# SCC 4700 COMPUTER

# INTERFACE MANUAL

SCIENTIFIC CONTROL CORPORATION

# SCC 4700 COMPUTER INTERFACE MANUAL

SCIENTIFIC CONTROL CORPORATION
1215 W. Crosby Rd., Carrollton, Texas 75006 214—242-6555

SCC 79 10-46-91

## TABLE OF CONTENTS

## TABLE OF CONTENTS (CONT'D)

# LIST OF ILLUSTRATIONS

## LIST OF ILLUSTRATIONS (CONT'D)

## LIST OF TABLES

# SECTION I

# SYSTEM DESCRIPTION

The SCC 4700 is a general purpose, high-speed, binary computer with a single address type of instruction. It features a high-speed magnetic core memory module consisting of 4,096 sixteen-bit words, with a 920-nanosecond cycle time, which permits a wide variety of real time applications.

The SCC 4700 has outstanding design features such as:

1. A microprogrammed read-only memory for flexible and economical internal logic
2. Fully integrated circuitry using the most advanced TTL integrated circuits
3. An etched circuit back-plane board eliminating "bird nest" wiring
4. "Register slice" internal organization for easy maintainability
5. Programmable memory protection (optional) which provides flexible read-only write-only, or execute-only protection
6. Memory mapping (optional) for implementation of multiprogramming techniques
7. Byte addressing for efficient processing of character strings, particularly those in ASCII or EBCDIC code
8. Real time byte-oriented I/O structure utilizing low cost multiplexor and high-speed selector channels for data transfer
9. Powerful multilevel priority interrupt system which minimizes program overhead and provides quick real time responses
10. Extensive instruction set which simplifies system and application programming while providing outstanding flexibility and power.

## CONFIGURATION

The main modules of the basic SCC 4700 (Figure 1-1) are as follows:

4K Memory Module

Central Processing Unit

Multiplexor Channel

Control/Display Panel.

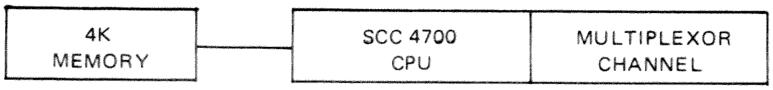


*Figure 1-1. Basic Configuration*

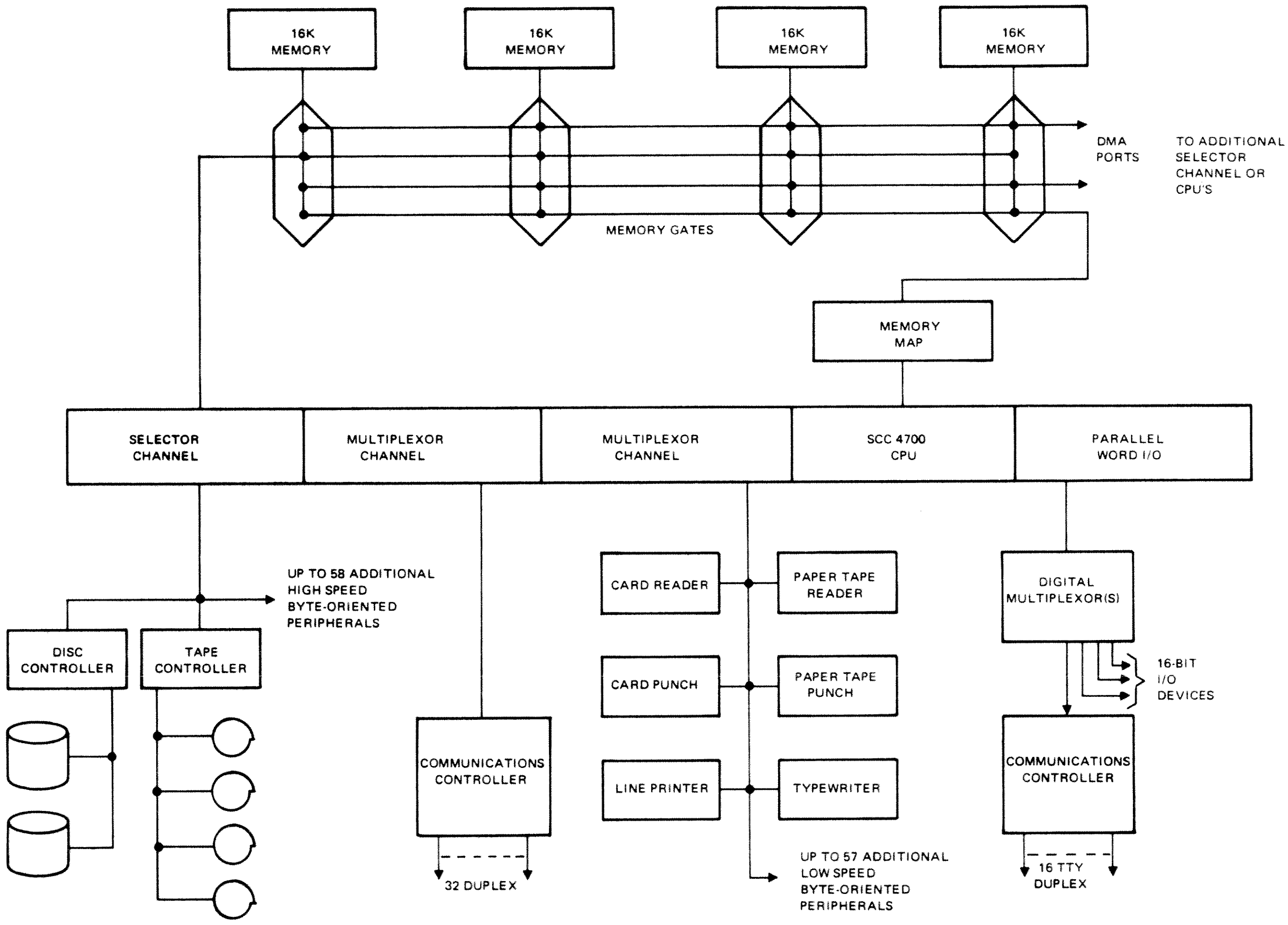


*Figure 1-2. Expanded Configuration Illustrating the Interfaces Described in this Manual.*

The SCC 4700 may be expanded as shown in Figure 1-2 to include the following:

Four 16K Memory Modules

Memory Mapping with up to 32 Associative Registers

Memory Gating with up to 8 DMA Ports

Central Processing Unit with Multiply/Divide, Double Precision, and Floating Point Hardware (Additional CPU's may be added for multiprocessing).

Four I/O Channels in any combination of multiplexor and/or selector channels

Peripheral Devices (up to 64 per channel).

## GENERAL CHARACTERISTICS

The advanced characteristics of Scientific Control Corporation's Model 4700 contributing to its outstanding efficiency for general purpose, real time, and time sharing applications are as follows:

1. **Large Memory Capacity**

   The basic memory module of the SCC 4700 consists of 4,096 sixteen-bit words. The memory capability is expandable to 65,536 words in 4,096-word increments.

2. **High-Speed Memory**

   The 4K, 8K, and 16K memory modules are 2-1/2 dimensional with a 920-nanosecond full cycle time.

3. **Memory Parity**

   Logic circuits are provided if the optional memory parity checking feature is desired. Memory modules with an extra parity bit in each word are available for use with this feature.

4. **Byte Addressing**

   Byte, word, double word, and triple word addressing are provided for maximum efficiency.

5. **Memory Accessing**

   The capability of asynchronous memory access by either the Central Processing Units, Selector Channel Units, or special system components is provided by the four-port memory gate. The memory gates permit independent memory module accessing and allow complete overlapping of input, processing, and output operations. Simultaneous requests of the same module are handled on a priority basis with the priority assignment being a system variable.

6. **Memory Mapping and Protection**

   This optional unit contains a set of associative registers through which program addresses are transformed to real core locations. The memory mapping system divides all available memory into "pages." For each page, the programmer may specify one of three levels of memory protection: read, write, or execute.

Accidental or unauthorized use of a memory location is prevented by checking the memory access code for the page being referenced before execution. Attempts to violate the specified protection causes a trap. In this manner, the protection feature permits the maintenance of system integrity in a multiprogramming environment and complete security of system and application programs.

7. **Microprogrammed Hardware**

The instruction repertoire is microprogrammed to provide a highly flexible and reliable instruction set. The design and organization of the microprogrammed instruction repertoire permits the addition of hardware arithmetic or special function instructions to be implemented economically.

8. **Extensive Instruction Set**

The powerful instruction set allowing efficient and flexible programming includes:

a. **Addressing Modes**

The addressing modes available to the programmer permit several modes of memory addressing: Indirect, Indexed, Direct, and Relative. Both pre-indexing and post-indexing are provided with a hardware index register.

b. **Load and Store Instructions**

Direct loading and storing of major programmable registers including byte and multi-register operations may be performed in a single instruction.

c. **Literal Instructions**

Literal instructions which utilize the last 9 bits of the instruction as a signed operand are included in the instruction set. This feature permits high-speed arithmetic and logical operations involving a single character operand.

d. **Inter-Register Instructions**

A special set of instructions called the Operate Instructions provide logical and arithmetic operations between the A, B, X, and E registers. This feature also provides the capability of performing a skip, depending on the result of the logical or arithmetic operation performed upon the registers. Data may also be exchanged between two registers in only 1.1 microsecond.

e. **Shift Instructions**

The instruction repertoire includes 16 indexable shift instructions. Shifts may be made to the left or right on a single 16-bit word in the A register or on two 16-bit words in the A and B registers.

f. **System Instructions**

System calls and returns permit up to 64 direct access entry points to monitor functions and provide efficient context switching by saving and restoring machine status and programmable registers.

g. **Privileged Instructions**

These instructions prevent unauthorized changing of machine status by user programs.

Also included in the instruction set are several special instructions which reduce total system cost and program overhead by providing efficient handling of special functions.

9. **Hardware Arithmetic**

Three optional high-speed hardware arithmetic instruction sets may be added to the standard instruction repertoire.

a. Integer Multiply and Divide

b. Fractional Double Precision to provide greater accuracy through additional significant digits.

c. Either Floating Point or Double Floating Point (but not both) to provide greater precision and versatility in scientific application areas.

10. **System Control**

System control and autonomy in multiprogramming environments are maintained and simplified through hardware design which facilitates transfer of control, parameters, and status information between the system and the user.

11. **Fully Parallel Operation**

Fully parallel internal processing of all instructions together with full parallel data transfer between memory and the CPU provide a high-speed system performance capability.

12. **Versatile I/O**

The SCC 4700 may use from one to four byte-oriented input/output channels to communicate with memory either directly or through the Central Processing Unit. The channel units may be multiplexor and/or selector channels, each of which can interface with up to 64 device controllers. These channels provide a wide variety of input/output capability from single byte (character) data transfer controlled directly by the CPU to transfer of data by blocks asynchronously and independently of the CPU.

Through the use of pointers, the input/output system permits data chaining to provide scatter read/gather write techniques and the use of separate control and data areas.

The I/O channels are compatible with all of the SCC 700 series configurations.

Data transfer of full 16-bit words directly to and from the accumulator is accomplished through the program controlled parallel I/O, standard with the SCC 4700. Special high-speed full word transfers directly to memory are performed by the optional Direct Memory Access (DMA) ports.

13. **Automatic Initial Program Loading**

For operator convenience, automatic initial program loading is provided.

14. **Watchdog Timer**

Program hangup due to I/O failure is prevented by a watchdog timer.

15. **Priority Interrupt System**

The SCC 4700 interrupt capability allows the normal execution of a program to be interrupted in order to execute a program of higher priority. The priority structure resolves contention problems arising from the simultaneous occurrence of interrupt conditions and permits servicing of interrupts according to priority.

Two interrupts are standard: console and I/O channel. In addition, two conditions cause traps to reserved locations: real time clock = 0 and unimplemented instructions.

Up to 256 external interrupts in 16 levels, power up, power down, three additional channel interrupts, memory parity, privileged instruction, floating point over/under flow, and memory protection may be added to the SCC 4700 to fulfill specialized requirements.

