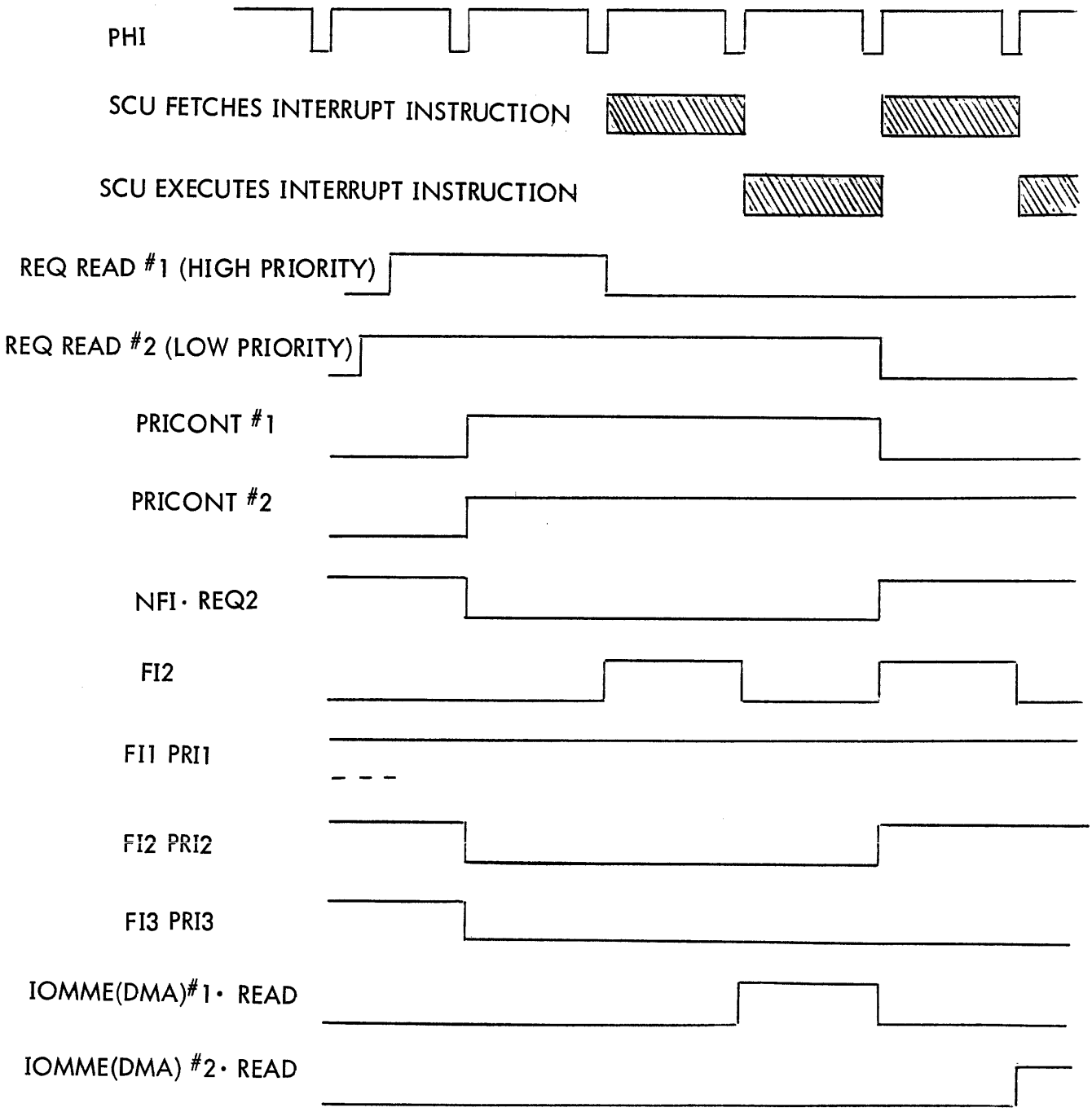


Figure 4-23. Timing Diagram, Fast Interrupt Priority Control

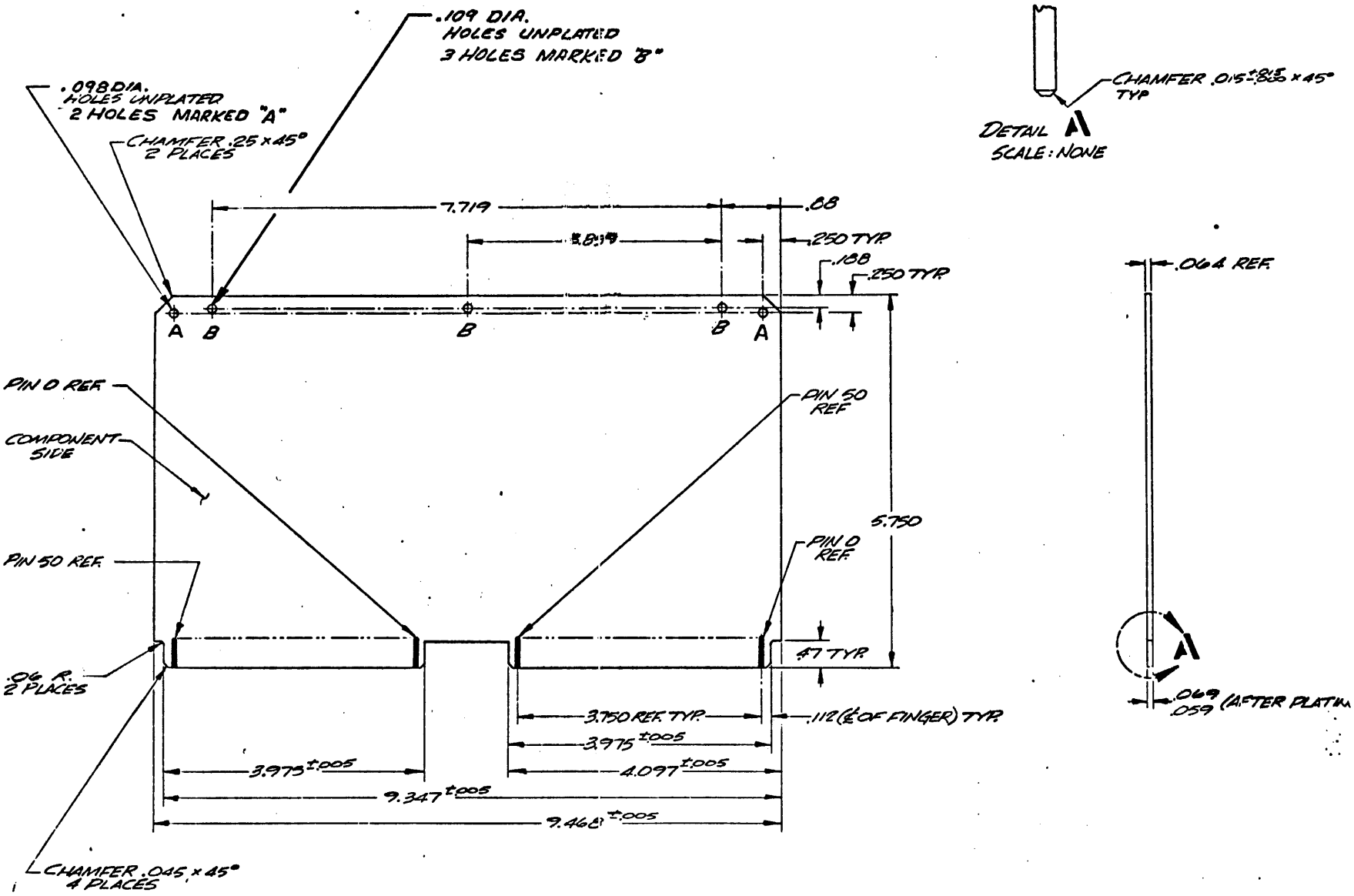


Figure 4-24. TT Logic Module

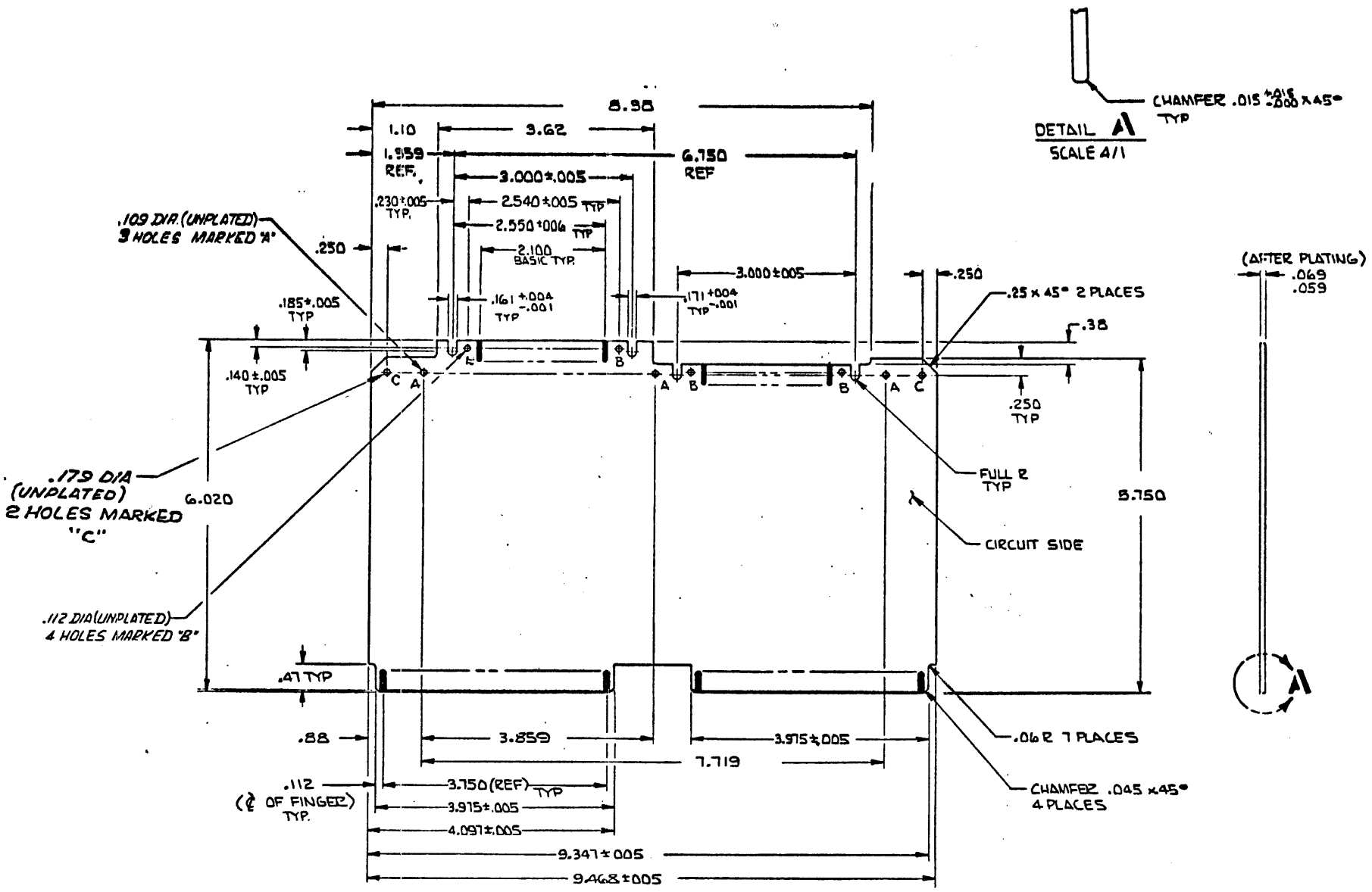


Figure 4-25. TT Logic Module with I/O Fingers

The SIOM provides two general-purpose 16-bit bidirectional input/output buses with buffered output. The SIOM responds to two 7-bit addresses and places the corresponding interrupt addresses on the A-Bus. Up to 128 bidirectional input/output buses, encompassing 2048 bits of input/output, are available with each SCU.

Connection from the front edge of the SIOM to the user's equipment is accomplished by means of ET11 connectors. One ET11 connector is required for each eight bits of IO bus, four connectors per module.

The SIOM is an example of an input/output module designed to the requirements of Section 4-2.

4-28 EXTERNAL INTERFACE

Table 4-2 lists the pin assignments for the four front-edge cable connectors on each SIOM. The signals include the even and odd 16-bit input/output buses, IO active, four external condition test lines, two multiplexed I/O interrupts, reset, two fast interrupts, and four byte disable signals.

4-29 I/O FUNCTIONS

A single SIOM, shown in Figure 4-26, accepts data from two 16-bit input/output buses, NIOBUSEVnn and NIOBUSODnn, and places it on the A-Bus. The module accepts data from the C-Bus and stores it in an even or odd register. From the registers data can be transferred out to the I/O buses or returned to the A-Bus. These bidirectional, non-function committed input/output buses may be used for all the transfers normally required of an interface: data, address, and control. They may be partitioned among these functions on either a byte (8 bits) or word (16 bits) basis. Data can be controlled in terms of four bytes: a left or right byte in either an even or odd register or bus.

As an example, suppose an interface requires an 8-bit address, a 16-bit data path, and 8 bits of control. Register n, left byte, may be assigned to the address; register n, right byte, to the control; and register n+1, to the data path. Output data is buffered, while input data is read directly from an IO bus to the SCU. Moreover, the module provides four bus-direction signals under micro program control which may be used to control data flow direction on each byte of the two buses.

In addition to these data transfer and storage functions, the SIOM accepts two multiplexed input/output interrupts, one for each word, which may be used for input/output rate control or any-other-device-to-SCU control functions. An IO Active (strobe) pulse is provided which is automatically pulsed by the SIOM whenever the right byte or entire even-numbered bus is either read or written. Two fast interrupts allow very fast data throughput rates to be achieved. System reset may be used for initialization when power turn-on occurs. Four external-condition test lines enable the SCU to check the status of external devices.

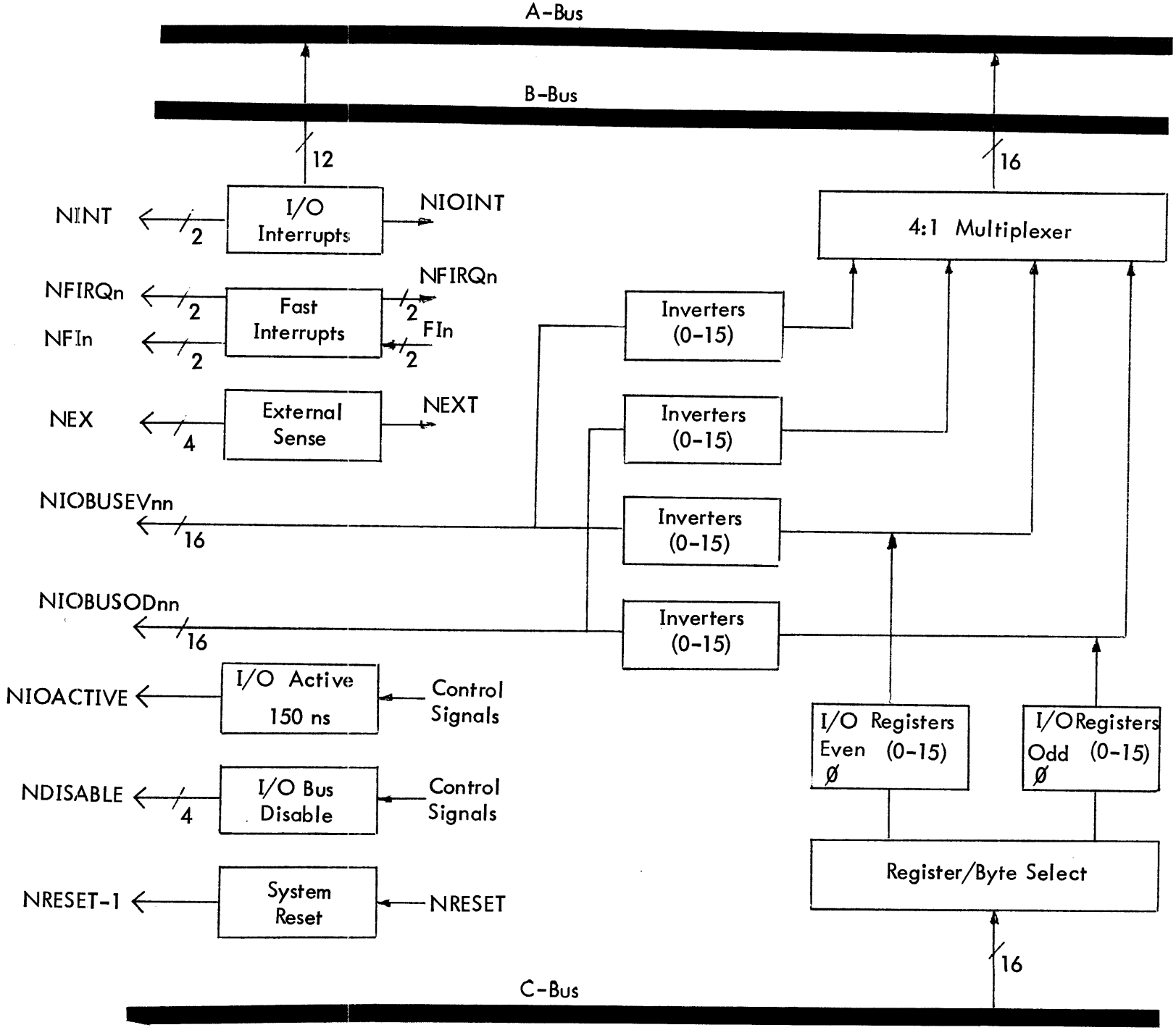


Figure 4-26. Block Diagram, Standard Input/Output Module

Table 4-2. SIOM External Interface

P2		P3		P4		P5	
Component Side		Etch Side		Component Side		Etch Side	
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
NIOBUSEV00	1	NIOBUSOD00	A	NIOBUSEV08	1	NIOBUSOD08	A
01	2	01	B	09	2	09	B
02	3	02	C	10	3	10	C
03	4	03	D	11	4	11	D
04	5	04	E	12	5	12	E
05	6	05	F	13	6	13	F
06	7	06	G	14	7	14	G
NIOBUSEV07	8	NIOBUSOD07	H	NIOBUSEV15	8	NIOBUSOD15	H
SPARE	9	SPAREF1	K	NIOACTIVE	9	SPAREF2	K
NEX(LE)	10	NEX(LO)	L	NEX(RE)	10	NEX(RO)	L
SPARE	11	SPARE	M	NINT(E)	11	NINT(O)	M
NRESET-1	12	NRESET-1	N	NRESET-1	12	NRESET-1	N
NFIRQ-1	13	NFIRQ-2	P	NFI-1 13	13	NFI-2	P
NDISABLE(LE)	14	NDISABLE(LO)	R	NDISABLE(RE)	14	NDISABLE(RO)	R

Input loading is one unit load (TTL = 1.6 ma)

Output drive capability is 12.5 unit loads, except IO Bus which is 29 unit loads less loads required for pullup resistors.

4-30 IO CONTROL

Seven fields of the micro instruction participate in the control of functions on the SIOM. Six of these fields have been characterized, for input/output purposes, in Section 4-6.and defined in full in Section 3. The seventh field, IO Control, is defined below, as used on the SIOM.

When IO Mode = 1 and C Bus/Register Control is not x'B', bits 24-27 become the Input/Output Control field.

Bits 24	Bits 25	IO Control Function
0	0	Read the IO register addressed by the RA Select and Device Select fields to the A-Bus.
0	1	Read the IO register addressed by the RA Select and Device Select fields to the A-Bus and write the data from the C-Bus to the IO register addressed by the RA Select and Device Select fields.
1	0	Read the IO bus addressed by the RA Select and Device Select fields to the A-Bus.
1	1	Read the IO bus addressed by the RA Select and Device Select fields to the A-Bus and write the data from the C-Bus to the IO register addressed by the RA Select and Device Select fields.

Bits 26	Bits 27	Special Control Function
0	0	No change.
0	1	Disable output to the IO bus. (Inhibits output from IO register byte addressed by RA Select, Device Select and A-Bus Control fields to corresponding section of IO bus. This output remains disabled until Enable IO Bus occurs.)
1	0	Enable output to IO bus. (Permits content of IO register byte addressed by RA Select, Device Select and A-Bus Control fields to be transferred to corresponding section of IO bus. This output remains enabled until Disable IO Bus occurs.)
1	1	Not assigned.

4-31 IO BUS

The IO Bus is a zero-volt true, TTL-compatible, bidirectional bus, diagrammed for a single line in Figure 4-27. Data to be output by the SCU to the bus is transferred from the C-Bus to a buffer register, using either the "read register/write register" micro opcode or the "read bus/write register" micro opcode. If the corresponding IO byte is enabled for output, the data is gated to the IO bus. Data may be read to the A-Bus from either the output register or the IO bus, the latter allowing the SCU to perform turnaround testing of the SIOM.

4-32 IO BUS DISABLE

Each byte of the two IO buses can be individually enabled for data output to the bus under micro program control, as shown in Figure 4-28. The byte to be enabled is addressed by RA Select, Device Select, and A-Bus Control. The Q output of the Enable flip-flop gates the output register to the IO Bus. It is also provided to the user from an open collector driver as a bus direction control signal, NDISABLE.

When NDISABLE is false, zero volts, output to the IO bus from the IO register of the selected byte is disabled and incoming data can be placed on this section of the IO bus. When NDISABLE is high, output of the selected byte to the IO bus is enabled and this section of the bus should not be used for input.

4-33 IO ACTIVE

Strobe pulses which are required for an I/O interface are normally generated by the micro program, using a pre-assigned IO bus bit as the strobe signal. This usage minimizes special hardware on the IO module, using micro opcode instead.

There are, however, certain instances where this technique is either inconvenient or slow. For these cases, a single IO Active signal is provided on each SIOM. This signal is keyed to one of the four IO bytes on the module (the least significant byte of the even-numbered register).

IO Active is generated unconditionally by two values of the IO Control field:

10 Read IO Bus to A-Bus

11 Read IO Bus to A-Bus and Write IO Register from C-Bus

IO Active can be generated conditionally by the following value of the IO Control field:

01 Read IO Register to A-Bus and Write IO Register from C-Bus. The condition requires that the byte be enabled for output at the end of execution of the micro instruction; that is, either enabled by the current micro instruction, or enabled by a preceding micro instruction and not disabled by the current one.

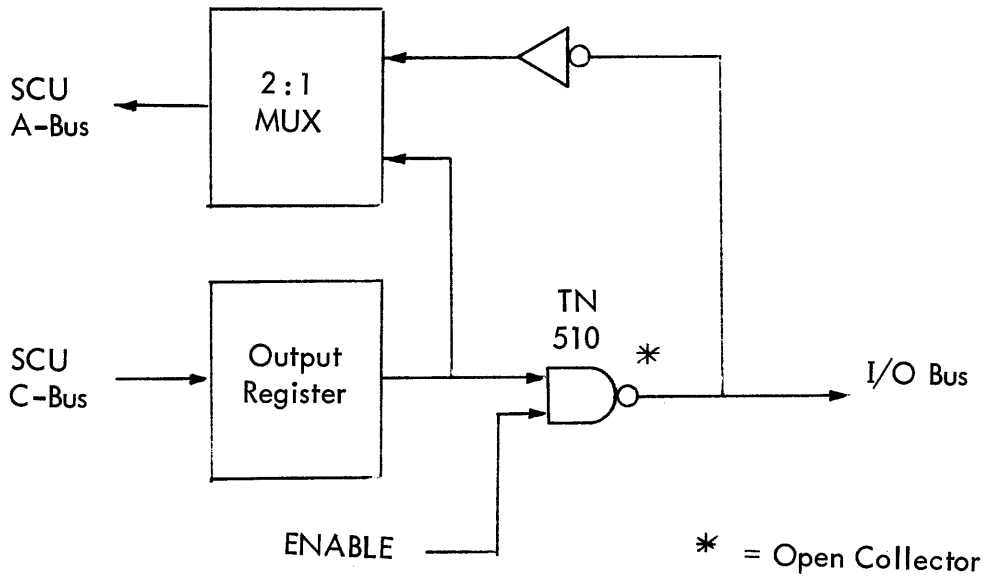


Figure 4-27. Block Diagram, Input/Output Bus

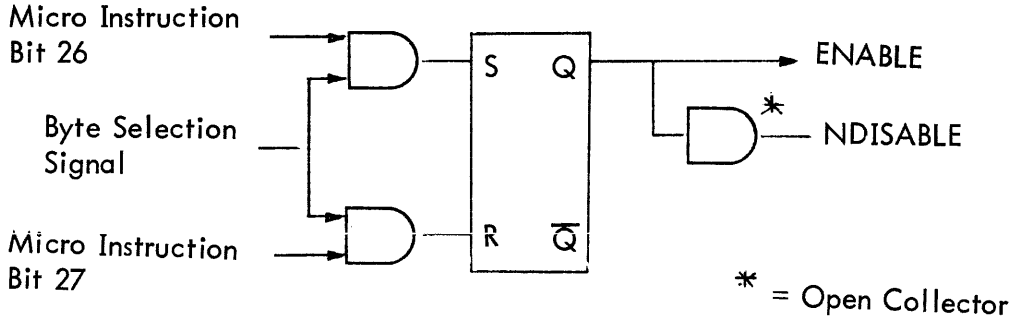


Figure 4-28. Block Diagram, Enable Control

Figure 4-29 shows the timing relationship between the instruction execution time, output data, and the IO Active strobe as measured at the front-edge connector. The IO Active strobe nominally occurs 100 nanoseconds after the micro instruction execution clock time, allowing 60 nanoseconds from data stable to the leading edge of the strobe. The pulse is nominally 150 nanoseconds in width.

4-34 MULTIPLEXED IO INTERRUPTS

The SCU has a single interrupt level associated with general input/output. This interrupt level is multiplexed among the IO locations and on the SIOM is submultiplexed between two interrupt inputs. See Figure 4-30.

Receipt of an active-low interrupt, NINT(E) or NINT(O), from the user sets the Request flip-flop on the next system clock, providing the SCU signal, NINTLOCK, is high. Request high permits three actions:

1. Causes NIOINT to be pulled active-low, if the particular SIOM has priority (PRIn = 1), initiating the multiplexed IO interrupt routine in the SCU.

2. Inhibits PRIn+1, the priority signal for the next IO module in the priority sequence.

3. Enables transfer of the SIOM's device address to the A-Bus on receipt of AXDASTROBE from the SCU.

When the interrupt service routine begins, the SCU hardware automatically block disables the multiplexed IO interrupts in order to avoid interference from other I/O interrupts. The I/O interrupts are re-enabled following the service routine.

The user should generate I/O interrupt signals on an inquiry/response basis, as indicated in Figure 4-31. That is, an interrupt signal should be driven active-low and remain in this state until the micro program responds. This response may take the form of a pulse on a line of the IO bus assigned for this purpose. The micro program should not generate this response until it has determined which user device generated the interrupt.

For more complete information on handling multiplex IO interrupts, refer to subsection 4-13.