16. **Real Time Clock**

Standard in the SCC 4700 is a real time clock which decrements a fixed location and causes a trap when the contents of the location reach zero.

17. **Power Failsafe Protection**

The design of the optional power failsafe feature ensures program integrity in the event of power system failures. If power failure occurs, this feature activates an interrupt which stores the active registers in memory and initiates software routines to provide an orderly system shutdown. Reserve power permits 5 milliseconds of normal operation after detection of power failure. When power returns to the system, the active registers are restored and a power up interrupt is generated.

18. **Remote Control Console**

The control console may be installed at a location remote to the CPU to provide ready access for the operator.

19. **All Integrated Circuits**

All circuits utilize the most advanced, highly reliable, fast-switching, low noise TTL integrated circuit components. These components are mounted on extra large printed circuit plug-in boards in a "register slice" arrangement.

## 20. Software

The software packages which are available with the basic SCC 4700 are as follows:

- SPL Assembler
- Relocatable Loader
- Debug Packages
- Basic FORTRAN Compiler
- Basic FORTRAN Run Time
- Basic FORTRAN Library
- Multiply/Divide Package
- Double Precision Arithmetic Package
- Floating Point Arithmetic Package
- Double Precision Fixed Point Math Subroutines
- Floating Point Math Subroutines
- Binary-Decimal, Decimal-Binary Conversion
- Basic Instruction Diagnostic
- Memory Diagnostic
- ASR 33 Diagnostic
- Multiply/Divide Diagnostic
- Double Precision Diagnostic
- Floating Point Diagnostic

Software for an expanded version of the SCC 4700 includes the Real Time Monitor and FORTRAN IV.

## 21. Teletype Input/Output Device

Two units are available which operate at a rate of 10 characters per second. The ASR 33 and ASR 35 both contain paper tape reader and paper tape punch in addition to the normal typewriter keyboard.

## 22. Selectric Typewriter

The IBM Selectric unit operates at a maximum speed of 15 characters per second and is equipped with a 15 inch pin feed platen, allowing use of continuous form paper.

## 23. Paper Tape Reader with Controller

This 8-level unidirectional unit operates at a nominal speed of 300 characters per second. A bidirectional paper tape spooler is also available.

## 24. Paper Tape Punch with Controller

Two 8-level units are available for punching at a rate of 50 or 120 characters per second.

25. **Disc Storage Unit and Controller**

Three models of the disc storage unit are available with an access time of 17 milliseconds and a memory capacity of 128K, 256K, or 512K bytes. The controller accommodates from one to four disc units.

26. **Disc Pack Drive and Controller**

Three models of the disc pack drive are available: two in non-IBM compatible format with 29- or 54-million bit capacity, and one in IBM compatible format with a capacity of 54-million bits. The controller accommodates from one to four disc pack drives and is available in IBM and non-IBM compatible format.

27. **Magnetic Tape Unit and Controller**

Nine magnetic tape unit models are available: five for 7-track tape and four for 9-track tape. The 7-track units can read or write in densities of 200, 556, or 800 bits per inch at speeds ranging from 25 to 150 inches per second. The 9-track units read/write 800 bits per inch at speeds ranging from 37.5 to 150 inches per second. All tape units are IBM compatible. The controllers for the 7- and 9-track tape units accommodate from one to four tape units.

28. **Line Printer and Controller**

Three models are available with printing rates of 300, 600, or 1000 lines per minute. Each model prints a selection of 64 characters with 132 characters per line.

29. **Card Reader and Controller**

One model is available for reading a 256-character set at a rate of 200 cards per minute.

30. **Card Punch and Controller**

One model is available for punching from a 256-character set at a rate of 100 cards per minute.

31. **Digital to Analog Converter**

The digital to analog converter is an addressable, multichannel, binary, two's complement converter which may be 7-bit plus sign bipolar or 8-bit unipolar.

32. **Multiplexor-Encoder**

This unit permits sequential sampling of 64 analog channels of 0 to 5 volts dc. The number of channels can vary in groups of 8 up to the maximum of 64. The multiplexor-encoder contains a buffer amplifier and a sample and hold amplifier. The unit outputs a 12-bit unipolar or 11-bit plus sign digital data word with a digitizing range of up to 40 kHz. The computer can address the multiplexor-encoder for sampling of the analog channels sequentially.

33. **Communications Equipment**

Three communications controllers are available. One provides line termination and multiplexing for one to three full-duplex line terminals. The second model provides for multiplexing of 1 to 32 full-duplex line terminals. The third model is a bit-oriented communication controller for 1 to 16 teletype circuits with selectable clock rates.

## CENTRAL PROCESSOR UNIT

The basic SCC 4700 Central Processor Unit contains nine 16-bit registers and may be expanded to include two additional registers and the memory map unit. The organization of the CPU provides extremely fast data and arithmetic capability.

Figure 1-3 below is a block diagram of the SCC 4700 Central Processor Unit. All arrows indicate data paths for 16-bit parallel transfers. The optional registers and the memory map unit are enclosed in heavy broken lines.

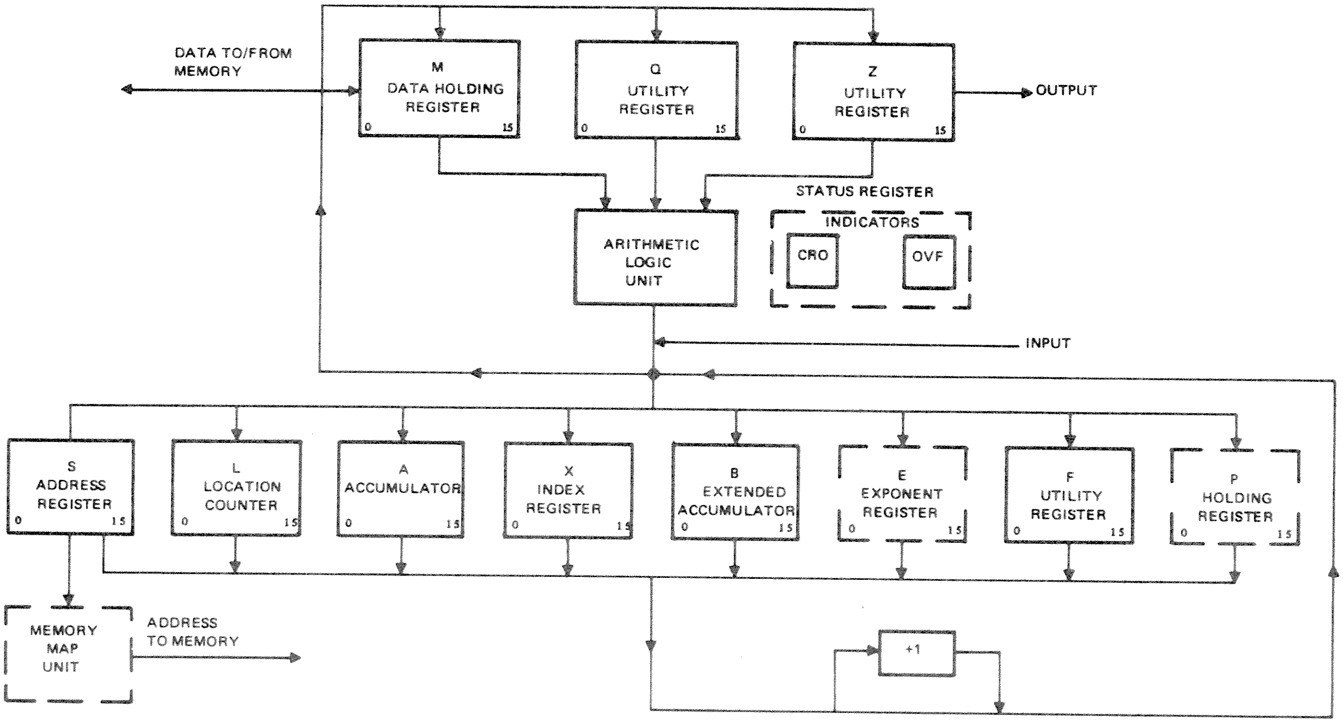


*Figure 1-3. SCC 4700 Central Processing Unit, Registers and Data Paths*

The registers of the CPU are as follows:

**M Register** – The M register accepts and transfers data to and from memory. The M register also serves as a holding register for instructions and data.

**Q Register** – The Q register is a utility register which serves as a holding register for the arithmetic logic unit.

**Z Register** – The Z register is a utility register which also serves as a holding register for the arithmetic logic unit.

**S Register** – The S register is the memory address register which supplies a 16-bit address to the memory unit (or to the memory map unit, if implemented).

**L Register** – The L register is the location counter (program address register) which contains the address of the instruction being fetched.

**A Register** – The A register is the main accumulator. Most arithmetic and logical operations use this register to hold the result of the operation.

**B Register** – The B register serves as a second accumulator and extension of the A register for double precision and floating point operations.

**X Register** – The X register is used for indexing and address modification.

**E Register (optional)** – The E register is utilized during floating point instructions to hold the exponent of a number in floating point format.

**F Register** – The F register is a utility register which serves as a holding register to provide high-speed, internal operations for double precision and floating point instructions.

**P Register (optional)** – The P register is a holding register used to save necessary data during memory mapping, memory protection, and memory parity operations.

**Status Register** – The Status Register is a composite of six indicators which indicate the mode of operation of the machine and the occurrence of certain special conditions. The state of these indicators (0 or 1) can be changed from the control console, under program control, or by the CPU in response to changes in processing or the operating environment.

1. **Indirect Mode Indicator** – This indicator is meaningful only when memory mapping is implemented. The Indirect Mode Indicator allows the system to use the user map on indirect memory references. Bit 10 is set (=1) by a SIUM instruction and reset (=0) by a System Call instruction, the SYSTEM RESET button on the console, or a SISM instruction. This indicator affects the interpretation of the Mode Indicator when indirect addressing is specified and the Mode Indicator is set to system mode. (Refer to Mode Indicator below.)

2. **Mode Indicator** – The Mode Indicator is meaningful only when the memory map unit is implemented. If bit 11 of the Status Register is set (=1), the CPU is in the user mode. Bit 11 is reset (=0) when executing instructions in the system mode; however, if indirect addressing is indicated, bit 11 is given the value of bit 10 above. Privileged instructions may be executed only when this bit indicates system mode. If memory mapping is not implemented, the machine will always be in the system mode.

3. **Interrupt Indicator** – The interrupt system may be enabled or disabled under program control. If bit 12 of the Status Register is set (=1), the interrupt system is disabled and any armed interrupts which occur are remembered but do not interrupt program execution. If bit 12 is reset (=0), the interrupt system is enabled and armed interrupts are allowed to go active according to their priority.

4. **Halt Indicator** – This indicator specifies whether the CPU is in run or halt status. The CPU is halted if bit 13 of the Status Register is set (=1) through use of the Halt instruction, the HALT button, or the SYSTEM RESET button on the console. (Only single step execution of instructions can be performed while this indicator is set.) Bit 13 is reset (=0) by certain interrupts, some traps, and the RUN button on the console.

5. **Overflow Indicator** – The overflow indicator (bit 14 of the Status Register) is set if the result of the arithmetic operation exceeds the maximum signed magnitude quantity which can be contained in the accumulator. Bit 14 is not reset (=0) if an overflow does not occur.

6. **Carryout Indicator** – This indicator (bit 15 of the Status Register) is set (=1) if a carry occurs in the adder from the high order position (bit 0). Bit 15 is reset (=0) if a carry from bit position 0 does not occur.

   An arithmetic operation may result in both an overflow and carryout.

**Memory Map Unit (optional)** – The Memory Map Unit contains 8 to 32 associative registers and 2 map table pointer registers. The number of associative registers may be increased in groups of four.