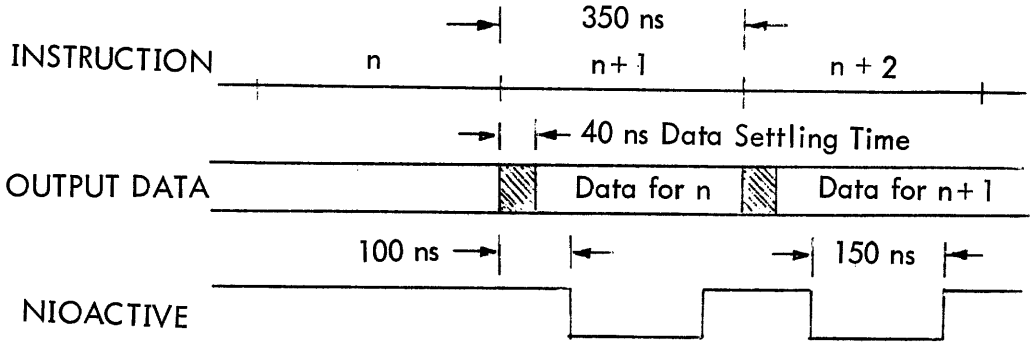


Figure 4-29. Timing Diagram, Example of IOACTIVE Signal

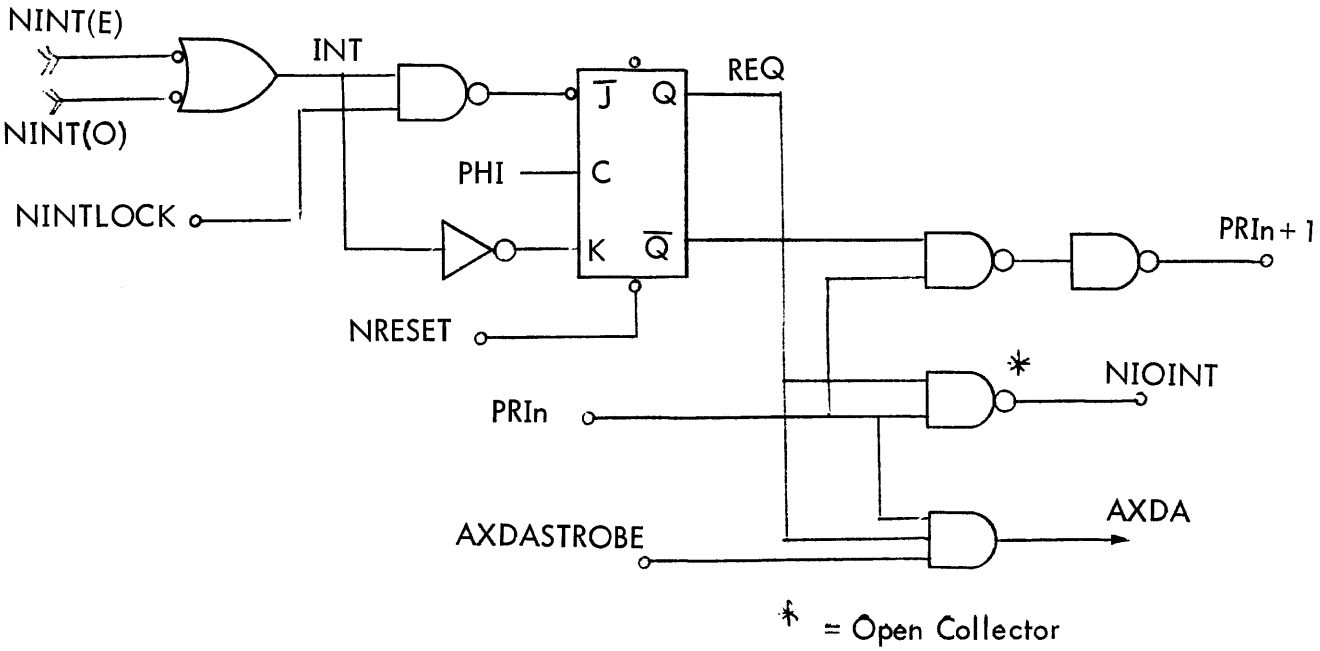


Figure 4-30. Multiplexed I/O Interrupt

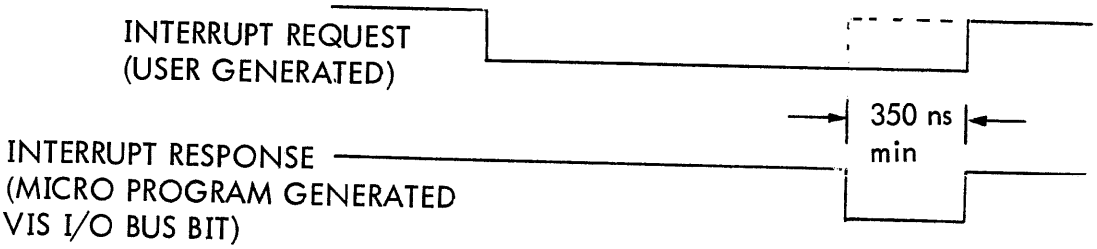


Figure 4-31. Timing Diagram, Multiplexed I/O Interrupt

4-35 EXTERNAL CONDITION TEST

Each SIOM accepts four external condition lines at the connectors listed in Table 4-3, loads the state of each line into a parallel register at each system clock, and multiplexes the one selected by micro opcodes to a single line, NEXT. This active-low signal is applied to the SCU, as described in subsection 4-17.

With IO Mode true, the RA Select, Device Select, and IO Control fields address the external condition lines as follows:

1. The most significant six bits of RA Select and Device Select address the SIOM that is to present its external condition to the NEXT line.

2. The least significant bit of RA Select and Device Select addresses the even or odd pair of external condition lines on the selected SIOM.

3. IO Control selects the first or second external condition line of each even or odd pair as follows:

Bits 24 - 27	
0 0 0 0	Place Line 1 on NEXT line
0 0 0 1	Place Line 2 on NEXT line

The external condition line addressed is tested by means of Branch Control = X'4', Test/Branch External Condition Set, or X'5', Test/Branch External Condition Reset. The operations resulting from these two micro opcodes are set forth in detail in Section 3. Briefly, if the test is successful, the least significant 12 bits of the C-Bus are loaded into the micro address register to serve as the next address. If not, the MAR increments.

To implement this test, a means of getting the desired micro address to the C-Bus is needed. One means is the use of a skip micro instruction. For example, Skip On External Test Successful can be accomplished by means of the following micro opcodes:

IOM	RA Sel	ALU	RBSel	ABus	BBus	CB/RC	BrCon	ALUSel	IOCon	DevSel
1	n n n	0	0	3	3	0	4 or 5	9	0 or 1	n n n n

A Bus Control = 3, transfers contents of the micro address register to A-Bus. It is important to note that the micro address transferred is the sequential address beyond the one corresponding to the skip micro instruction. That is, if the skip micro instruction is called n, the next micro address is n + 1.

RB Sel = 0 (This field defines Bit Position, when IOM = 1 and B Bus Control = 3.) Places a one in the least significant bit of the B-Bus, B15.

ALU Mode = 0

ALU Select = X'9' The hardware forces Cn to zero and the ALU performs A plus B. Since the value on the A-Bus is equivalent to n + 1 and the value on the B Bus is 1, the sum is n + 2. This sum is placed on the C-Bus.

If the test is successful, the C-Bus n + 2 value is loaded into the micro address register at the clock following the skip micro opcode. All zeros are loaded into the micro control register at this same time. The micro instruction for micro address n + 2 is transferred into the micro control register on the second clock. In effect, the micro program skipped one micro instruction and used the second micro instruction.

On the other hand, if the test is unsuccessful, micro address n + 1 addresses control memory via the memory address multiplexer, transferring micro instruction n + 1 to the micro control register at the clock following the skip micro opcode. The micro address register increments to n + 2 at the same clock. (Micro instruction n + 1 should be devised to avoid micro instruction n + 2.)

Note that the next micro address, if the test is successful, cannot be taken from the Immediate Address field or the Emit field, since IO Control and Device Select preempt those fields.

A(0-11) <= MAR(0-11);

B15 <= 1;

C(0-15) <= A(0-15) plus B(0-15)

If TEST = 1, then MAR(0-11) <= C(4-15). PHIn; MCR(0-31) <= 0. PHIn;
MAM(0-1) <= MAR(0-11);
MCR(0-31) <= CM(0-31) (MAM0-11). PHIn + 1
If TEST = 0, then

MAM(0-11) <= MAR(0-11);

MCR (0-31) <= CM(0-31) (MAM0-11). PHIn;

MAR plus 1 at PHIn.

4-36 FAST INTERRUPTS

The SCU accommodates two fast interrupts for high-speed data transfers, as detailed in Section 4-18. Each SIOM provides front-edge connectors to which to connect the Fast Interrupt Requests,

NFIRQn, and responses, NFIn, as listed in Table 4-2. The Fast Interrupt Requests are wired through the SIOM unchanged. The Fast Interrupt responses from the SCU are placed on the front-edge connectors from open-collector inverters on the SIOM. Both fast interrupts are made available through any SIOM. Both NFIRQ1 and NFIRQ2 have higher priority than the multiplexed I/O interrupts.

The fast interrupts, as they are made available through the SIOM, must be dedicated to a single high-speed I/O device. Each fast interrupt cycle-steals a single cycle from the SCU to execute a single micro instruction which typically transfers a word or byte in or out between an IO bus and scratch pad/main memory and increments a general register which contains the memory address.

Figures 4-32 and 4-34 show the fast-interrupt interface timing for a single data transfer or for multiple transfers at slower throughput rates. Times shown on the figures are measured at the I/O connector of the SIOM. Figures 4-33 and 4-35 show timing at maximum throughput rates. Note that for this mode of operation, the interrupt request generated by the I/O device, NFIRQn, may be held low for the entire duration of a block transfer. It must be raised within 180 nanoseconds after the falling edge of the response pulse, FIn, for the last word to be transferred in order to assure that an extra interrupt is not triggered.

4-37 SYSTEM RESET

The system reset signal, NRESET, is driven low when either the SCU power supply goes on (whether because of manual action or following a power failure) or the Clear switch on the maintenance control panel is pressed. This signal may be used to reset the I/O device interface to a quiescent condition.

4-38 TELETYPE, PAPER-TAPE, RTC INTERFACE

The Teletype, Paper-Tape, Real-Time Clock module interfaces to and controls a teletype keyboard/printer and/or paper-tape reader/punch, a high-speed paper-tape reader, and a high-speed paper-tape perforator, and provides a real-time clock.

This module can be installed in I/O location X'00' or X'01' of the SCU. Each device is controlled independently. The devices can be operated simultaneously with or without using interrupts.

The teletype controller interfaces to models ASR33, ASR35, KSR33, and KSR35, which have speeds of 10 characters per second. The micro program can cause the teletype to type out characters, and can read in characters typed on the keyboard. With the ASR model, the micro program can cause the unit to punch characters on a paper tape and can read in characters from a punched tape.

The high-speed paper-tape controller interfaces to a 300 characters/second high-speed paper-tape reader and a 75 characters/second high-speed paper-tape perforator.