## CPU LOGIC

The logic signals required to perform the transfer and arithmetic functions within the CPU are shown in Figure 1-4. These signals are provided by the microcontroller portion of the CPU as shown in Figure 1-5. Generally, the operation is as follows: When the microcontroller receives an operand from memory (the operand consists of a ROM microcode address), the control logic loads the microcode address into the O register. The address is then transferred to the 9-bit ROM address input register to address the desired location. The addressed location will in turn enable the required lines (by activating diodes) to generate the transfer or control signals. These signals are actually bits

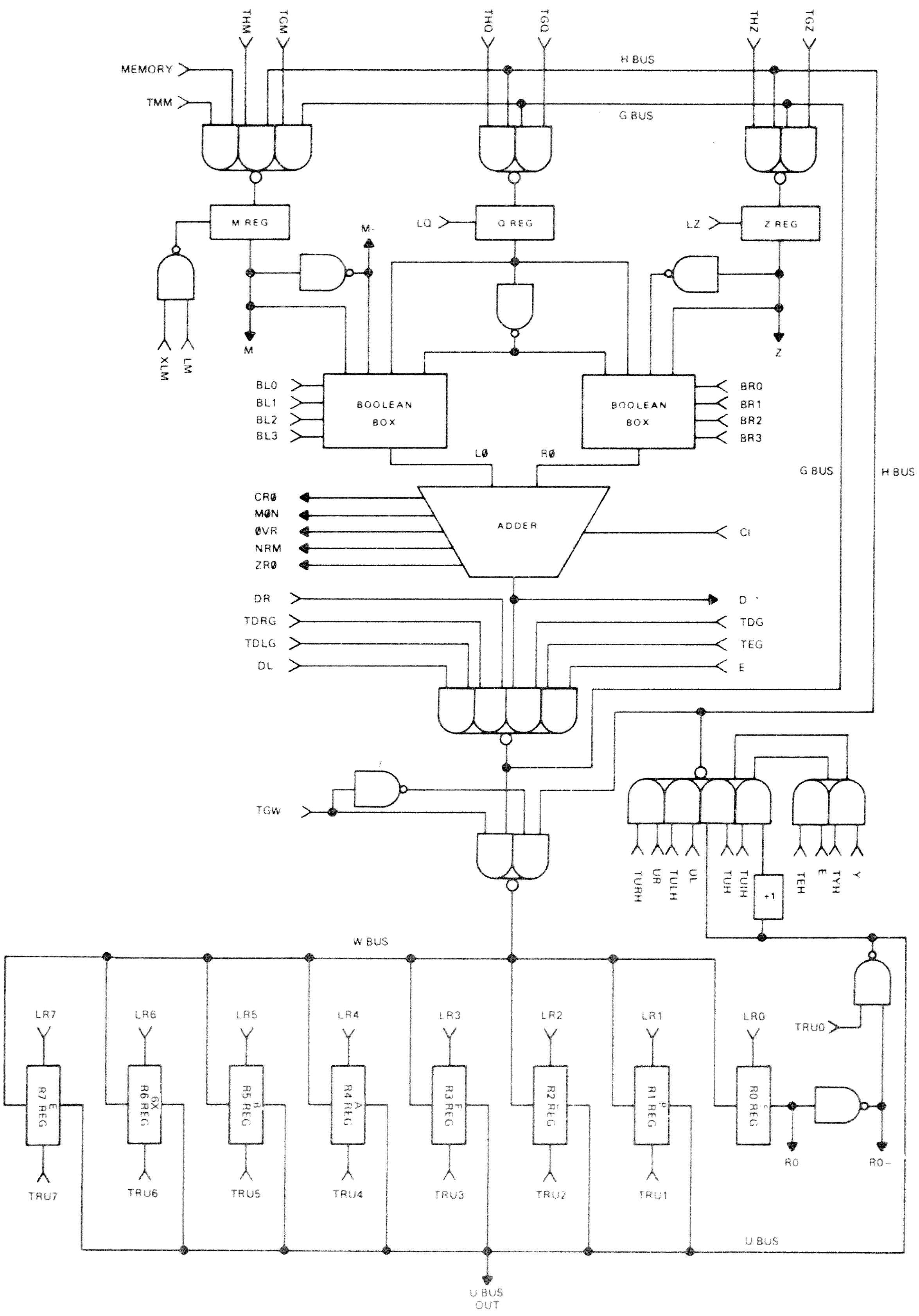


*Figure 1-4. Microprocessor Execution Unit Showing Control Signals*

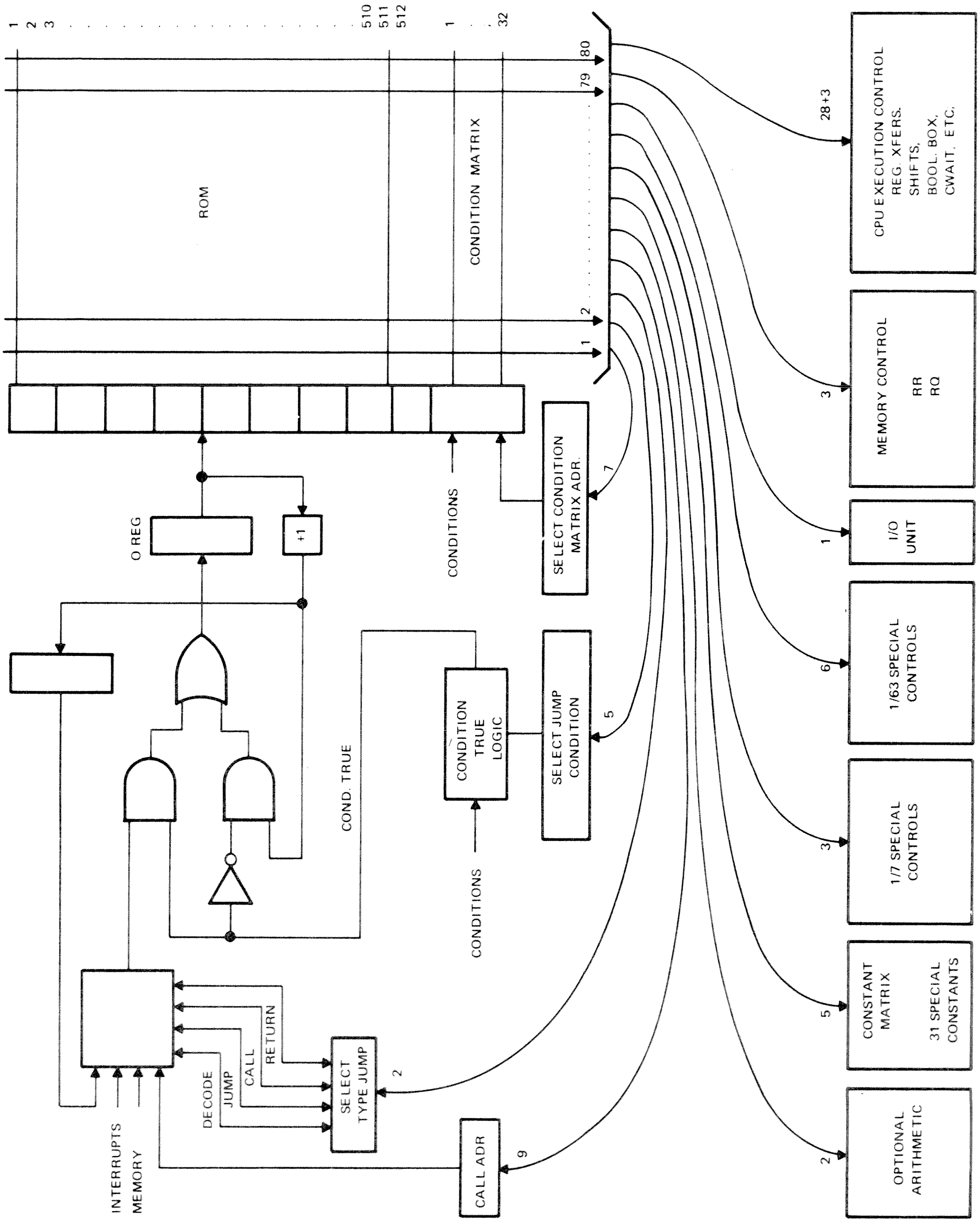


*Figure 1-5. Microprocessor Logic Block Diagram*

of an 80-bit word. For example if a microcode location is addressed that requires transferring the data on the G bus to the W bus and loading the data into the address S register, the ROM lines will generate bit 1 (TGW) and bit 66 (LRO). In order for the ROM signals to be generated, the signals must be enabled by the condition matrix. The condition matrix enables the signals if the required conditions are true. Microcode operations generally occur sequentially (200 to 201 to 202, etc.); however, an interrupt, call or jump causes the microcode logic to call the new address. When this occurs, the address in the O register is incremented by one and stored in the OS register, at the end of the transfer operation the microcode logic issues a RETURN signal to the control logic. On receiving the signal, the control logic loads the OS register back into the O register to continue where it left off prior to the transfer. In order for I/O interrupts to occur, the ROM lines must generate the allow interrupt (AINT) bit 31. Signals for other functions such as special controls, constants, optional arithmetic, etc., have bit numbers as indicated in Figure 1-5. All CPU transfer, arithmetic, and logic functions, once they have been enabled by the microcontroller signals, occur on clock pulses, which are also generated by the microcontroller. Bits 32, 33 and 34 (CKCTL0-2) of the ROM 80-bit word provide the required clock pulses. Two other bits are used to inhibit the clock when more time is required to load or unload memory. These are bits 27 (CWAIT) and bit 28 (MWAIT).

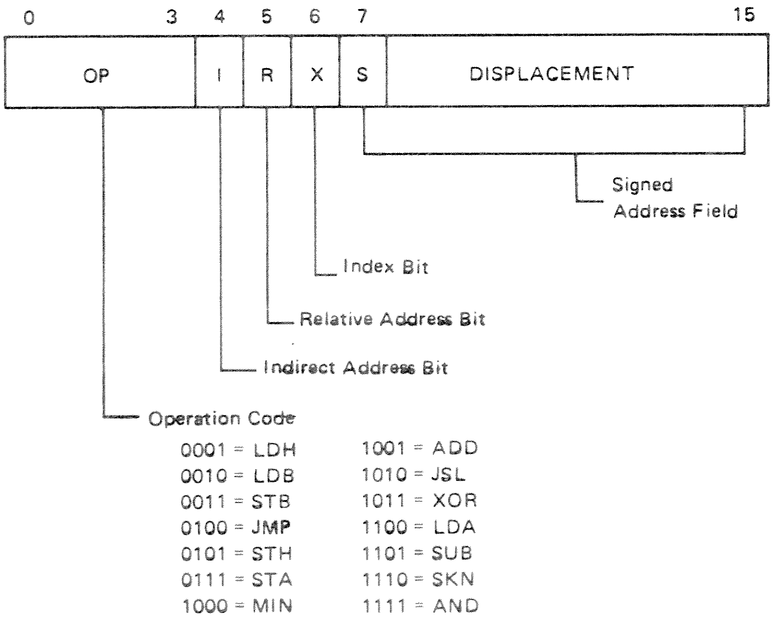
## FORMATS

### INSTRUCTION FORMATS

The three types of formats used in forming instructions are (1) Basic, (2) Literal, and (3) Extended Op Code. A description of each of these formats and their variations is given below.

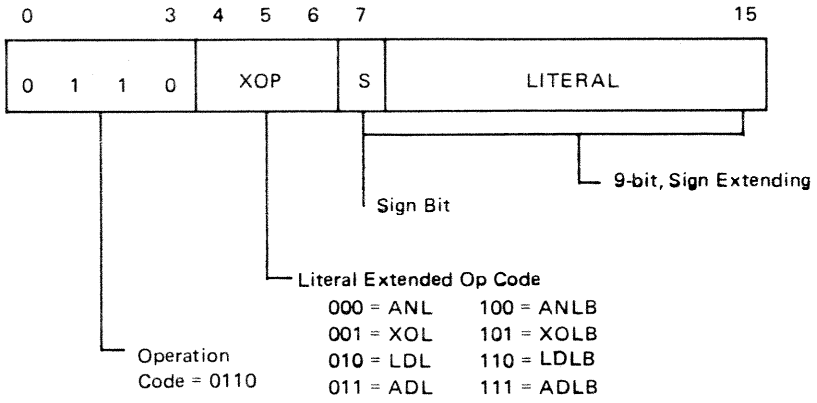
1. **Basic Format**

The most commonly used instructions for memory referencing operations are those written in the basic format. (Refer to "Addressing" for the various modes available.)



## 2. Literal Format

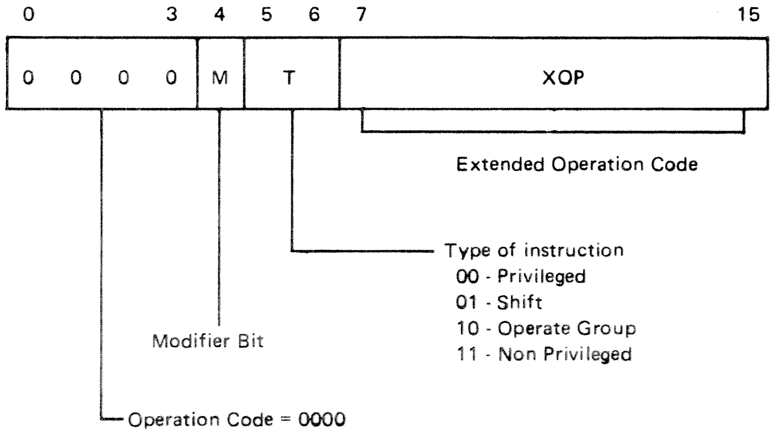
This format applies to all literal instructions, such as LDL (Load A, Literal) and LDLB (Load B, Literal). Literal instructions are assigned an operation code of 0110 and follow the format below.



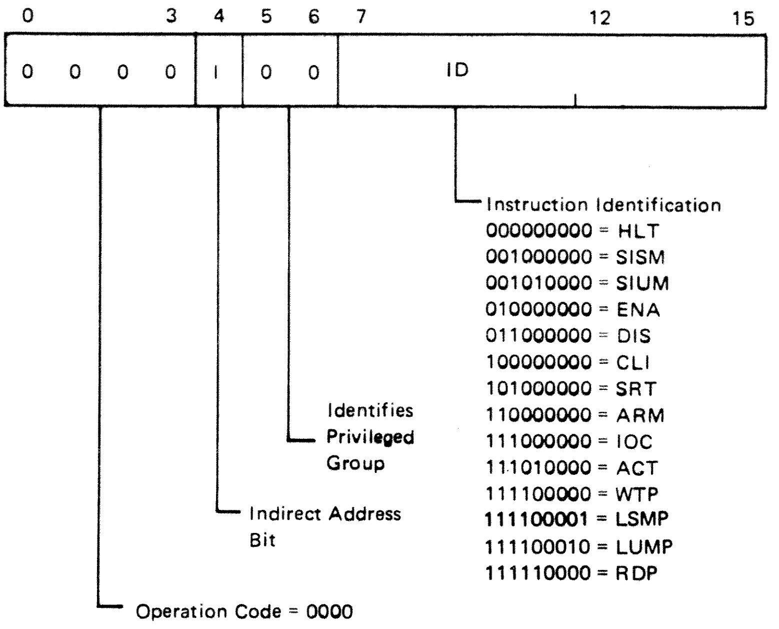
The least significant nine bits of the literal instruction become a 16-bit operand when the sign bit (bit 7) is propagated; i.e., locations 0 through 6 are made equal to bit 7.

## 3. Extended Op Code Format

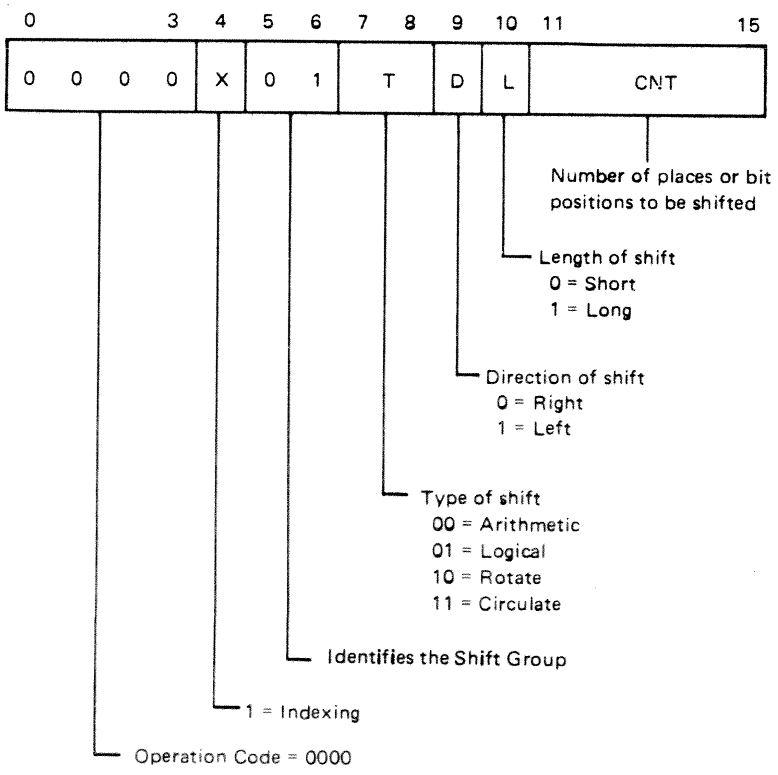
This format is used for instructions which fall into the categories of (a) Privileged, (b) Shift, (c) Operate, or (d) Non-Privileged. The general Extended Op Code format is shown below and is followed by a breakdown of each of the above categories.



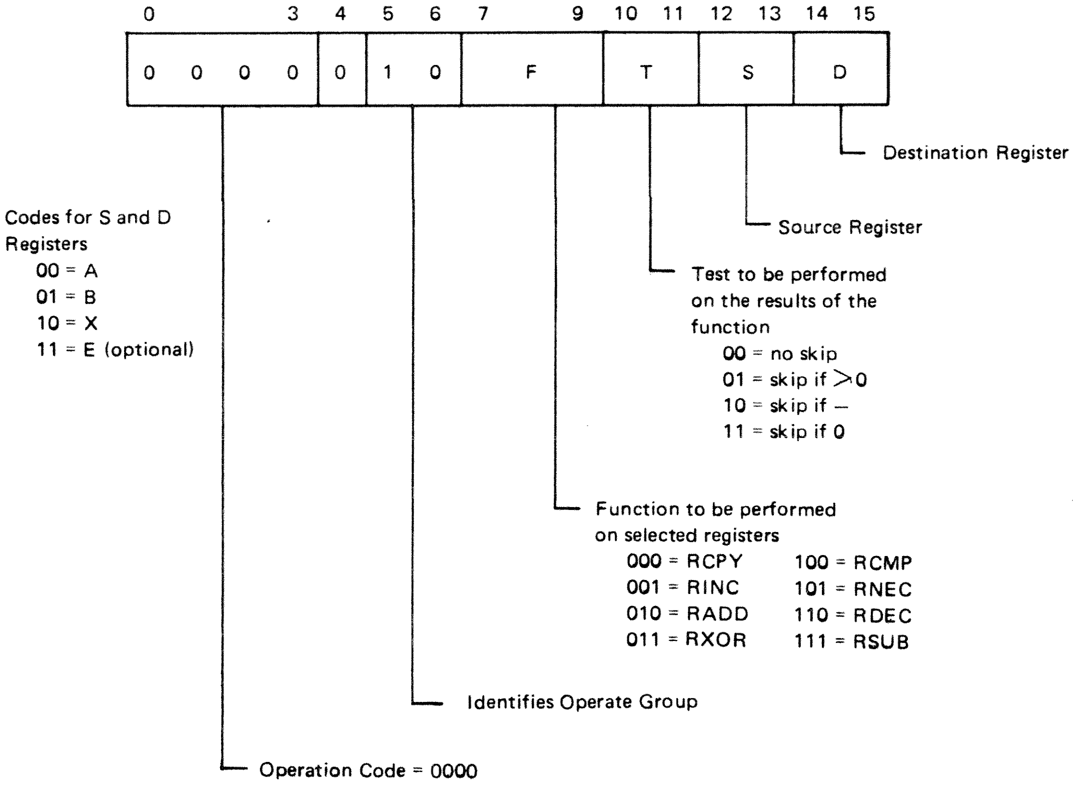
## a. Privileged



## b. Shift



## c. Operate



Destination Register

Source Register

Test to be performed on the results of the function

| | |
|---|---|
| 00 | = no skip |
| 01 | = skip if >0 |
| 10 | = skip if – |
| 11 | = skip if 0 |

Function to be performed on selected registers

| | |
|---|---|
| 000 = RCPY | 100 = RCMP |
| 001 = RINC | 101 = RNEC |
| 010 = RADD | 110 = RDEC |
| 011 = RXOR | 111 = RSUB |

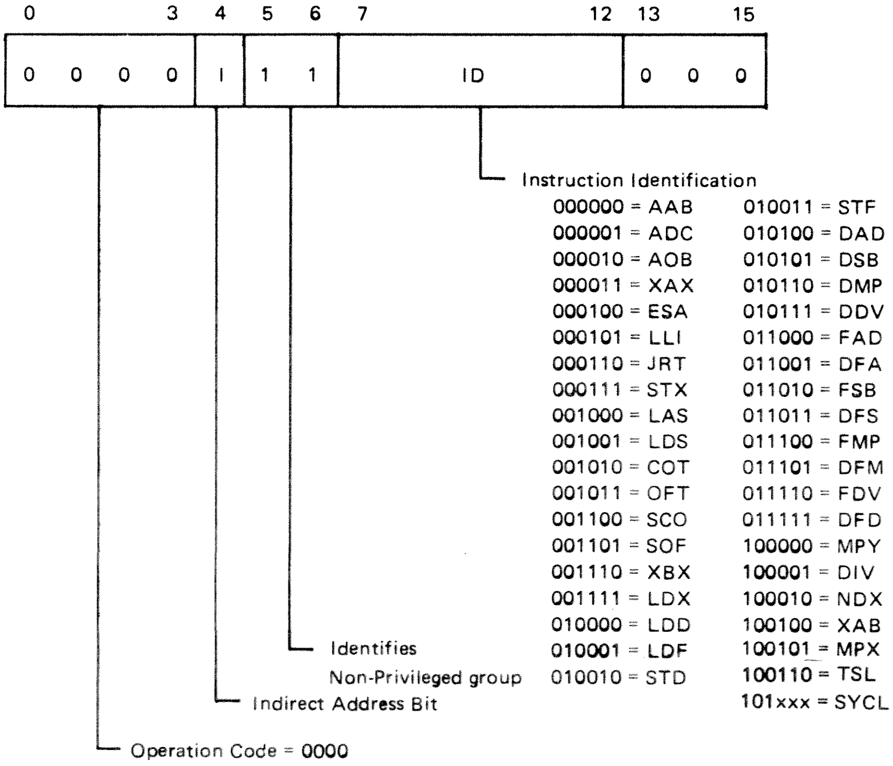
Identifies Operate Group

Operation Code = 0000

Codes for S and D Registers

| | |
|---|---|
| 00 | = A |
| 01 | = B |
| 10 | = X |
| 11 | = E (optional) |

## d. Non-Privileged



Instruction Identification

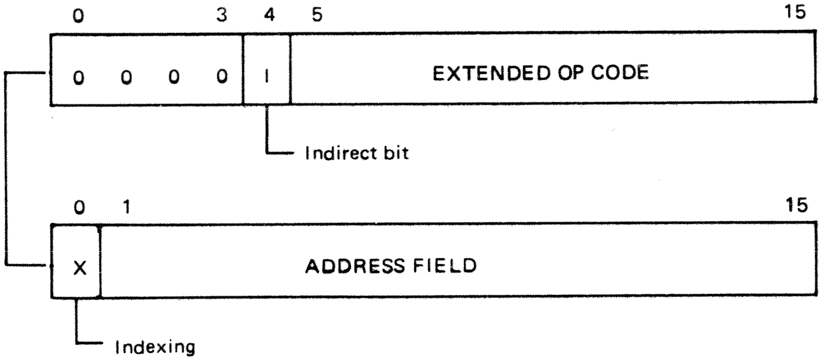
| | |
|---|---|
| 000000 = AAB | 010011 = STF |
| 000001 = ADC | 010100 = DAD |
| 000010 = AOB | 010101 = DSB |
| 000011 = XAX | 010110 = DMP |
| 000100 = ESA | 010111 = DDV |
| 000101 = LLI | 011000 = FAD |
| 000110 = JRT | 011001 = DFA |
| 000111 = STX | 011010 = FSB |
| 001000 = LAS | 011011 = DFS |
| 001001 = LDS | 011100 = FMP |
| 001010 = COT | 011101 = DFM |
| 001011 = OFT | 011110 = FDV |
| 001100 = SCO | 011111 = DFD |
| 001101 = SOF | 100000 = MPY |
| 001110 = XBX | 100001 = DIV |
| 001111 = LDX | 100010 = NDX |
| 010000 = LDD | 100100 = XAB |
| 010001 = LDF | 100101 = MPX |
| 010010 = STD | 100110 = TSL |
| | 101xxx = SYCL |

Identifies Non-Privileged group

Indirect Address Bit

Operation Code = 0000

### Double Word



Certain instructions require two words. The first word is in the Extended Op Code format and the second has the format of an indirect address. There is no relative bit, because the 15-bit address of the second word makes relative addressing unnecessary.