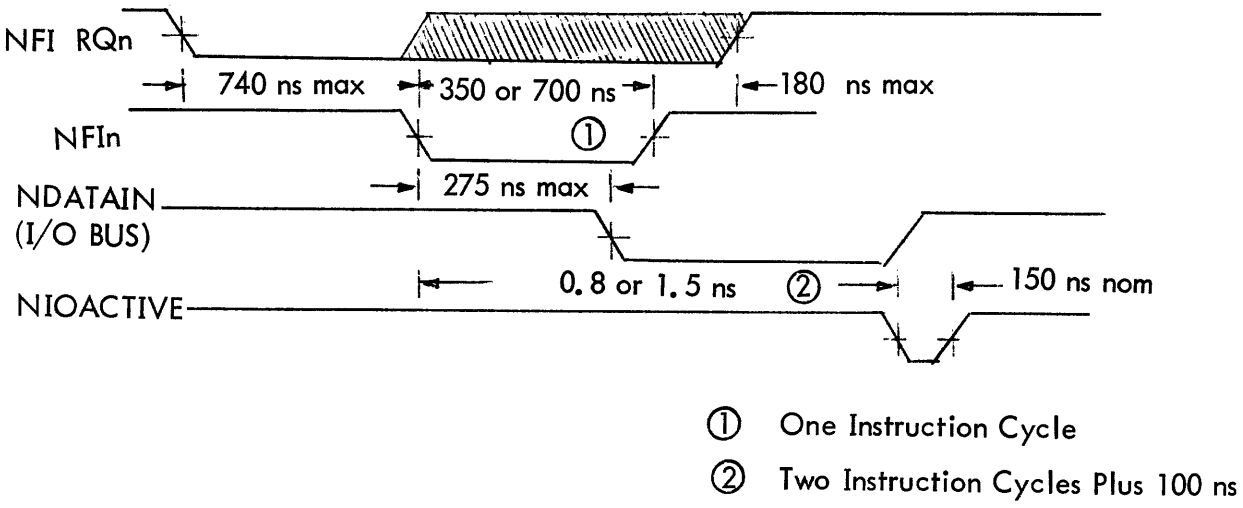


Figure 4-32. Fast Interrupt Timing for Single Input Data Transfer

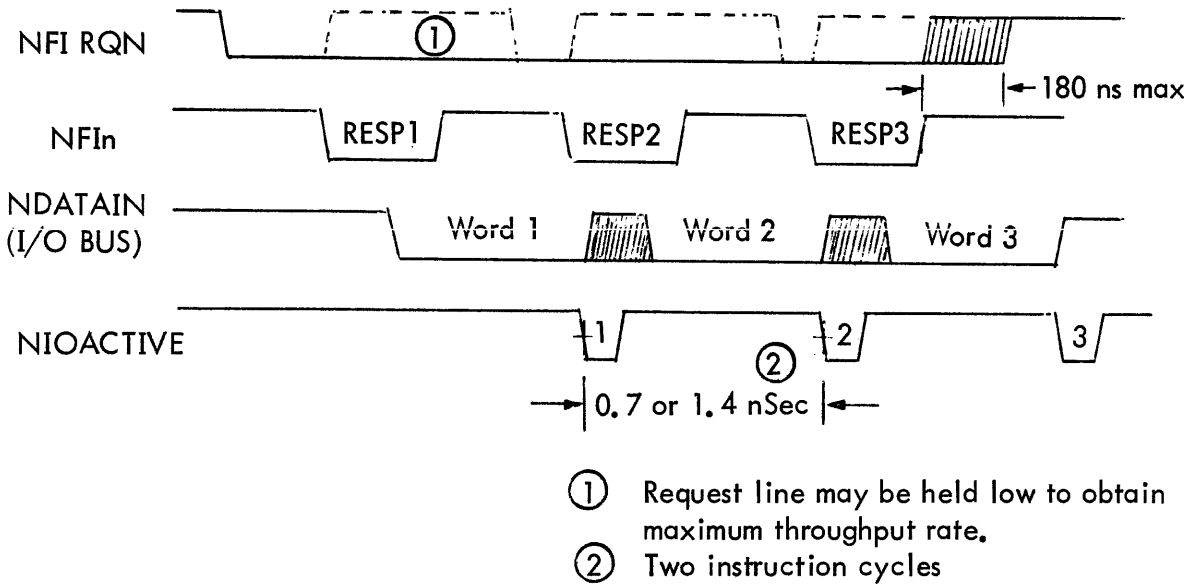


Figure 4-33. Fast Interrupt Timing for Multiple Input Data Transfer

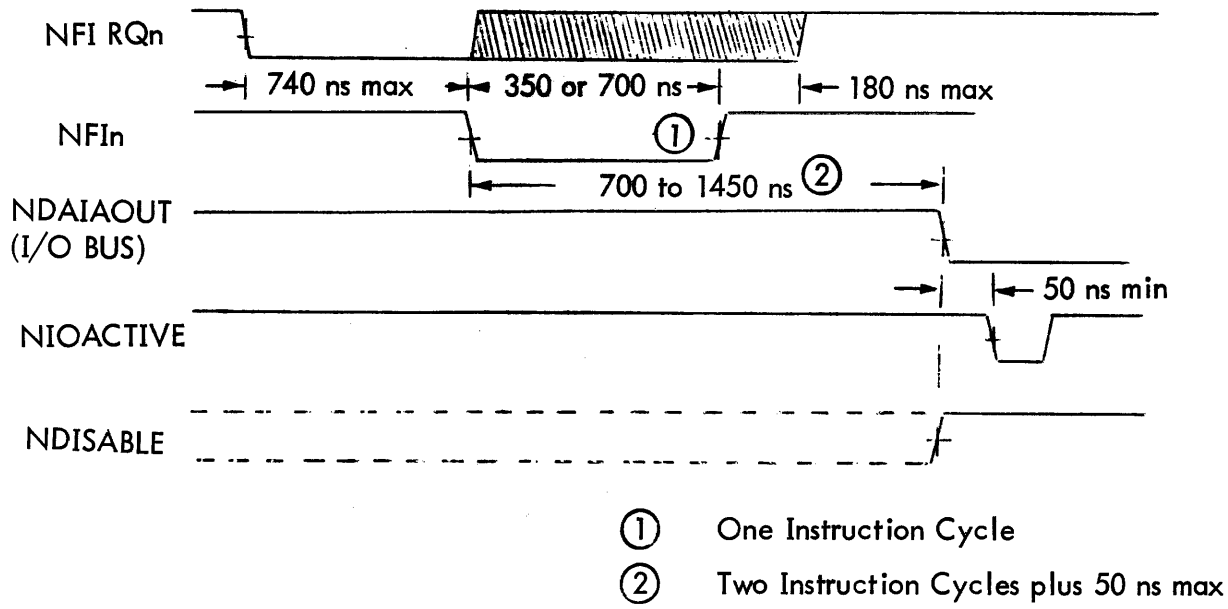


Figure 4-34. Fast Interrupt Timing for Single Output Data Transfer

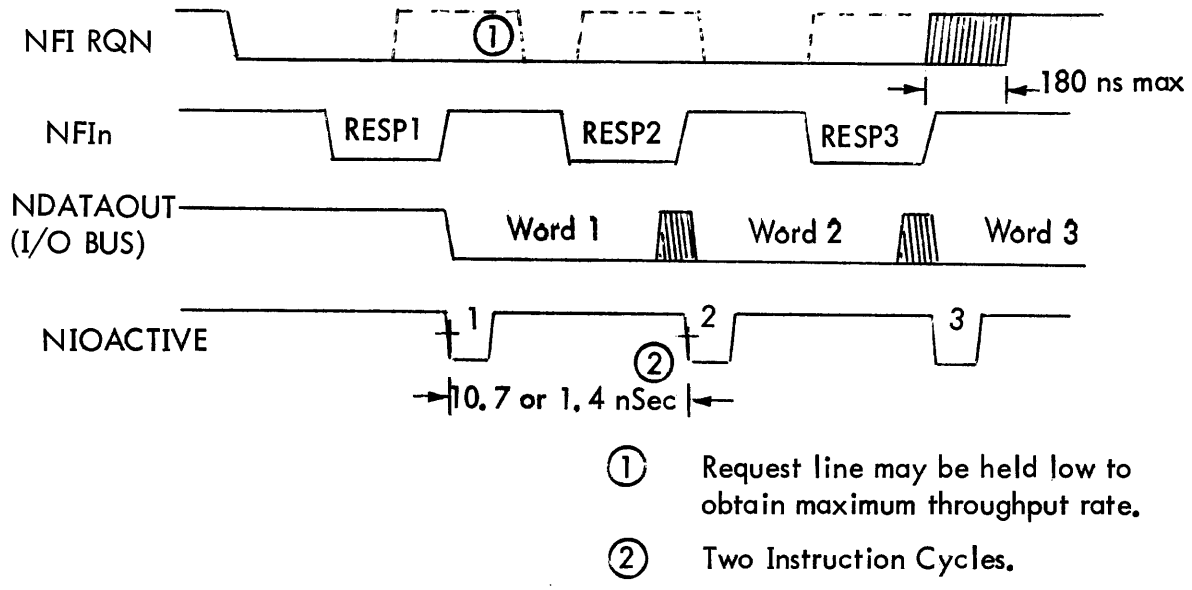


Figure 4-35. Fast Interrupt Timing for Multiple Output Data Transfer

4-39 DATA FLOW

This interface module places inputs from the devices on the A-Bus and takes outputs for the devices from the C-Bus, as shown in Figure 4-36.

Teletype In. Serial data on one line from the teletype keyboard or paper-tape reader is converted to parallel and multiplexed to the least significant byte of the A-bus. A status byte containing information on the current activity and condition of the interface module and the teletype, generated on the interface module, can also be multiplexed to the A-Bus.

Teletype Out. The least significant eight bits of the C-Bus are converted to serial form and transmitted on one line to the teletype printer or paper tape punch. A control byte used to operate the teletype interface and teletype can be transferred from the least significant byte of the C-Bus to the teletype control logic.

High-Speed Paper-Tape Reader. Eight bits of parallel data from the high-speed paper tape reader is multiplexed to the least signigicant byte of the A-Bus. Alternately, the high-speed paper tape status byte, originating on the interface module, can be multiplexed to the least significant byte of the A-Bus.

High-Speed Paper Tape Punch. The least significant byte of the C-Bus is transmitted in parallel form via the high-speed paper tape punch output buffer to the high-speed paper tape punch. A control byte, used to operate the interface logic and the high-speed paper tape reader/punch can be transferred from the least significant byte of the C-Bus to the interface logic. In addition, the data in the output buffer can be multiplexed to the A-Bus.

Real-Time Clock. The real-time clock is a counter stepped by a power line frequency (50 or 60-Hz) signal from the SCU power supply. Any binary number from 0 to 255 can be preset into the counter from the C-Bus. This action also starts counting from the preset number up. The binary number in the counter can be multiplexed to the A-Bus. This action stops the counter. When the counter reaches the last count, 255, it generates a multiplexed IO interrupt. The counter continues to count after generating this interrupt until its counter is read to the A-Bus.

4-40 SELECTION

The IO module is addressed by the RA Select (RSA01-03) and Device Select (EMIT28-31) fields when IOM is true. The specific device on the IO module, such as the teletypewriter, high-speed paper tape punch/reader, or real-time clock, is selected by the IO Control field (EMIT24-27), defined in subsection 4-43.

Two I/O addresses are available for this module: x'00' or x'01'. The NEMIT31 line on the module is normally jumpered to provide address x'00', but this jumper can be removed and jumpers added to invert NEMIT31, providing address x'01'.

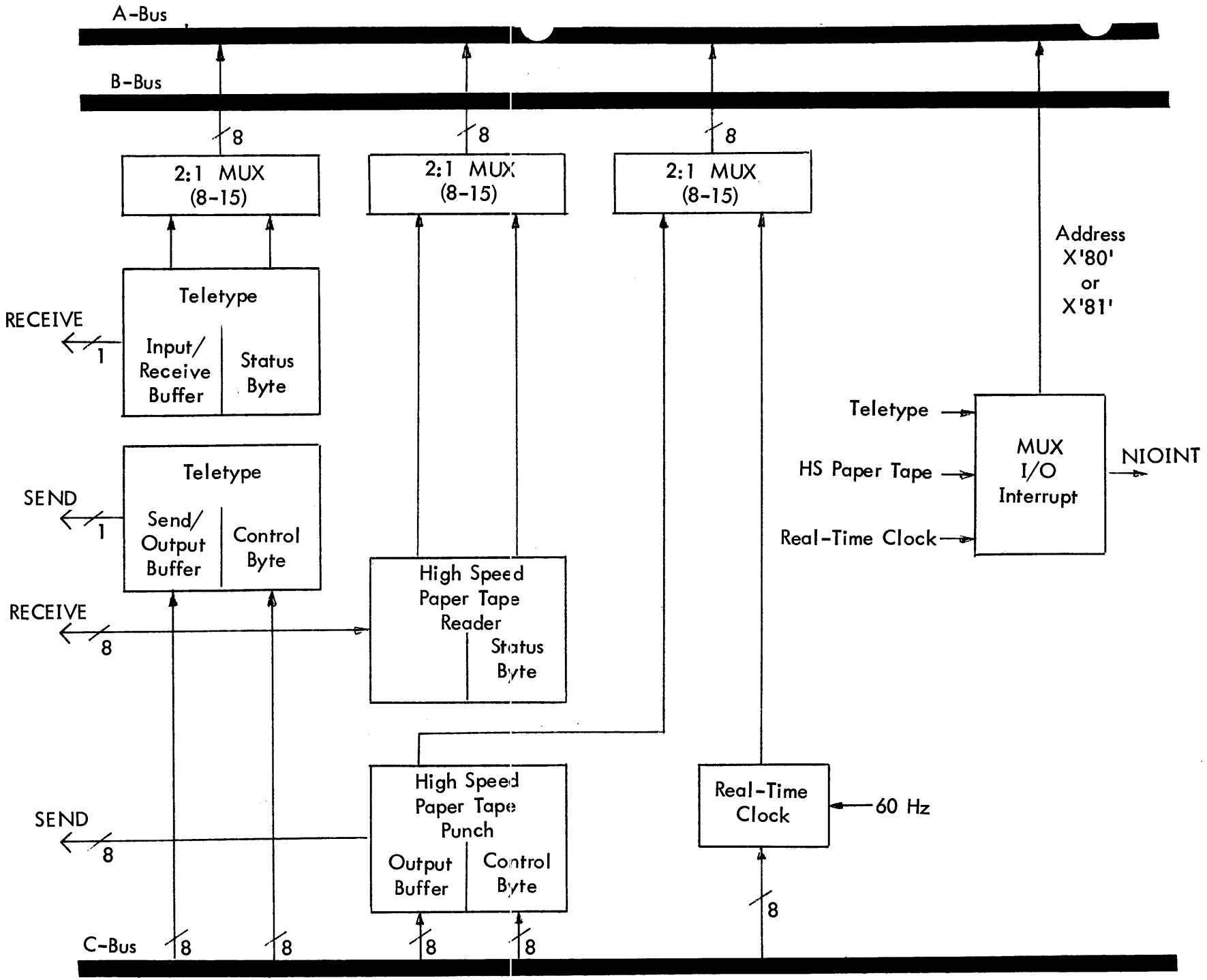


Figure 4-36. Block Diagram, Teletype, High-Speed Paper Tape, Real-Time Clock Module

4-41 I/O INTERRUPTS

Each device on the module is capable of generating an input to the multiplexed I/O Interrupt under the following circumstances:

a) Teletype: when the send buffer is empty or the receive buffer is full, providing the send or receive interrupt is armed.

b) High-Speed Paper Tape: when the punch output buffer is empty or the reader input is ready, providing the punch or reader interrupt is armed.

c) Real-Time Clock: when the last count is reached.

Each of these conditions sets a flip-flop on the I/O module which, in turn, gives rise to the active-low open-collector signal, NIOINT. Thus, whenever one or more devices are requesting service, NIOINT transmits this fact to the SCU. The SCU, however, can accept NIOINT only when the SCU I/O Interrupt is enabled.

NIOINT causes the micro program to jump to control memory location x'008', as detailed in Section 4-13. The micro program enables the transfer of the module's interrupt address to the A-Bus, as explained in Section 4-14. The module's interrupt address is either x'80' or x'81', corresponding to the module address x'00' or x'01' – that is, for interrupt purposes, bit 8 of the A-Bus is forced to 1. This address is placed on the right byte of the A-Bus, providing the module currently has the highest priority and the interrupt micro instruction in control memory location x'008' contains a Push function, C-Bus/Register Control = x'B' (Immediate Address), and IOM = 1. The address can also be read to the A-Bus by MCR = x'E'.

The interrupt request NIOINT remains low until the condition which caused the interrupt is reset. The condition must be cleared before re-enabling the multiplexed I/O interrupt in order to prevent the initiation of another interrupt. The multiplexed I/O interrupt is re-enabled by executing a Pull to exit the interrupt handler routine. In responding to the interrupt, the SCU reads the teletype and paper tape status bytes to determine which of the devices caused the interrupt.

4-42 Interface Connections

Interface connections for the devices are listed in Table 4-3. The interface signals are defined in the following text.