The instructions may be indexed and indirectly addressed. Setting bit 0 of the second word specifies pre-indexing. If indirect addressing (bit 4) is specified, then the second word becomes a pointer (which may be indexed) and the indirect address will be post-indexed if bit 0 of the indirectly addressed location is also set.

Depending on the type of instruction, the second word will contain an address, a pointer, or data. For instance:

1. The optional arithmetic instructions require address information in the second word.
2. Input/output control commands will contain control information for the channels and devices unless indirect addressing is specified.

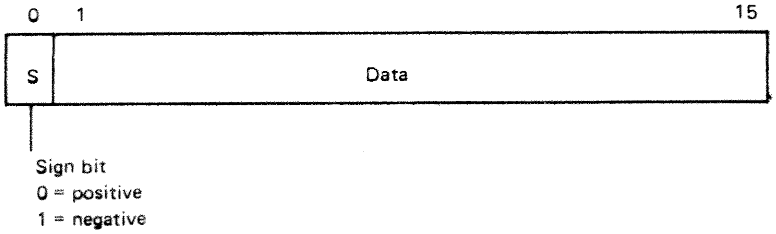
The address portion of this type of instruction and the word to which it points are assembled in PAR statements. One level of indirect addressing and two levels of indexing may be specified.

## DATA FORMATS

Four types of formats are used to represent numerical data: (1) Integer, (2) Double Precision, (3) Floating Point (Short), and (4) Double Floating Point (Long).

1. **Integer**

The basic data format is a 16-bit binary integer with the sign located in bit 0.

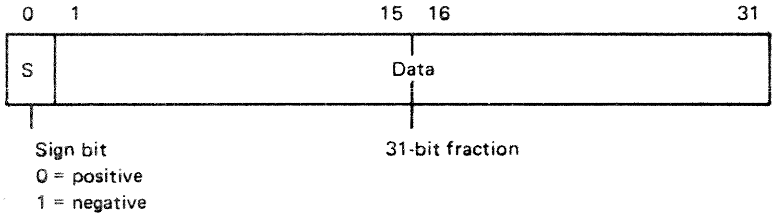


The number represented is defined as a binary or a hexadecimal integer, with bit 1 being the most significant position and bit 15 the least significant.

The maximum range of signed integers which may be represented by a single word is $-8000_{16} \leqslant i \leqslant 7FFF_{16}$ (decimal: $-32768 \leqslant i \leqslant +32767$). Negative quantities are expressed in two's complement form. (The two's complement of a number is obtained by inverting each bit of the binary number and adding one.)

## 2. Double Precision

Double precision arithmetic utilizes two machine words to represent a 31-bit, signed binary fraction with the following format:

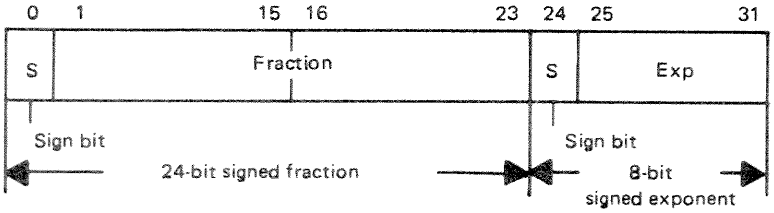


Numbers up to $2^{31}$ may be represented in this format.

The minimum range is $-8000,0000_{16} \leqslant i \leqslant 7FFF,FFFF_{16}$ (decimal: $-1.0000000000 \leqslant i \leqslant 0.9999999996$). Negative double precision numbers are represented in two's complement form.

## 3. Floating Point (Short)

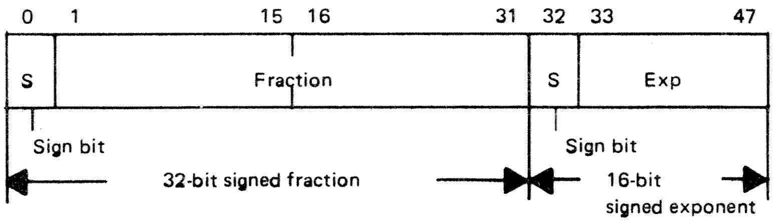
The short floating point format is one of two optional formats available for floating point numbers on the SCC 4700. The short format uses two machine words to represent the floating point number:



The fraction occupies bits 0-23 and the exponent uses bits 24-31. The radix point of the fraction is assumed to be immediately to the left of the high-order fraction digit (right of the sign bit).

### 4. Double Floating Point (Long)

The double floating point format is the second optional format available for expressing a floating point number on the SCC 4700. The long format utilizes three machine words in the following format:



The fraction occupies bits 0-32 and the exponent uses bits 32-47. The radix point of the fraction is assumed to be immediately to the left of the high-order fraction digit (right of the sign bit).

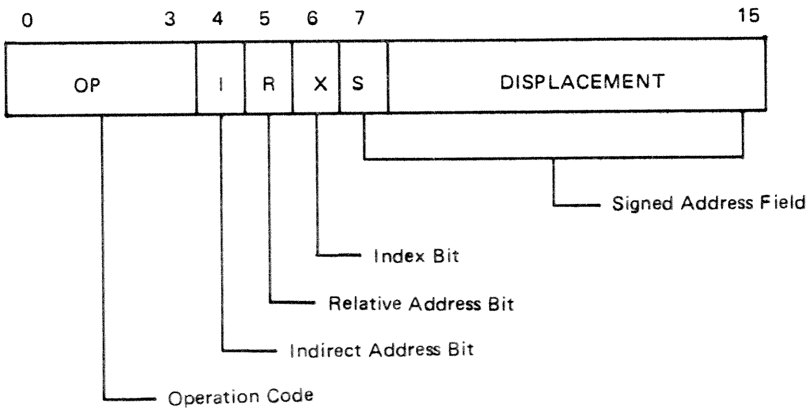
## ADDRESSING

Address modification in the SCC 4700 is based on two concepts:

**Primary Address:** The intermediate address which is determined before indirect addressing and post-indexing are applied. It becomes the effective address if indirect addressing is not applied.

**Effective Address:** The final address which is formed after all address modification and indexing have been performed.

### Modes of Addressing



There are five possible modes available for instructions of basic format, each of which results in a different effective address when implemented. They may be used either singly or in combination. They are:

1. **Direct (Bit 5 = 0)**

   The primary address is determined by the address field of the instruction. In the direct mode, the primary address always refers to the first 512 locations of memory, unless indexed.

2. **Relative (Bit 5 = 1)**

   The primary address is the sum of the address field with sign of the instruction extended and the contents of the location counter; i.e., $(L) \pm (INS)_{8-15}$. (For byte addressing: $2\,(L) \pm (INS)_{8-15}$.)

3. **Primary Indexed (Bit 6 = 1)**

   Indexing may be applied to the primary address to form the primary indexed address.

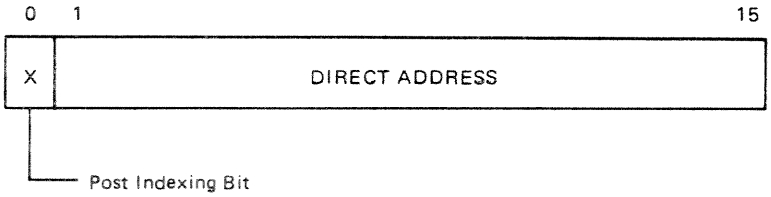
   Direct Indexed – The index register becomes a base register and the effective address is $(X) + (INS)_{7-15}$.

   Relative Indexed – The index register is added to the relative address; i.e., $[\,(L) \pm (INS)_{8-15}\,] + (X)$. (For byte addressing: $[\,2\,(L) \pm (INS)_{8-15}\,] + (X)$.)

4. **Indirect (Bit 4 = 1)**

   The indirect address bit is always applied after the contents of the primary address have been obtained. If the indirect address bit is zero, the contents of the location specified by the primary address are used as the operand of the instruction.

   If the indirect address bit is a one, the contents of the location specified by the primary address is interpreted not as an operand, but as a 15-bit operand address. (Bit 0 of the indirectly addressed location is tested for post-indexing). Indirect addressing requires one additional memory cycle (920 nanoseconds) on all instructions.



   If the primary address of a basic instruction is the current location plus one, the location counter is incremented to skip (L+2) the next word in the instruction sequence. In this way, both operands and full addresses may be included "in-line". This indirect address technique makes possible addressing up to 32K ($7FFF_{16}$).

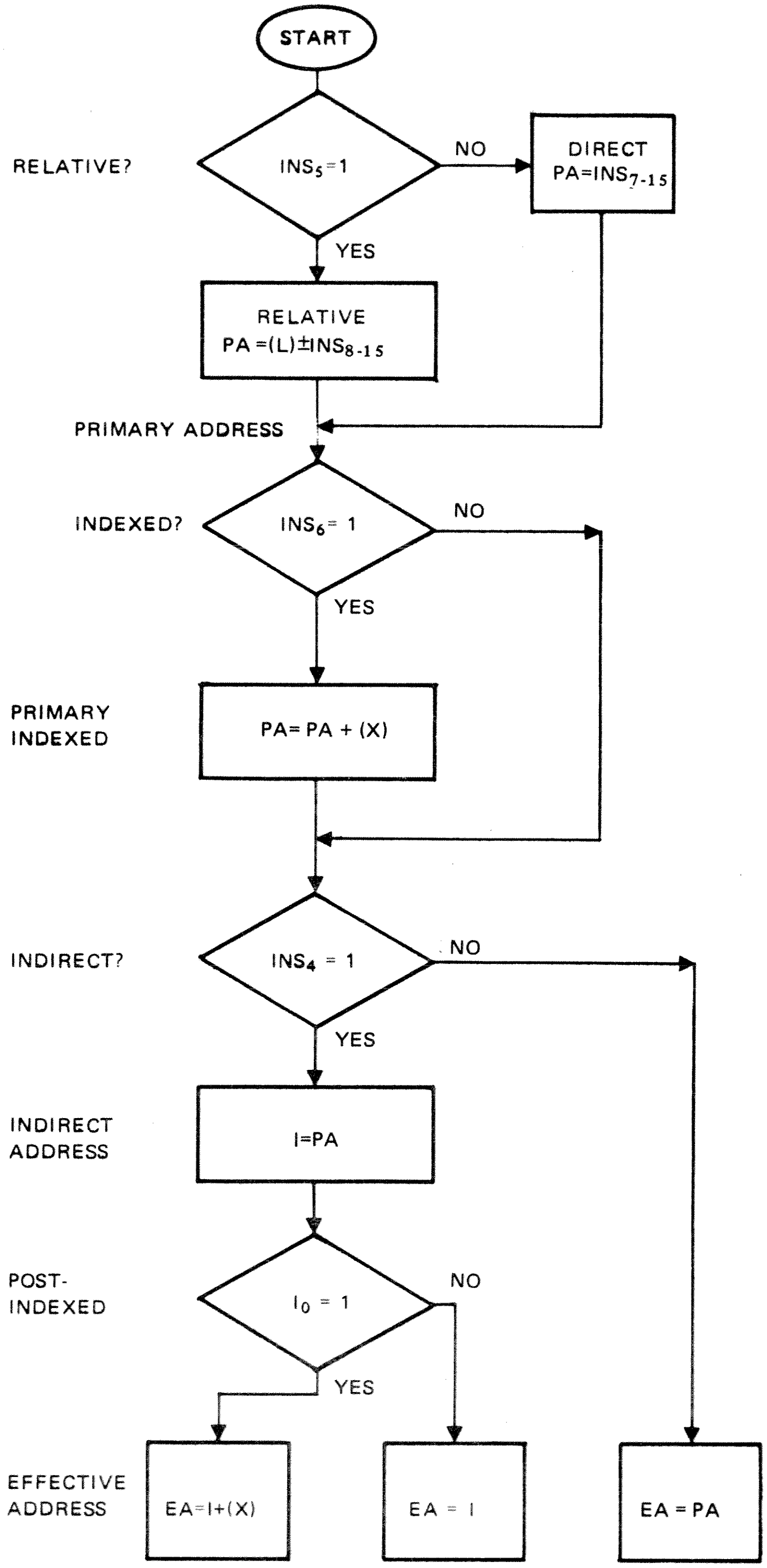


*Figure 1-6. Basic Addressing Mode Flow Diagram*

5. **Post-Indexed (Bit 0 = 1)**

In this mode, bit 0 is checked after the contents of the indirectly addressed location specified by the basic instruction are obtained. If bit 0 is equal to one, the contents of the index register are added to the other 15 bits to form the effective operand address.

Figure 1-6 illustrates the flow of an instruction through the basic modes of addressing.

**Byte Addressing**

Address modification of instructions that are byte addressable (such as LDH, Load Halfword) is accomplished in the same manner as word-oriented instructions with the following exceptions:

1. Post-indexing is not permitted.
2. The range (in the number of words) is half that of word-oriented instructions unless indirect addresses are used. In this case, the full 16 bits are used as the byte address.
3. The location counter is treated as a word address while the index register is considered a byte address.

All conversion of the byte address to align the data in the proper half of the word is performed automatically by the CPU.

The even numbered byte locations will be contained in the most significant eight bits of each word. For example:

| Location | Byte Address | |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 2 | 3 |
| 2 | 4 | 5 |

Figure 1-7 is a flow diagram of the halfword addressing mode.

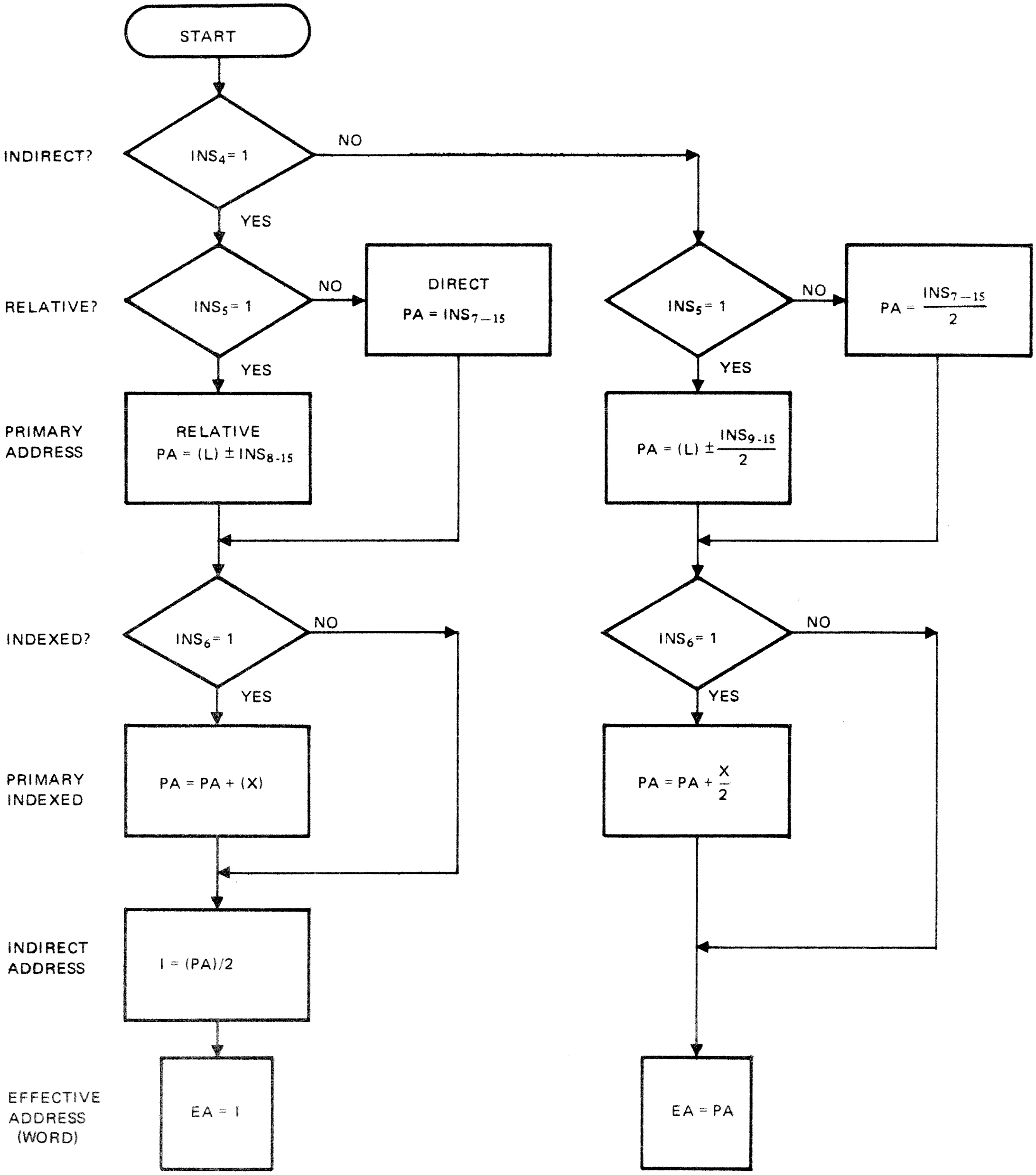


*Figure 1-7. Halfword Addressing Mode Flow Diagram*

## CONTROL AND DISPLAY CONSOLE

The control and display console is an operator interface station used to control the computer system. The console front panel provides the required pushbuttons, rotary switches, register displays and status indicator lights as shown in Figure 1-8. There are 64 colored display lamps, 31 pushbutton bit switches, one keylock and one register select switch. The following list identifies the displays and switches.

Displays (left to right) (top to bottom) — Description

ST — This row displays CPU status conditions as follows:

POWER – Green light indicates power-on condition.

MPE – Red light indicates memory parity error in data transferred to/from memory.

HALT – Red light indicates CPU is in HALT mode.

OVF – Orange light indicates a register overflow condition has occurred.

CRO – Orange light indicates an arithmetic carryout condition has occurred.

L — 16-bit row of lamps display the contents of the location counter. Yellow light indicates that a bit is set (ON=1).

REG — 16-bit row of white light indicators used in conjunction with the REG SELECT switch. When the DSPY (display) button is pushed, the contents of the register indicated by the REG SELECT switch (M, X, A, B or E) are displayed (ON=1).

SW — 16-bit row of orange light indicators displays the contents loaded into the register by the operator. The operator depresses the 0 through 15 pushbuttons, as desired. The display indicates the bits that are set (ON=1).

SW CLEAR — A pushbutton switch used to reset the 16-bits of the switch register (RESET=OFF=0).

0-15 — Sixteen pushbuttons used to set the corresponding bit (Set=1).

POWER OFF-ON-SW-LOCK — A four-position rotary key switch used as follows:

OFF – All system power supplies off.

ON – All power supplies on.

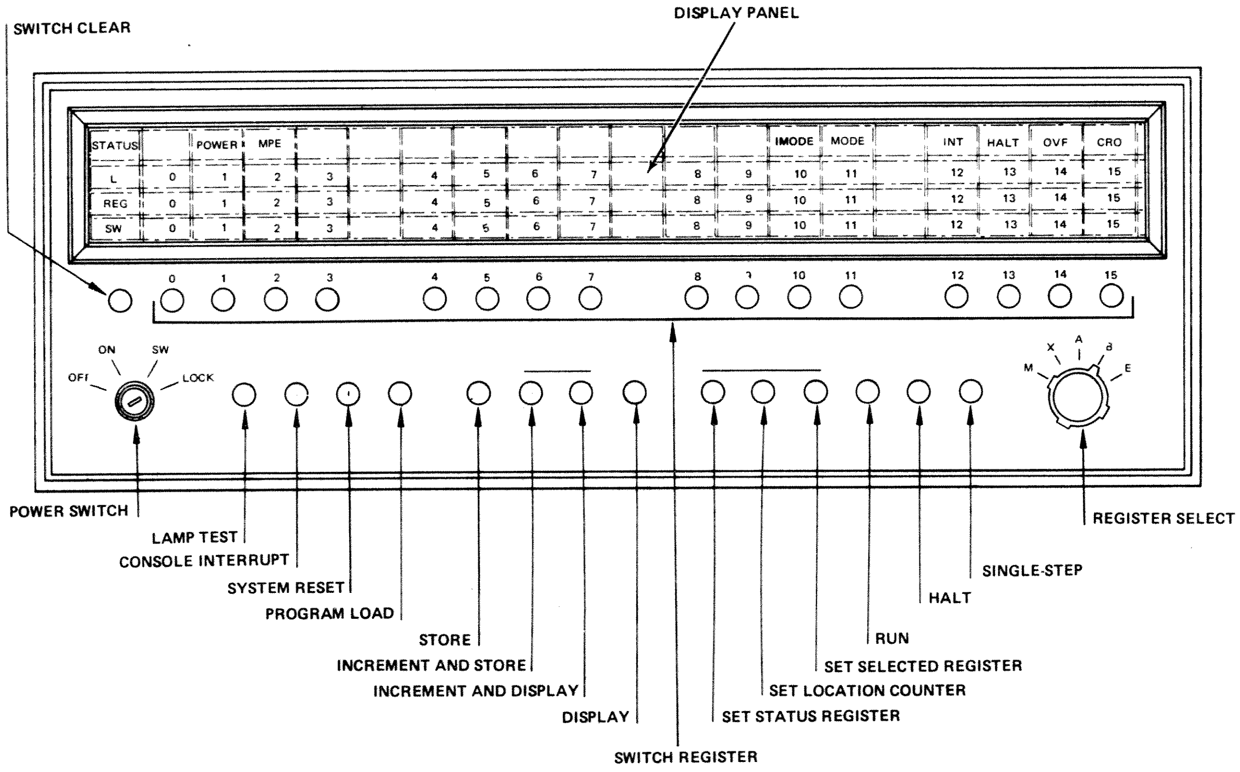


*Figure 1-8. Control Console Front Panel*

POWER (cont'd)

SW – In this position, the SW CLEAR, 0-15, LAMP TEST and CONSOLE INTER switches are operative in the run or halt mode and data may be entered into the switch register for program modification and control; however, registers and memory locations can not be altered or examined from the control console.

LOCK – Power is on and all console switches are inoperative except LAMP TEST.

LAMP TEST

A pushbutton switch turns all indicator lamps on regardless of state when depressed.

CONSOLE INTER

Console interrupt pushbutton switch causes a system trap to location four in reserved core memory. If CPU is in the halt mode when the switch is depressed, the CPU will transfer to the run mode.