Table 4-3. Device Interface Connections

Module		Teletype	
Signal	Pin	Signal	Pin
Send (to Teletype) (TTYRTN1)	P2-1		P2-8
Receive (from Teletype) (TTYIN)	P2-2		P2-6
Common	P2-3		P2-4 P2-5, 7
Reader On +	P2-4		P2-14
Reader On -	P2-5		P2-15
		Reader	
RDATA1	P4-8	CH1	J1-D
RDATA2	P4-9	CH2	J1-E
RDATA3	P4-5	CH3	J1-F
RDATA4	P4-11	CH4	J1-H
RDATA5	P4-4	CH5	J1-J
RDATA6	P4-7	CH6	J1-K
RDATA7	P4-6	CH7	J1-L
RDATA8	P4-10	CH8	J1-M
SPKTB	P4-2	SPKTB	J1-N
RFWD	P4-1	FWD	J1-18
RSTOP	P4-3	STOP	J1-20
		Punch	
Data 1	P5-K		J1-1
Data 2	P5-N		J1-2
Data 3	P5-H		J1-3
Data 4	P5-G		J1-4
Data 5	P5-P		J1-5

Table 4-3. Device Interface Connections (Continued)

Module		Teletype	
Signal	Pin	Signal	Pin
		Punch	
Data 6	P5-L		J1-6
Data 7	P5-M		J1-7
Data 8	P5-R		J1-8
PCMD	P5-D		J1-11
PRDY	P5-E		J1-12
NSYSRDY	P5-F		J1-13

Send: Serial, 11-unit code teletype signal consisting of start mark, eight character marks or spaces, and two stop marks.

Receive: Same type of signal as Send.

Reader On: Causes teletype paper tape to step.

Common: Common for Send and Receive.

RDATAn: Reader Data output lines, at +5v for a one, or hole condition, and 0v for a zero, or no hole condition.

SPKTB: Sprocket Channel. A pulse of approximately 60 percent of data channel width.

RFWD: Reader Forward moves paper tape forward.

RSTOP: Reader Stop stops tape motion.

DATAn: Punch data output lines, at +3v to +5v for a one, or hole condition, punched when the Punch Command is given, or 0v for a zero, or no hole condition.

PCMD: Punch Command. Causes tape to advance one position and to punch hole if data is true.

PRDY: Punch Ready. Indicates the tape unit is ready to accept a Punch Command.

NSYSRDY: System Not Ready. Indicates that power is off or internal voltages have not stabilized. False state indicates that internal voltages have stabilized after power is turned on.

The interface signals are automatically generated or received and processed in response to micro program instructions, defined in subsection 4-43.

4-43 MICRO PROGRAM CONTROL

The Teletype, Paper Tape, and Real-Time Clock interface module is controlled primarily by the IO and Special Control field , as summarized in Table 4-4. In addition, IO Mode must be true, and RA Select and Device Select used to address the module.

Micro opcodes are assigned to allow bits 24 and 25 to select the device:

24 25	Device
0 0	Teletype
0 1	High-Speed Paper Tape Reader/Punch
1 0	High-Speed Paper Tape Reader/Punch
1 1	Real-Time Clock

Bit 26 indicates data (0) or control/status information (1), and bit 27 specifies read (0) or write (1).

The micro program contral operations and terms are defined in greater detail in the following text.

Bit 24-27 (Hexadecimal)	Function
0	Read Teletype Data. Inputs a byte (character) from receiver buffer register to right byte of A-Bus. The data may be of two types, either ANSCII or binary. Convention requires that bit 8 is "0" for ANSCII data. Removes Receive Buffer Full interrupt.
1	Write Data to Teletype. Transfers right byte of C-Bus to output buffer register. Removes Send Buffer Empty interrupt.
2	Read Teletype Status Byte. Inputs status byte from teletype interface unit to right byte of A-Bus. The teletype status byte conveys information on the current activity and condition of the controller and the teletype. This byte may be used in a sense mode where the micro program continually scans the byte or in an interrupt mode where the micro program responds to an interrupt. Each bit of the teletype status byte has the following significance: A08: Data Overrun. Indicates that a data byte was left in the receive buffer too long and an incoming data byte destroyed it. This condition may be cleared by reading the input buffer. A09: Real-Time Clock Interrupt. Detects real-time clock interrupt. This bit sets when the real-time clock counter overflows, and resets when the contents of the counter is read to the A-Bus.

Table 4-4. Teletype, Paper Tape, Real Time Clock Control

24-27 (Hexadecimal)	Function	Action*
0	Read Teletype Data	TT⇒A-Bus
1	Write Data to Teletype	C-Bus⇒TT
2	Read Teletype Status Byte	TTS⇒A-Bus
3	Write Teletype Control Byte	C-Bus⇒TTC
4	Read HS Paper-Tape Reader	PTR⇒A-Bus
5	Not assigned.	
6	Read HS Reader/Punch Status Byte	PTS⇒A-Bus
7	Write HS Reader/Punch Control Byte	C-Bus⇒PTC
8	Not assigned.	
9	Write Punch Output Data	C-Bus⇒PTP
A	Not assigned.	
B	Not assigned.	
C	Read RTC Counter and Reset	RTC⇒A-Bus
D	Write Output Data to RTC and Start	C-Bus⇒RTC
E	Not assigned.	
F	Not assigned.	

*Definition of Terms

PTC	Paper Tape Control Byte for both reader and punch
PTP	Paper Tape Punch
PTR	Paper Tape Reader
PTS	Paper Tape Status Byte for both reader and punch
PTC	Real Time Clock
TT	Teletype
TTC	Teletype Control Byte
TTS	Teletype Status Byte

A10: Break (Framing Error). A byte without the two stop bits has been received. This condition is interpreted as a long space or break. After Break has been detected, it is necessary to input a byte to clear Break. Break disappears at the next input byte. In responding to Break, do not use control bit, reset All.

A11: Tape Reader Started and Busy. Start Reader-Continuous or Step Reader Once command has been given and binary data should be coming in from the paper-tape reader at the rate of one character per hundred milliseconds. This bit is reset either by a Reset All command or , when the Step Reader Once command is used, by the next character received. The Step Reader Once command allows the next byte to be read before actually stopping.

A12: Receive Interrupt Armed. The arm receive interrupt flip-flop is set. Data coming in from the teletype causes an interrupt when the receive buffer is full and the IO Interrupt is enabled. This interrupt is reset by Read Data, Reset All, or Disarm Receive Interrupt commands. Read Data removes the cause of the interrupt, while the other two disarm the receive interrupt, resetting this bit to zero.

A13: Send Interrupt Armed. The arm send interrupt flip-flop is set. Data going out to the teletype causes an interrupt when the send buffer is empty and the I/O Interrupt is enabled. This interrupt is reset by Write Data, Reset All, or Disarm Send Interrupt commands. Write Data removes the cause of the interrupt, while the other two disarm the send interrupt, resetting this bit to zero.

A14: Receive Buffer Full. A complete byte has come into the input register. This byte must be transferred to the SCU within 100 milliseconds or data overrun occurs. This bit is reset by Read Data or Reset All commands.

A15: Send Buffer Empty. The data byte in the holding register has been inserted into the output register and a new data byte is needed. This bit is set true after a Reset All command. This bit is reset only by outputting a data byte (Write Data to Teletype).

3 Write Teletype Control Byte. Outputs teletype control byte from right byte of C-Bus to teletype interface unit. The teletype control byte controls the operation of the controller and teletype. Each bit of the teletype control byte has the following significance:

C08: Reset All. Resets interrupt flip-flops, data registers, data full, and test mode. Also sets Send Buffer Empty teletype status bit. In essence, this command clears the I/O device. If a byte is being transmitted, this command stops the data transmission.

C09: Test Mode. Selects a test mode for the teletype interface portion of the module. This mode provides a turnaround test capability for the

teletype data path. When the control byte is output with this bit a one, the Test Mode flip-flop sets and switches the receiver input data from the teletype to the transmitter output. The data output may be read back in and compared with the output data using the IO Control functions. The teletype may remain connected to the module with the teletype power on or off. If the power is on, the teletype prints and/or punches the output data. No input data may be accepted from the teletype in the Test Mode. The Test Mode is reset by the Reset All command or by the SCU maintenance control panel Clear switch.

C10: Step Reader Once (Stop Next). Starts a 30-millisecond pulse after which the Start Reader--Continuous flip-flop resets, insuring reading one more byte, otherwise reads one byte only if the unit was not busy. Teletype Status Byte, Bit 11, goes to zero when the next byte is received. This command serves as the stop command for the Start Reader--Continuous command mode.

C11: Start Reader--Continuous. Starts reader reading in continuous mode. mode. Reader continues to read until Step Reader Once (Stop on Next) or Reset All command is issued.

C12: Disarm Receive Interrupt. Disarms the receive interrupt, preventing the I/O interrupt from being generated when the receive buffer is full. Has no effect on Receive Buffer Full teletype status bit.

C13: Disarm Send Interrupt. Disarms the send interrupt, preventing the I/O interrupt from being generated when the send buffer is empty. Has no effect on Send Buffer Empty teletype status bit.

C14: Arm Receive Interrupt. Sets the receive interrupt flip-flop to allow an interrupt to occur any time the receiver buffer is full.

C15: Arm Send Interrupt. Sets the send interrupt flip-flop to allow an interrupt to occur any time the send buffer is empty and the I/O interrupt is enabled.

4 Read HS Paper-Tape Reader. Inputs a byte (character) from the high-speed paper tape reader to the right byte of the A-Bus. To initiate data input from the paper tape reader, a paper-tape/control byte with bit 13, Start Reader, true is issued, by means of IO Control = x'7', Write Reader/Punch Control Byte. Bit 11, Arm Reader Interrupt, may also be true in order to arm the data input interrupt. Then, the interrupt generated for each character by the sprocket channel sets bit 15 of the paper tape status byte, Input Data Ready (Reader). The input data is available for approximately 1.5 milliseconds maximum after the interrupt request. Reading the data resets bit 15 of the paper tape status byte, Input Data Ready (Reader) and the interrupt request. When all of the data has been read, the reader is stopped by output of the paper tape control byte with bit 14, Stop Reader, true. If a single character is to be read at a time, the reader is started and stopped after each character is read. To stop the reader on a character or space after the character, the

stop command must be output within 50 microseconds after the Input Data Ready goes true.

Figure 4-37 is a timing diagram of data input from the reader and Figure 4-38 is a flow diagram of this input.

5 Not assigned.

6 Read HS Reader/Punch Status Byte. Inputs reader/punch status byte to right byte of A Bus. The paper-tape reader/punch status byte conveys information on the current activity and condition of both the reader and punch. This byte may be used in a sense mode where the micro program continually scans the byte or in an interrupt mode where the micro program responds to an interrupt. Each bit of the reader/punch status byte has the following significance:

A08: Not used.

A09: Punch Power On. Indicates that punch power is on and internal voltages have stabilized.

A10: Punch Interrupt Armed. Punch buffer empty interrupt has been armed and interrupts are allowed, providing the multiplexed I/O interrupt is also enabled.

All: Punch Not Ready. Indicates punch power is off or the punch is not ready to accept data or a punch command. This bit is true during the punch cycle for approximately 13 milliseconds after the punch command.

A12: Punch Output Buffer Empty. Output data buffer register is emtpy and may accept another data byte.

A13: Reader Run Control Set. Sets when paper-tape control byte bit 13, Start Reader, is true. Indicates that the reader is set to run and is busy.

A14: Reader Interrupt Armed. Sets when reader interrupt is armed to allow input data interrupts.

A15: Reader Input Data Ready. Sets when data may be input from the reader and resets after data is input.

7 Write HS Reader/Punch Control Byte. Outputs reader/punch control byte from right byte of C Bus to high-speed paper-tape reader/punch interface unit. This control byte controls the operation of the controller and paper tape. Each bit of this control byte has the following significance:

C08: Arm Punch Interrupt. Enables punch buffer empty interrupt, allowing the multiplexed IO interrupt from the punch to be generated.

C09: Disarm Punch Interrupt. Disables punch buffer empty interrupt, preventing the multiplexed IO interrupt from the punch.

C10: Reset Punch Interface Unit. Unconditionally resets all control functions of the punch interface, including buffer empty, punch ready, and disarm punch interrupt.