SYS RESET

System reset pushbutton switch. Operative only when POWER switch is in ON mode. Resets all indicators and switch settings. Operation of the switch generates the master clear signal to initialize the CPU and all channel interrupts.

PROGRAM LOAD

Program load pushbutton switch. Operative only when POWER switch is in ON mode. Programs may be loaded from any device.

STORE

Store pushbutton switch. Operative only when POWER switch is in ON position, disabled in the run mode. Stores data from the switch register into the address specified by the location register.

INCREMENT STORE

Increment and store pushbutton. Operative only when POWER switch is in ON position, disabled in the run mode. When operated, the location register is incremented by one (L+1) and then the contents of the switch register are stored into the address specified by the new address in the location register.

INCREMENT DISPLAY

Increment and display pushbutton. Operative only when POWER switch is in ON position, disabled in the run mode. When operated, the L register is incremented by one (L+1) and then the contents of memory location specified by the new address in the location counter are transferred to the M register and can be displayed selecting the M register on the REG SELECT switch.

DISPLAY

Display pushbutton switch. Operative only when the POWER switch is in the ON position, disabled in the run mode. When operated, the contents of the memory address specified in the location register are transferred to the M register and can be displayed by selecting the M register with the REG SELECT switch.

SET STATUS

Set status pushbutton switch. Operative only when the POWER switch is in the ON position, disabled in the run mode. When operated, the contents of the switch register are transferred to the corresponding bit in the status register except for HALT which can be set only by the HALT pushbutton or reset by the SYS RESET pushbutton.

SET L

Set location register pushbutton switch. Operative only when the POWER switch is in the ON position, disabled in the run mode. When operated, the contents of the switch register are transferred to the location register.

SET REG

Set register pushbutton switch. Operative only when the POWER switch is in the ON position, disabled in the run mode. When operated, transfers the contents of the switch register to the register selected by the REG SELECT switch.

RUN

Pushbutton switch used to reset HALT indicator and put the machine in the run mode. Operative only when the POWER switch is in the ON position.

HALT

Pushbutton switch used to stop program execution, set the HALT indicator, and update the display registers. Only the CPU is halted, I/O devices continue to run. The CPU may be returned to the run mode by interrupts as listed in the Interrupt Subsystem Section.

REG SELECT

Register select switch used to select register M, X, A, B or E to be displayed in the REG row of indicators, or to select the register into which data will be entered by the SET REG button. Operative only when the POWER switch is in the ON position.

The Control and Display console is 8 inches high, 22-5/16 inches wide and 5-1/4 inches deep. The console is mounted in the front of the basic machine. It is inserted into the cabinet at a 10-degree slant above a convenient 16-inch by 26-inch writing shelf. An optional remote console is available. This optional remote console can be connected to the CPU by up to 30 feet of cable and is available with a Selectric typewriter for operator messages and commands.

## SYSTEM INTERFACES

System interfaces are described in subsequent sections of this manual. These sections include all input/output interfaces such as the parallel I/O, channel I/O, and interrupt subsystem interfaces. The channel and interrupt sections describe arm/disarm, enable/disable and priority logic used in the system. System interfaces, required for expanded system configuration such as memory gate and digital multiplex interfaces, are also described in the following sections.

# SECTION II

# INPUT/OUTPUT INSTRUCTIONS

The capabilities of the SCC 4700 input/output system are:

1. Data chaining which permits scatter read, gather write techniques
2. Mixed mode operations with different devices on same multiplexor channel
3. Servicing of multiple slow devices on same channel
4. High data transfer rates for fast devices
5. Full word or byte data transfer.

The I/O system uses one to four channel units to communicate with memory either directly or through the Central Processing Unit. The channel units may be multiplexor and/or selector channels, each of which can interface up to 64 device controllers.

Data transfers in the I/O system may be byte- or word-oriented. Byte transfers utilize either the multiplexor or selector channel and may be in either single byte or block transfer mode. Full word data transfer is performed through the parallel I/O interface and is controlled by the Read Parallel (RDP) and Write Parallel (WTP) instructions.

I/O control operations are separated into the following categories to direct the functions of three kinds of control equipment:

*Instructions* to be executed by the CPU

*Commands* to be executed by a channel

*Orders* to be executed by a device.

## INPUT/OUTPUT CONTROL

The SCC 4700 input/output system utilizes four instructions: three for parallel I/O and one for channel I/O.

### Parallel I/O Instructions

A 16-bit, word-oriented, parallel I/O interface is used to read and control all devices on the parallel I/O bus, such as interrupts and special devices.

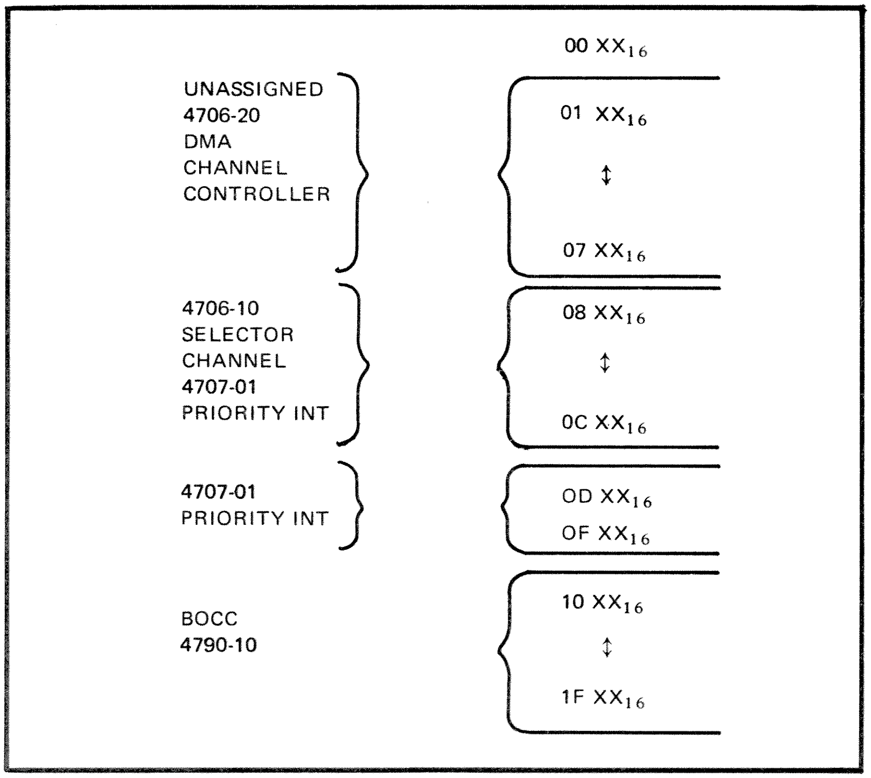
The device to be used is determined by the address data contained in the Activate (ACT) instruction. Data transfer between a device and the accumulator is accomplished by the Read Parallel I/O (RDP) and Write Parallel I/O (WTP) instructions.

## ACT ACTIVATE PARALLEL I/O 1.84 $\mu$sec.

| 0 |  |  |  | 4 |  |  | 7 | 8 | 9 |  | 11 |  |  |  | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | I | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| CONTROL INFORMATION |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

ACT is a non-interruptable, privileged instruction used to activate any I/O device which utilizes the parallel I/O interface. It is necessary to give an ACT instruction if more than one device is connected to the parallel interface.

ACT is a double word instruction, the first word of which is in the Extended Op Code format. The second word contains the control information or the address of the control information. This double word instruction must occupy contiguous memory locations but may start on even or odd numbered locations. See Figure 2-1 for typical second word formats.



XX REPRESENTS THE LOWER 8 BITS OF THE SECOND WORD AND ARE FOR USE BY AN SCC I/O DEVICE OR CUSTOMER EQUIPMENT.

*Figure 2-1. Typical Second Word Format of ACT Instruction*

The interpretation of the control information in the second word depends on the device and the user's system requirements. If bit 4 of ACT is set (=1), the second word becomes a pointer to the control information. This pointer is in the indirect address format and may be indexed.

| | |
|---|---|
| X | ADDRESS OF CONTROL INFORMATION |

2nd word if I = 1

The I/O control information is then output directly from memory and the location counter is incremented by one.

WTP **WRITE PARALLEL I/O** **1.95 μsec.**

| 0 | | | | | | | 7 | 8 | 9 | 10 | | | | | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

If device ready: (A) → Device, (L) + 2 → L.

WTP is a privileged instruction which transfers a 16-bit data word from the accumulator to the active device on the parallel I/O interface.

If the receiving device is ready, transfer is performed and the location counter is incremented by two.

If the device is not ready when tested by the WTP, the transfer is not performed and the machine proceeds to the next instruction after incrementing the location counter by one.

RDP **READ PARALLEL I/O** **1.95 μsec.**

| 0 | | | | | | | 7 | 8 | 9 | 10 | 11 | | | | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

If device ready: (Device) → A, (L) + 2 → L.

RDP is a privileged instruction which transfers a 16-bit data word from the active device on the parallel I/O interface to the accumulator.

The transfer occurs only if the device is ready; the location counter is then incremented by two.

If the device is not ready when tested by RDP, the transfer is not performed and the machine proceeds to the next instruction after incrementing the location counter by one.

## Channel I/O Instruction

The operation of the 8-bit, byte-oriented channels are controlled by an Input/Output Control (IOC) instruction.

### IOC INPUT/OUTPUT CONTROL *

| 0 |  |  |  | 4 |  |  | 7 | 8 | 9 |  |  |  |  |  | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | I | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| CHANNEL COMMAND |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

IOC is a privileged instruction which transmits the control information (channel command) required by either the multiplexor or selector channel controller.

IOC is a double word instruction, the first word of which is in the Extended Op Code format. The second word contains the channel command or the address of the channel command, depending on the value of bit 4.

If bit 4 is set (=1), the second word becomes a pointer to the channel command. This pointer is in the indirect address fɔrmat and may be indexed.

|  |  |
|---|---|
| X | ADDRESS OF CHANNEL COMMAND |

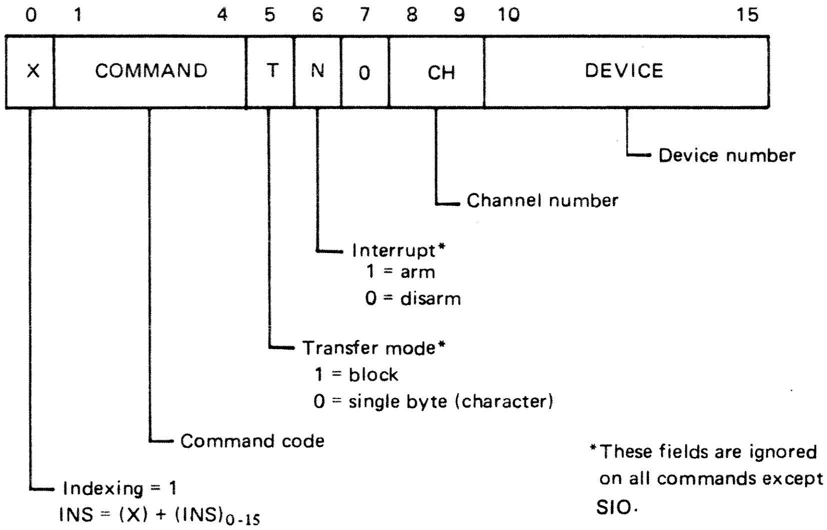
2nd word if I = 1

The channel command is then output directly to the channel controller, which changes it to a sequence of signals acceptable to the control unit. (Refer to the section "Channel Commands" for commands available to the programmer.)

*The IOC is normally executed in 2.87 $\mu$sec (except Start I/O), providing an acknowledge from the device is received immediately; otherwise, it delays and checks for an acknowledge at 1.35 $\mu$sec intervals. If an acknowledge is not transmitted after 10 $\mu$sec, the "watchdog timer" (not the device) generates an acknowledge and the instruction proceeds. (An optional feature provides an external interrupt when the "watchdog timer" generates the acknowledge.)