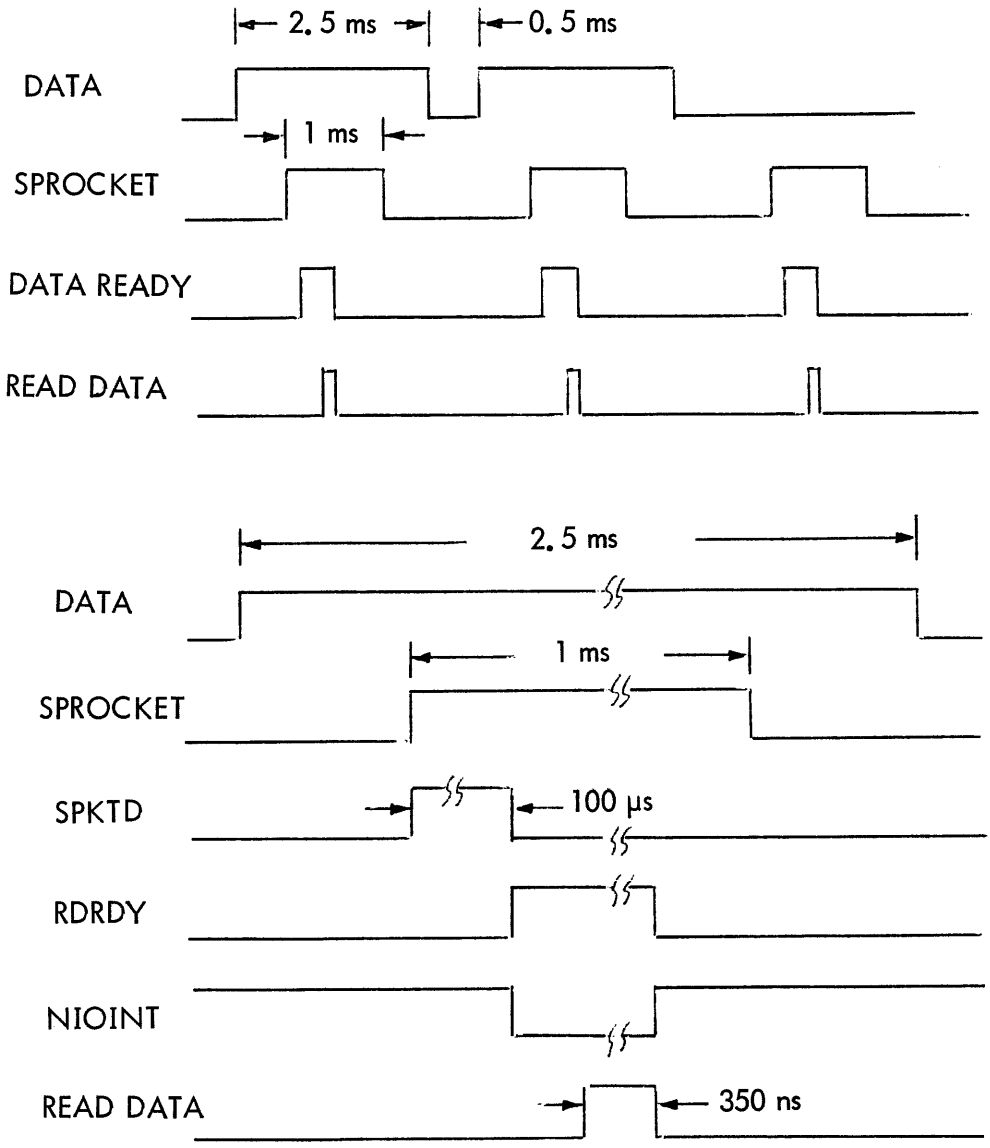


Figure 4-37. Timing Diagram, Paper Tape Reader Input Data

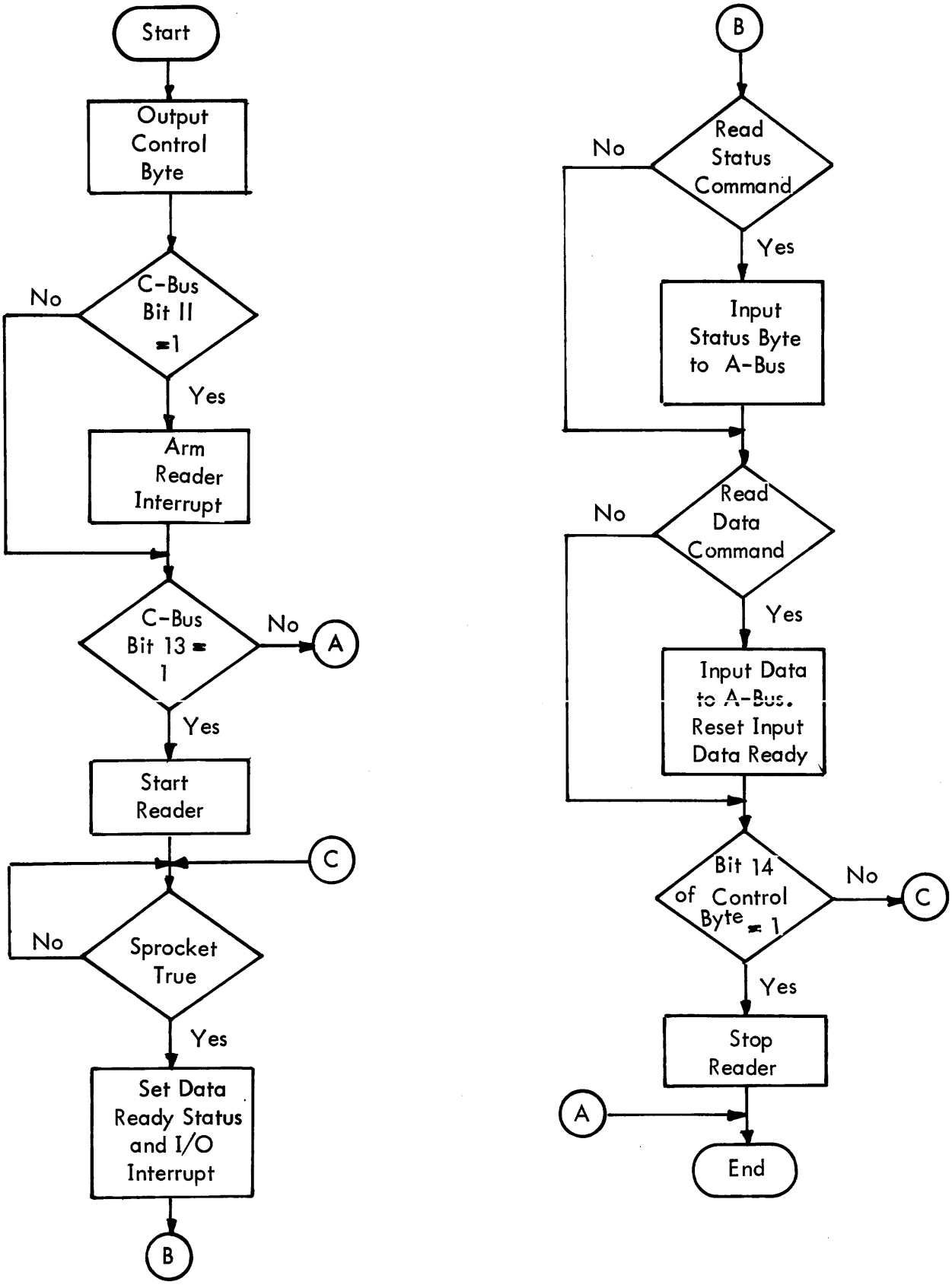


Figure 4-38. Functional Flow Diagram, Paper Tape Reader

C11: Arm Reader Interrupt. Enables reader input data interrupt, allowing the multiplexed I/O interrupt from the reader to be generated.

C12: Disarm Reader Interrupt. Disables reader input data interrupt, preventing the multiplexed I/O interrupt from the reader.

C13: Start Reader. Starts forward motion of paper tape and sets reader in run mode.

C14: Stop Reader. Stops motion of paper-tape reader. To stop the paper tape reader on a character or on the space following the character, the stop command must be applied within 50 microseconds from the time that the stop character is read.

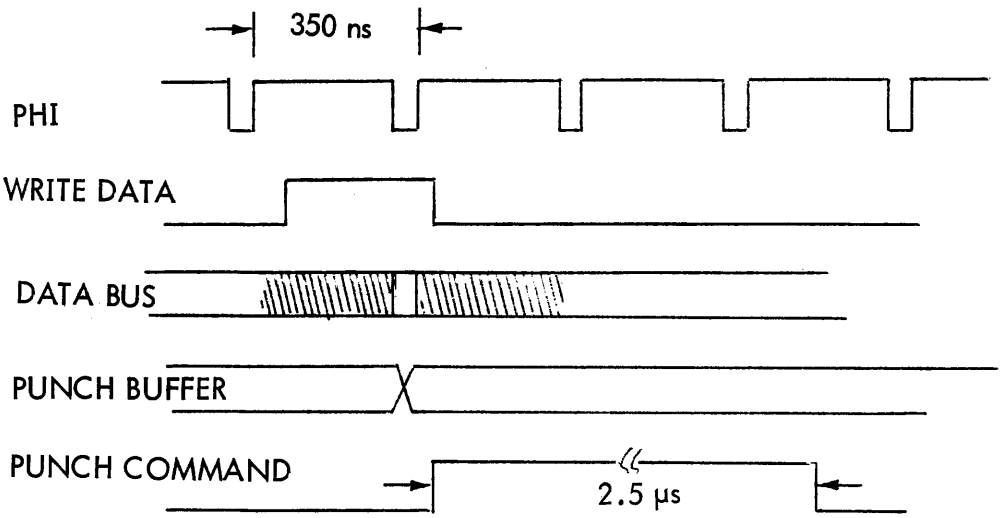
C15: Reset Reader Interface Unit. Unconditionally resets all control functions of the reader interface, including start control, interrupt, and reader ready.

8 Not assigned.

9 Write HS Punch Output Data. Transfers right byte of C Bus to paper-tape punch output data buffer register and generates the punch command. Data may be outputted to the paper-tape punch a single byte at a time, with or without using the input to the multiplexed IO interrupt. Punch timing is diagrammed in Figures 4-39 and 4-40.

Without Interrupt. If the interrupt is not being used, the paper-tape status byte is brought in by means of IO Control = x'6' and tested for Punch Power On (bit 9), Punch Not Ready (bit 11), and Punch Output Buffer Empty (bit 12). If these conditions are met, a data byte may be output by means of Write Punch Output Data. Upon loading the data buffer register, a punch command is generated and status bit 11, Punch Not Ready, sets and status bit 12, Punch Output Buffer Empty, resets. Punch Not Ready status remains true for 13 milliseconds.

With Interrupt. If the multiplexed I/O interrupt is being used, the paper-tape control byte with bit 8, Arm Punch Interrupt, is output initially by means of IO Control = x'7'. Then, provided Punch Power On and Punch Output Buffer Emtpy are true and Punch Not Ready is false, an output to the multiplexed I/O interrupt is generated. As a result of the interrupt, the SCU reads the address of the I/O module to the A-Bus and reads the status bytes to determine which of the devices caused the interrupt. Paper-tape status byte is read in by means of IO Control = x'6'. If status bit 12, Punch Output Buffer Empty, is true, a data byte may be output by means of Write Punch Output Data command. Upon loading the output data buffer register, a punch command is generated and status bit 12, Punch Output Buffer Empty and the interrupt request reset. Punch Not Ready status sets and remains true for 13 milliseconds, preventing the generation of another interrupt.



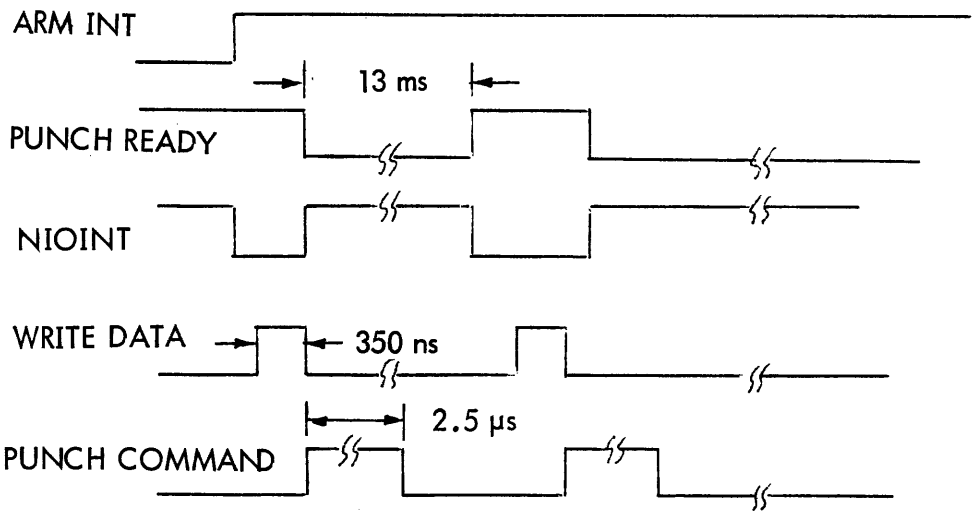


Figure 4-39. Timing Diagram, Paper Tape Punch Data Output Timing

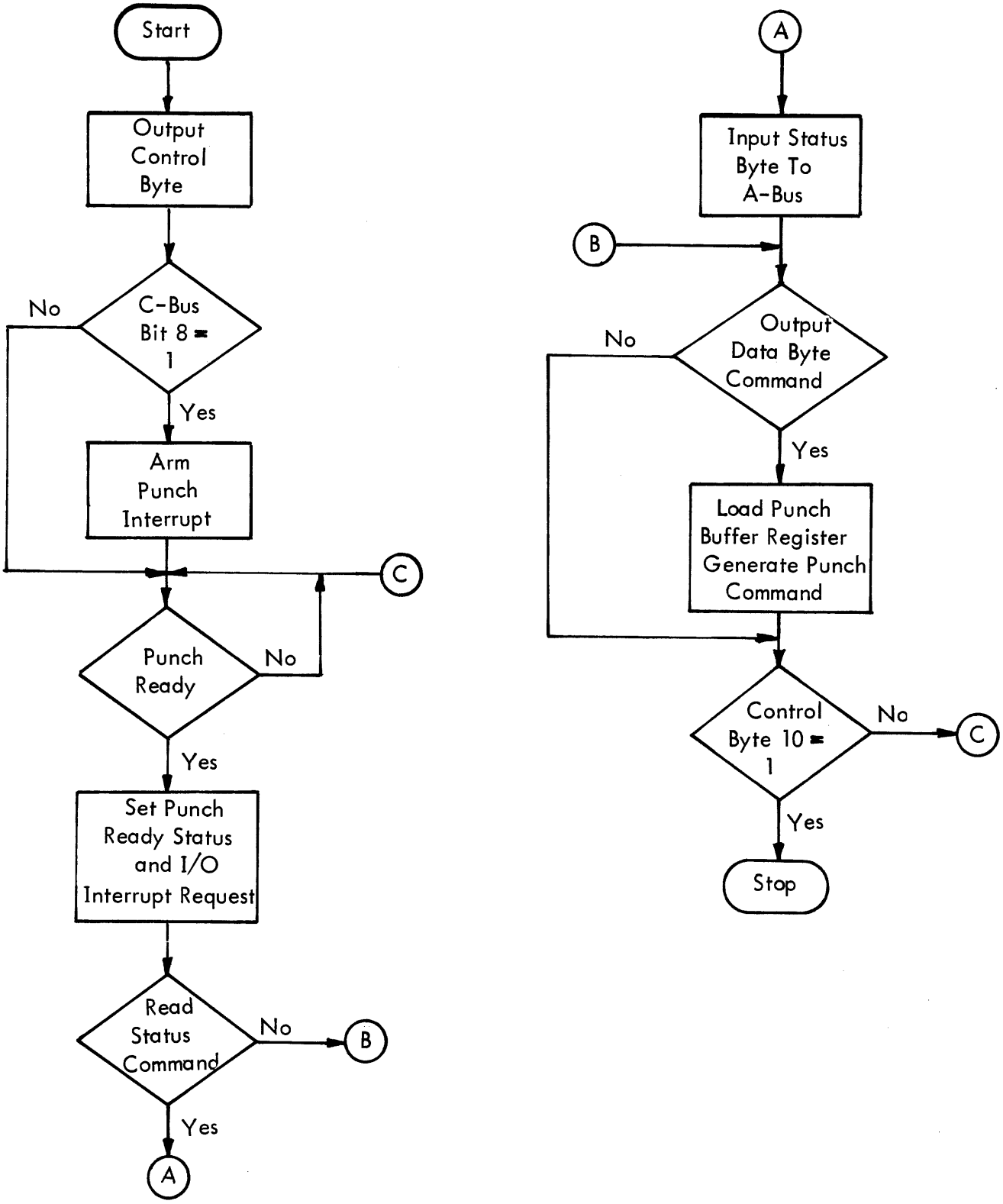


Figure 4-40. Functional Flow Diagram, Paper Tape Punch

Upon completion of output data to the punch, a punch reset should be executed or the punch interrupt disarmed to inhibit any additional interrupt requests from the punch.

A Not assigned.

B Not assigned.

C Read RTC Counter and Reset. Reads contents of the real-time clock counter to the right byte of the A-Bus, stops the RTC counter, and resets the interrupt request. The input data read is the one's complement of the actual count.

D Write Output Data to RTC and Start. Outputs a data byte from the right byte of the C-Bus to the eight-bit real-time clock counter and starts the RTC counter. The output data byte must be the one's complement of the actual number of counts required.

E Not assigned.

F Not assigned.

SECTION V
OPERATOR CONTROLS

The System Control Unit is capable of operating automatically without operator controls or indicators, of being operated manually from the optional control panel, or of being exercised by means of a resident field verification micro program.

5-1 AUTOMATIC OPERATION

The SCU can operate automatically without operator controls or indicators in some operating modes. One example is the operating mode in which the control micro program is fixed in a read only memory. In this case, when power is turned on, the unit executes a power-on interrupt and proceeds to execute the micro program. A control panel is not needed for this type of operation.

In other operating modes, however, a control panel may be necessary in order to load external data into the SCU. One example is the emulation mode, in which a macro instruction set is loaded from an I/O device into scratch pad/main memory. Another example is the mode in which one micro program can be substituted for another in a random-access read/write control memory.

In short, whenever the mode used requires loading scratch pad/main memory or read/write control memory on an unpredictable basis (that is, when loading cannot be planned in advance and included as micro instructions in the fixed micro program), the control panel is required.

5-2 CONTROL PANEL

The maintenance control panel, Figure 5-1, provides controls and indicators that are used to display the current status of the machine and to make changes in that status. The panel serves three purposes:

1. To load data from an I/O device into scratch pad/main memory and/or read/write control memory.

2. To debug or check out programs.

3. To maintain or troubleshoot the machine.

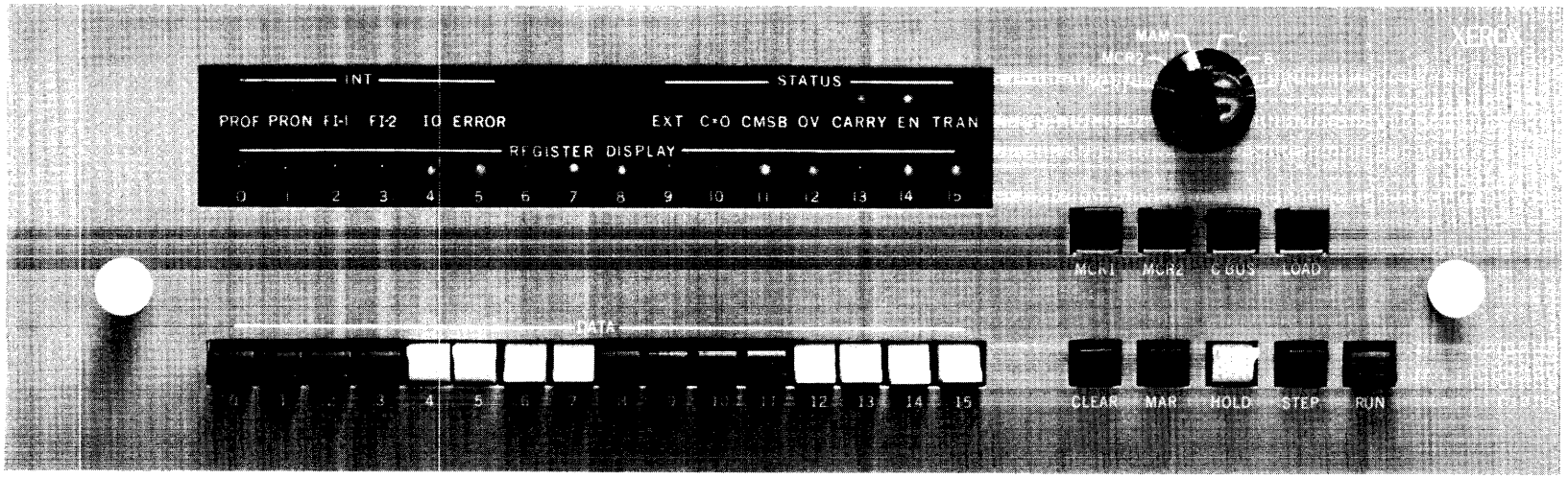


Figure 5-1. Maintenance Control Panel

5-3 CONTROLS

Controls include 16 Data Entry switches, four switches to enter data into registers or the C-Bus, five function switches, and one rotary display select switch.

Data Entry. The state set in these 16 switches is entered into the Micro Control Register, Micro Address Register, or C-Bus when one of the following switches is pressed:

MCR-1 Alter. When pressed, enters the data into the most significant half of the Micro Control Register.

MCR-2 Alter. When pressed, enters the data into the least significant half of the Micro Control Register.

MAR Alter. When pressed, enters the data into the Micro Address Register.

C-Bus Alter. When pressed, enters the data onto the C-Bus and executes the micro instruction in the Micro Control Register while blocking other sources from the C-Bus. Data entry is inhibited in Run mode.

Clear. When pressed, resets the Micro Control Register, Micro Address Register, Interrupt Register, and Status Register.

Run. When pressed, transfers the SCU from Halt to Run mode. That is, the Run switch applies the master clock to the timing and control logic, permitting micro instructions to be executed on machine cycle time. All other switches except Hold and Autoload are inhibited when the Run switch is active.

Step. Each time the Step switch is pressed, the SCU fetches the next micro instruction while executing the present micro instruction. This switch is useful in troubleshooting and program debugging.

Hold. When pressed, allows the execution of the micro instruction presently contained in the Micro Control Register, but does not advance or change the contents of this register. The Micro Address Register is not modified by the micro instruction being executed.

Autoload. This switch has two functions, depending on the state of Run switch. When the SCU is halted (Run switch off), pressing this switch fetches the micro instruction in control memory location X'A', the Autoload function. This micro instruction, when the Step/Run switch sequence is performed, normally leads to a control memory subroutine that loads data from a peripheral unit into the main memory. When the SCU is running (Run switch on), pressing this switch fetches the micro instruction in control memory location X'B', the Console Interrupt function. This micro instruction may lead to a Console Interrupt subroutine.

Select. A six-position rotary switch used to inspect data in registers and buses:

1. MCR-1: Most significant 16 bits of Micro Control Register
2. MCR-2: Least significant 16 bits of Micro Control Register
3. MAM: Memory Address Multiplexer (12 bits)
4. C: C-Bus (16 bits)
5. B: B-Bus (16 bits)
6. A: A-Bus (16 bits)

5-4 INDICATORS

The indicators are red light emitting diodes, arranged in three rows; Interrupt status, Machine status, and Register or bus data.

In Run mode the indicators come on and go off at one-eighth the micro instruction operating rate and therefore cannot be interpreted correctly. In Step mode the indicators are on or off for the period between operations of the Step switch.

5-5 Interrupt Indicators

The top row of indicators contains six light-emitting diodes that report from left to right in order of priority the state of the six bits in the Interrupt Register on the MCR-1 module:

PROFF. Power Off. The indicator illuminates during the micro-instruction cycle after the receipt of the Power Off signal from the power supply reporting that power is dropping.

PRON. Power On. The indicator illuminates during the next micro-instruction cycle after the receipt of the Power On signal from the power supply reporting that power has returned.

FI1. Fast Interrupt 1. The indicator illuminates during the next micro-instruction cycle after the receipt of Fast Interrupt 1 and remains on until the higher priority Power On interrupt subroutine, if it is in process, is completed.

FI2. Fast Interrupt 2. The indicator illuminates during the next micro-instruction cycle after the receipt of Fast Interrupt 2 and remains on until any higher priority interrupt subroutines, if any are in process, are completed.

IO. Input/Output. The indicator illuminates during the next micro-instruction cycle after the receipt of the Input/Output interrupt, providing data from an I/O module is not presently being accepted, and the I/O interrupts are enabled. The indicator remains on until any higher priority interrupt subroutines, if any are in process, are completed.

ER. Parity Error. The indicator illuminates during the next micro-instruction cycle after the receipt of a Parity Error interrupt, reporting a parity error in control memory, scratch pad, or main memory. The indicator remains on until any higher priority interrupt subroutines, if any are in process, are completed.

5-6 Status Indicators

The second row of indicators reports the state of the six status terms currently contained in the Status Register, plus the External condition line:

OV. Overflow. This indicator illuminates in arithmetic operations if the result of the A operand and the B operand is greater than the largest number that can be accommodated in the bit space available, either 8-bit or 16-bit operation. In addition, this indicator represents the current state of the special function, Set Overflow Flag, or the previous state of the same function being pulled from the Push Stack.

C = O. C-Bus Equals Zero. This indicator illuminates when direct examination of the C-Bus establishes that all C bits are false, or when the C = 0 term being pulled from the Push Stack is true.

CMSB. C-Bus Most Significant Bit. This indicator represents the state of C00 in word operation or C08 in byte to byte operation, indicating the sign of the number on the C-Bus: 0 for positive, 1 for negative. During a Pull micro opcode, the indicator assumes the state of the CMSB term being read out of the Push Stack.

CARRY. Carry. This indicator represents the state of the Carry bit from the most significant bit of either the byte or word being operated on. During a Pull micro opcode, the indicator assumes the state of the Carry term being read out of the Push Stack.

ENABLE. Enable I/O Interrupts. This indicator represents the state of the MCR Control micro opcode, Enable I/O Interrupts, or the previous state of the same function being pulled from the Push Stack.

TRAN. Control Mode Flag. This indicator represents the state of the current special function, Set Control Mode Flag, or the previous state of the same function being pulled from the Push Stack.

EXT. External Condition Test Bit Flag. This indicator represents the state of the external sense line defined in the RA Select and Device Select fields only. NOTE: No provision is made to save the status of this bit during a PUSH/PULL operation.

5-7 Register Display Indicators

The third row of indicators reports the state of 16 bits of the register or bus selected by the Select rotary switch.

5-8 MANUAL OPERATION

A variety of manual operations can be carried out from the Maintenance Control Panel, for example--

To examine the operation of a particular few micro instructions somewhere in the control memory:

1. Press Clear switch, resetting the Micro Address Register, the Micro Control Register, and other registers.

2. Enter the micro address for the first micro instruction of the control memory sequence to be examined on the 12 low-order Data Entry switches.

3. Press the MAR Alter switch, entering this micro address into the Micro Address Register.

4. If desired, the 12-bit micro address can be inspected on the Register Display by turning the Select rotary switch to the MAM position.

5. Press Step switch, generating a single clock which transfers the micro instruction addressed by the micro address to the 32-bit Micro Control Register. This clock also advances the Micro Address Register to the next micro address.

6. If desired, the content of each half of the Micro Control Register can be inspected on the Register Display by turning the Select rotary switch to the MCR1 or MCR2 position.

7. From this point, operations depend upon the micro opcodes contained in the micro instruction now in the Micro Control Register. In general, however, inputs to the Arithmetic Logic Units can be observed on the A-Bus and B-Bus and the result of the logical or arithmetic operation on the C-Bus.

8. Press the Step switch again, placing the next micro instruction in the Micro Control Register. The results of these micro opcodes can now be observed on appropriate buses or registers.

5-9 FIELD VERIFICATION

The optional Field Verification Module contains a set of 256 micro instructions in a read only memory similar to a control memory. This micro program verifies the operation of all configurations of standard and optional modules, except input/output modules, translators, and maintenance control panel.

The diagnostic procedure follows:

Turn off SCU power. Place Test switch (S1) on Field Verification module in location 11 in ON position. Place Bank Select switch (S2) on Field Verification module in SEL1 position.

Turn SCU power on, causing the unit to execute a modified Power On interrupt to control memory location X'F05' where the first micro instruction of the field verification micro program is stored.

From this micro instruction, the field verification micro program proceeds automatically through the steps listed in Table 5-1.

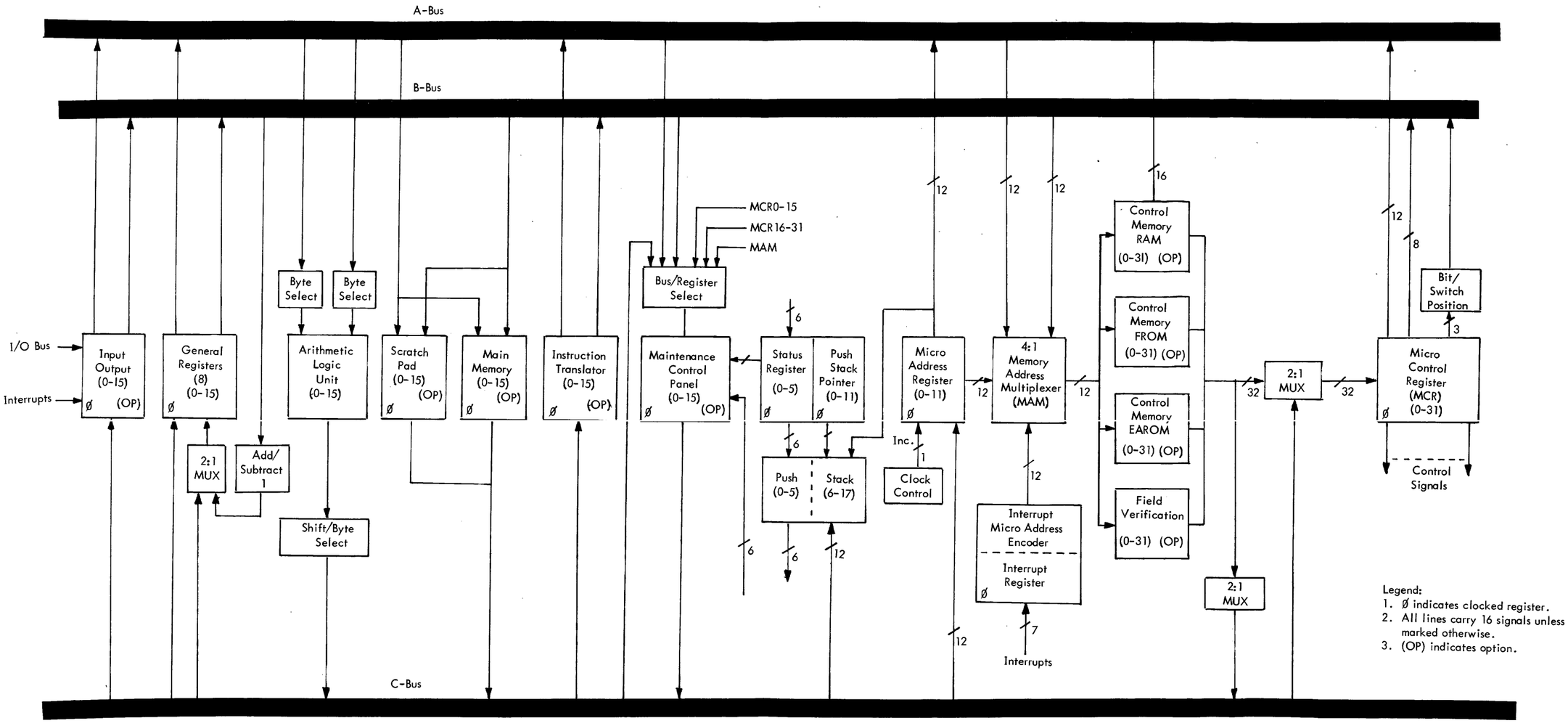
The micro program halts if any step fails. The micro address of the halt is displayed on eight light-emitting diodes, DS9 - DS16, on the Field Verification module, indicating the micro instruction that could not be completed.

Table 5-1. Functions Tested by Field Verification Micro Program

Halt	Functions Tested	Modules Involved
03	Transfer bits 20-31 of micro instruction (immediate address) from Micro Control Register via A-Bus, ALU, C-Bus to Micro Address Register.	MCR1, MCR2, MAR, ALU
34	Store data successively in every address of general registers 0 - 7, read out, and compare with original data in.	GR
38	Test word-length arithmetic operation designed to result in a negative number on the C-Bus.	ALU
3A	Test word-length arithmetic operation designed to result in overflow.	ALU
3D	Test arithmetic operation designed to result in zero on C-Bus.	ALU
3E	Test arithmetic operation designed to result in a carry.	ALU
42	Test remaining arithmetic functions, designed to result in C-Bus = 0.	ALU
50	Test logical operations	ALU
54 55 57 58	Test byte mode arithmetic operations	ALU
7A	Push data into push stack 16 times and pull data from push stack 16 times.	MAR

Table 5-1. Functions Tested by Field Verification Micro Program (Continued)

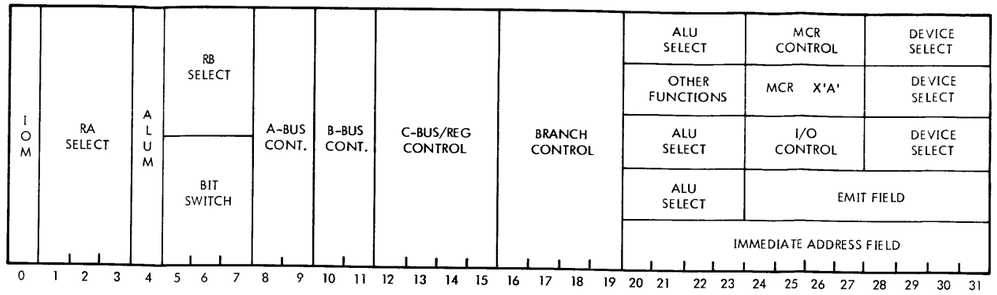
Halt	Functions Tested	Modules Involved
7E	Enable IO Interrupts, simulate all interrupts (except Power On), and test receipt of all interrupts. Halt signifies no interrupt received.	MCR1, MAR
81	Test Autoload or Console Interrupt priority	MCR1
86	Test Error Interrupt priority. (When testing control memory, scratch pad, or main memory, if a parity error occurs, the field verification micro program returns to this test and halts at this location.)	MCR1
88	Test IO Interrupt priority.	MCR1
8E	Test Fast Interrupt 2 priority.	MCR1
92	Test Fast Interrupt 1 priority	MCR1
96	Test Power Off Interrupt priority.	MCR1
AA	Write into each read/write control memory location, read and compare with data in; or read each control memory read only location and obtain check sum of zero. (CM ROM's are programmed with one word equal to a constant that makes the sum of all words in all CM ROM's installed zero.) If program stops at this step, replace CM ROM modules one by one with spares containing the same micro program until the defective module is found.	CM1 = CM5
D9	Write and read walking 1's and 0's pattern in scratch pad/ main memory by pages. If program stops at this step, the B-Bus indicators, DS1 - DS8, on the Field Verification module give the binary number of the 256-word block of memory at which failure occurred.	SP/MM
FB	Test complete. The B-Bus indicators give the binary number of the 256-word block of memory successfully tested.	
FB	To run the field verification micro program in a loop, without halting at FB, place EXT switch (S3) on Field Verification module to ON. Start test by turning SCU power off, then on, as before. Stop test by placing Test switch (S1) on Field Verification module to OFF position.	


A-Bus
B-Bus
C-Bus
I/O Bus
Interrupts
Input Output (0-15) (OP)
General Registers (8) (0-15)
Byte Select
Byte Select
Arithmetic Logic Unit (0-15)
Scratch Pad (0-15) (OP)
Main Memory (0-15) (OP)
Instruction Translator (0-15) (OP)
Bus/Register Select
MCR0-15
MCR16-31
MAM
Maintenance Control Panel (0-15) (OP)
Status Register (0-5)
Push Stack Pointer (0-11)
Push (0-5)
Stack (6-17)
Micro Address Register (0-11)
Inc.
Clock Control
4:1 Memory Address Multiplexer (MAM)
Interrupt Micro Address Encoder
Interrupt Register
Interrupts
Control Memory RAM (0-31) (OP)
Control Memory FROM (0-31) (OP)
Control Memory EAROM (0-31) (OP)
Field Verification (0-31) (OP)
2:1 MUX
2:1 MUX
Add/ Subtract 1
Shift/Byte Select
Bit/ Switch Position
Micro Control Register (MCR) (0-31)
Control Signals
Legend:
1. Ø indicates clocked register.
2. All lines carry 16 signals unless marked otherwise.
3. (OP) indicates option.

ALU Select

ALU Select Bits 20–23 (Hexidecimal)	ALU Mode Reset		ALU Mode Set
	Arithmetic Function Carry In False (Branch Control = 0)	Arithmetic Function Carry In True (Branch Control = 1)	Logic Function
0	C ⇐ A	C ⇐ A plus 1	C ⇐ $\overline{A}$
1	C ⇐ A + B	C ⇐ (A + B) plus 1	C ⇐ $\overline{A + B}$
2	C ⇐ A + $\overline{B}$	C ⇐ (A + $\overline{B}$) plus 1	C ⇐ $\overline{A}$ · B
3	C ⇐ Minus 1	C ⇐ 0	C ⇐ 0
4	C ⇐ A plus A · $\overline{B}$	C ⇐ A plus A · $\overline{B}$) plus 1	C ⇐ $\overline{A \cdot B}$
5	C ⇐ (A + B) plus A · $\overline{B}$	C ⇐ (A + B) plus A · B plus 1	C ⇐ $\overline{B}$
6	C ⇐ A minus B minus 1	C ⇐ A minus B	C ⇐ A ∀ B
7	C ⇐ A · $\overline{B}$ minus 1	C ⇐ A · $\overline{B}$	C ⇐ A · $\overline{B}$
8	C ⇐ A plus A · B	C ⇐ A plus A · B plus 1	C ⇐ $\overline{A}$ + B
9	C ⇐ A plus B	C ⇐ A plus B plus 1	C ⇐ $\overline{A \forall B}$
A	C ⇐ (A + $\overline{B}$) plus A · B	C ⇐ (A + $\overline{B}$) plus A · B plus 1	C ⇐ B
B	C ⇐ A · B minus 1	C ⇐ A · B	C ⇐ A · B
C	C ⇐ A plus A	C ⇐ A plus A plus 1	C ⇐ 1's
D	C ⇐ (A + B) plus A	C ⇐ (A + B) plus A plus 1	C ⇐ A + $\overline{B}$
E	C ⇐ (A + $\overline{B}$) plus A	C ⇐ (A + $\overline{B}$) plus A plus 1	C ⇐ A + B
F	C ⇐ A minus 1	C ⇐ A	C ⇐ A

Key: $\overline{n}$ = 1's complement of n; + = OR function;
· = AND function; plus = add; minus = subtract;
∀ = Exclusive OR function; ⇐ = Replaced with.



Other Functions

Bits 20–23 (Hexadecimal)	Other Functions (MCR = X'A')
8	Reset OV Flag
9	Set OV Flag
A	Reset Control Mode Flag
B	Set Control Mode Flag

A-Bus Control

Bits 8 9	Function
0 0	RA Selected Source to A-Bus
0 1	Left Byte of A-Bus to ALU-A (Right Justified)
1 0	Right Byte of A-Bus to ALU-A (Right Justified)
1 1	MAR to A-Bus (Right Justified)

B-Bus Control

Bits 10 11	Function
0 0	RB Selected Source to B-Bus
0 1	Left Byte of B-Bus to ALU-B (Right Justified)
1 0	Right Byte of B-Bus to ALU-B (Right Justified)
1 1	If IOM = 0, Emit Field to B-Bus (Right Justified) If IOM = 1, RB Select Defines Bit Position on B (8–15). RB Select Bits 5 6 7 — Bit Position Assignment 0 0 0 — 1 to B15 0 0 1 — 1 to B14 0 1 0 — 1 to B13 0 1 1 — 1 to B12 1 0 0 — 1 to B11 1 0 1 — 1 to B10 1 1 0 — 1 to B09 1 1 1 — 1 to B08

C-Bus/Register Control

Bits 12–15 (Hexadecimal)	Function
0	No Strobe to any Register
1	C-Bus to MAR (2 Clock Periods)
2	Use IO Mode, RA Select, Device Select and A-Bus Control as Destination
3	Use RB Select and B-Bus Control as Destination
4	Shift ALU Left One; Use IO Mode, RA Select, Device Select and A-Bus Control as Destination
5	Shift ALU Left One; Use RB Select and B-Bus Control as Destination
6	Shift ALU Right One; Use IO Mode, RA Select, Device Select, and A-Bus Control as Destination
7	Shift ALU Right One; Use RB Select and B-Bus Control as Destination
8	Increment B-Bus and Place Result in Register Selected by RB Select
9	Increment B-Bus; C-Bus to MAR (2)
A	Increment B-Bus; Use IO Mode, RA Select, Device Select, and A-Bus Control as Destination
B	If IOM = 0, Immediate Addressing Invoked to MAR, If IOM = 1, Device Address for IO Interrupt
C	Increment B-Bus; Shift ALU Left One; Use IO Mode, RA Select, Device Select and A-Bus Control as Destination
D	Not Assigned
E	Increment B-Bus; Shift ALU Right One; Use IO Mode, RA Select, Device Select, and A-Bus Control as Destination
F	Decrement B-Bus; Use IO Mode, RA Select; Device Select, and A-Bus Control as Destination

I/O Mode

Bit 0	Mode	RA Select Bits 1, 2, 3	RB Select Bits 5, 6, 7
0	General Register	Addresses GR(0–7)	Addresses GR(0–7)
1	I/O	With Bits 28–31 (Device Select) Addresses IO(0–127)	Addresses GR(0–7)

MCR Control (IOM = 0; BBC ≠ 3; CB/RC ≠ X'B')

Bits 24–27 (Hexadecimal)	Function
0	No Special Function
1	C-Bus To Translator (Optional)
2	Reset MAR (2)
3	Read Vector 1 to A-Bus from Translator (Optional)
4	Enable I/O Interrupts
5	Read Vector 2 to A-Bus from Translator (Optional)
6	Disable I/O Interrupts
7	Read Argument to B-Bus from Translator (Optional)
8	Read MCP Data Entry Switches to C-Bus (Optional)
9	Not Assigned
A	Other Functions
B	Not Assigned
C	Write Micro Control Memory (Optional) (2)
D	Read Micro Control Memory
E	Read I/O Interrupts to A-Bus
F	Not Assigned

Branch Control

Bits 16–19 (Hexadecimal)	Function
0	Normal Mode Carry False
1	Normal Mode Carry True
2	Read Scratch Pat [1] /Main Memory [2] (Optional)
3	Write Scratch Pad [1] /Main Memory [2] (Optional)
4	Test/Branch on External Condition Set [3]
5	Test/Branch on External Condition Reset [3]
6	Push [4]
7	Pull [2]
8	Test/Branch OV = 1 [3]
9	Test/Branch C-Bus = 0 [3]
A	Test/Branch Carry = 1 [3]
B	Test/Branch CMSB = 1 [3]
C	Test/Branch OV ≠ 1 [3]
D	Test/Branch C-Bus ≠ 0 [3]
E	Test/Branch Carry ≠ 1 [3]
F	Test/Branch CMSB ≠ 1 [3]

(1) One Clock Period for Scratch Pad
(2) Double Clock Period for Main Memory
(3) One Clock Period if Unsuccessful; Double Clock Period if Successful.
(4) One Clock Period with Immediate Addressing; Double Clock Period Otherwise.