## Channel Commands

The I/O channel commands are sent directly to the channel controller in the following single word format.



*These fields are ignored on all commands except SIO.

*Each channel command must be immediately preceded by an IOC instruction or contained in the location specified by the address of the IOC.*

Device numbers are given in table 2-1.

### HIO HALT I/O*

| 0 | | | | | | | | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CH | DEVICE |

This command causes an immediate termination. The addressed device is halted and the settings of the entire device controller, including the status information, are cleared. HIO inhibits interrupts from the device and may cause information to be lost.

* HIO, TWC, system reset, and local reset all cause termination. A device also terminates on any error requiring operator intervention (such as card jam) or at the end of a record if a data error has been detected.

# TABLE 2-1
## PERIPHERAL DEVICE NUMBERS
## (4700 Series)

| INPUT | | | OUTPUT | | | REMARKS |
|---|---|---|---|---|---|---|
| OCTAL | HEXA-DECIMAL | FUNCTION | OCTAL | HEXA-DECIMAL | FUNCTION | STANDARD DEVICES |
| 00 | 00 | | 01 | 01 | | |
| 02 | 02 | Read Storage No. 1 | 03 | 03 | Write Storage No. 1 | Drum or Disk |
| 04 | 04 | Keyboard No. 1 | 05 | 05 | Printer No. 1 | Teletype ASR 33 |
| 06 | 06 | Tape Reader No. 1 | 07 | 07 | Tape Punch No. 1 | |
| 10 | 08 | Byte Input No. 1 | 11 | 09 | Byte Output No. 1 | |
| 12 | 0A | Read Storage No. 2 | 13 | 0B | Write Storage No. 2 | Drum or Disk |
| 14 | 0C | Keyboard No. 2 | 15 | 0D | Printer No. 2 | Teletype ASR 35 |
| 16 | 0E | Tape Reader No. 2 | 17 | 0F | Tape Punch No. 2 | |
| 20 | 10 | | 21 | 11 | Line Printer | |
| 22 | 12 | | 23 | 13 | X-Y Plotter No. 1 | |
| 24 | 14 | Keyboard No. 3 | 25 | 15 | Printer No. 3 | IBM Selectric |
| 26 | 16 | Card Reader | 27 | 17 | Card Punch | |
| 30 | 18 | Byte Input No. 2 | 31 | 19 | Byte Output No. 2 | |
| 32 | 1A | | 33 | 1B | X-Y Plotter No. 2 | |
| 34 | 1C | | 35 | 1D | | |
| 36 | 1E | | 37 | 1F | | |
| 40 | 20 | Incremental Tape Read A | 41 | 21 | Incremental Tape Write A | |
| 42 | 22 | Analog - Digital Converter (in) | 43 | 23 | Digital - Analog Converter (out) | |
| 44 | 24 | System in A No. 1 | 45 | 25 | System Out A No. 1 | Unique Device (Controller Oriented) |
| 46 | 26 | System in B No. 1 | 47 | 27 | System Out B No. 1 | |
| 50 | 28 | Byte Input No. 3 | 51 | 29 | Byte Output No. 3 | |
| 52 | 2A | | 53 | 2B | | |
| 54 | 2C | System in A No. 2 | 55 | 2D | System Out A No. 2 | Unique Device (Controller Oriented) |
| 56 | 2E | System in B No. 2 | 57 | 2F | System Out B No. 2 | |
| 60 | 30 | Magnetic Tape A No. 1 Read | 61 | 31 | Magnetic Tape A No. 1 Write | |
| 62 | 32 | Magnetic Tape B No. 1 Read | 63 | 33 | Magnetic Tape B No. 1 Write | |
| 64 | 34 | Magnetic Tape C No. 1 Read | 65 | 35 | Magnetic Tape C No. 1 Write | |
| 66 | 36 | Magnetic Tape D No. 1 Read | 67 | 37 | Magnetic Tape D No. 1 Write | |
| 70 | 38 | Magnetic Tape A No. 2 Read | 71 | 39 | Magnetic Tape A No. 2 Write | |
| 72 | 3A | Magnetic Tape B No. 2 Read | 73 | 3B | Magnetic Tape B No. 2 Write | |
| 74 | 3C | Magnetic Tape C No. 2 Read | 75 | 3D | Magnetic Tape C No. 2 Write | |
| 76 | 3E | Magnetic Tape D No. 2 Read | 77 | 3F | Magnetic Tape D No. 2 Write | |

## SIO START I/O

| 0 | 4 | 5 | 6 | | 8 9 | 10 15 |
|---|---|---|---|---|---|---|
| X | 0 0 0 1 | T | N | 0 | CH | DEVICE |

This command causes the device controller to be selected. If bit 5 = 1, SIO initiates channel operation for block transfers by starting automatic data transmission under control of the multiplexor or selector channel. The transfer mode (bit 5) and interrupt arm/disarm (bit 6) are transferred to the controller.

The device must be available when issuing the SIO command.

SIO clears previous status information.

## XMT TRANSMIT

| 0 | 3 | 8 9 | 10 15 |
|---|---|---|---|
| X | 0 0 1 0 0 0 0 | CH | DEVICE |

This command inputs or outputs a character to or from the least significant half of the accumulator (the most significant half is cleared on input). Direction of transfer is determined by the device number for the selected device:

even = input

odd = output

The device must be ready before executing a XMT command. XMT will then reset the ready bit and remove the device interrupt if in single byte mode.

## EOA EXECUTE ORDER IN ACCUMULATOR

| 0 | 3 4 | 8 9 | 10 15 |
|---|---|---|---|
| X | 0 0 1 1 0 0 0 | CH | DEVICE |

This command transmits an order (contained in the least significant half of the A register) to the device controller to initiate the desired operation. The action initiated by the selected device depends on the order format (which varies with each device).

## IDN INPUT DEVICE NUMBER

| 0 | | 2 | | | | | | | | | | | | | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This command is not available to the programmer. It is used by the microcode of the CPU to identify an interrupting device when servicing the multiplexor channel for block transfers.

## OUS OUTPUT UNIT STATUS

| 0 | | 2 | | 4 | | | | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 0 | 1 | 0 | 1 | 0 | 0 | 0 | CH | DEVICE |

This command transmits 8 bits of data from the least significant half of the A register to the status register of the device controller. Bits of the status register may be set only; the device status cannot be reset or cleared by this command. The ready and available bits of the status register are not affected.

## TWC* TERMINATE WHEN COMPLETE

| 0 | | 2 | 3 | 4 | | | | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 0 | 1 | 1 | 1 | 0 | 0 | 0 | CH | DEVICE |

This is the usual command given to halt a device. Devices with predetermined record length will immediately stop data transfer; however, other devices, such as paper tape, may transmit the character currently being read before stopping transfer. When the device completes its current operation, the device and controller are terminated. The device gives a channel interrupt if armed when terminated. The status register remains valid.

## SDR SKIP IF DEVICE IS READY

| 0 | 1 | | | 4 | | | | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 0 | 0 | 1 | 0 | 0 | 0 | CH | DEVICE |

This command tests the status register of the device and, if the device is ready, increments the location counter by two.

* HIO, TWC, system reset, and local reset all cause termination. A device also terminates on any error requiring operator intervention (such as card jam) or at the end of a record if a data error has been detected.

## SDA

### SKIP IF DEVICE IS AVAILABLE

| 0 | 1 |  | 3 |  |  |  |  | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 0 | 1 | 0 | 0 | 0 | 0 | CH | DEVICE |

This command tests the status register of the device and, if the device is available, increments the location counter by two.

## IIU

### INPUT INTERRUPTING UNIT

| 0 | 1 |  | 3 | 4 |  |  |  |  |  |  |  |  |  |  | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This command inputs the device number of the interrupting device into the least significant 6 bits of the index register and clears the high order bits.

## IUS

### INPUT UNIT STATUS

| 0 | 1 | 2 |  |  |  |  |  | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 1 | 0 | 0 | 0 | 0 | 0 | CH | DEVICE |

The command transmits the status register of the device being addressed into the least significant 8 bits of the A register and clears the most significant bits. IUS also clears the interrupt from the device.

## ISB1

### INPUT STATUS BYTE 1

| 0 | 1 | 2 |  | 4 |  |  |  | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 1 | 0 | 1 | 0 | 0 | 0 | CH | DEVICE |

This command transmits additional device status information into the least significant 8 bits of the accumulator and clears the high order bits. The status information varies with each type of device and controller.

## ISB2

### INPUT STATUS BYTE 2

| 0 | 1 | 2 | 3 |  |  |  |  | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 1 | 1 | 0 | 0 | 0 | 0 | CH | DEVICE |

This command transmits another byte of device status information (different from the byte transferred by ISB1) to bits 8-15 of the accumulator and resets bits 0-7. The status information varies with each type of device and controller.

**ISB3**

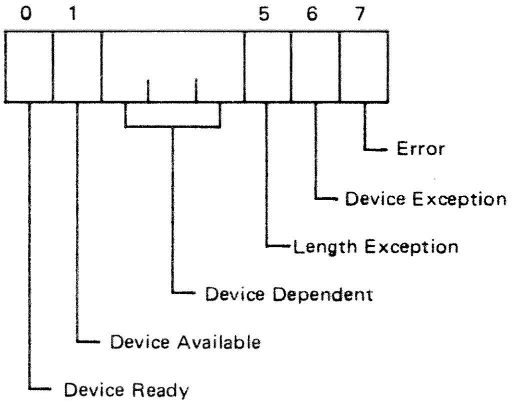
## INPUT STATUS BYTE 3

| 0 | 1 | 2 | 3 | 4 | | | | 8 9 | 10 15 |
|---|---|---|---|---|---|---|---|---|---|
| X | 1 | 1 | 1 | 1 | 0 | 0 | 0 | CH | DEVICE |

This command transmits another byte of device status information (different from the bytes transferred by ISB2 and ISB3) to bits 8-15 of the accumulator and resets bits 0-7. The status information varies with each type of device and controller.

### Device Status

The format of the 8-bit device status register is as follows.



Bit 0 This bit set to a one indicates the device is ready and waiting for data transfer in character mode. If data transfer is in block mode, this bit will cause a transfer request to be generated.

Bit 1 This bit set to a one indicates the device is available for selection and use.

Bits 2-4 Device controller dependent.*

Bit 5 This bit set to a one indicates an error in the anticipated block count or record length.*

Bit 6 This bit set to a one indicates an unusual condition in the device completion, such as tape mark on tape, special characters, end of tape, out of paper, etc.*

Bit 7 This bit set to a one indicates an error condition which may have caused the data to be transferred incorrectly, such as parity error, rate overrun, card jam, etc.*

*Bits 2-7 may be set by channel command OUS (Output Unit Status). The status byte is cleared by channel commands SIO and HIO.

### Device Orders

To initiate an input/output operation by a device, the device controller must transmit control information called a device order to the device in question. Due to the number of optional devices available for the SCC 4700, the codes, format, and interpretation of the device orders are not included in this manual. However, this information is included in the manual concerning the individual peripheral device.

## INPUT/OUTPUT CHANNELS

The input/output system of the SCC 4700 utilizes from one to four data channels. The basic SCC 4700 includes one multiplexor channel and is expandable to include up to four channels in any combination of multiplexor and/or selector channels.

### Multiplexor Channel

The multiplexor channel is under direct control of the Central Processing Unit and uses CPU data and address paths to memory.

This channel is capable of transferring data to and from up to 64 devices on a multiplex or "party-line" basis. Data transfer may be performed in single byte and block modes with each active device controller operating independently; i.e., some device controllers may be engaged in data transfer in the single byte mode, while other device controllers are transferring data in the block mode.

### Selector Channel

Although activated by the Central Processing Unit, the selector channel operates independently of the CPU in providing direct memory access at a high data rate between a device and memory. Up to 64 byte-oriented device controllers may be serviced by the selector channel; however, only one device may be active at one time.

Block data transfer is accomplished through the command and control information sent directly from memory to the channel. When data transfer is in the block mode, "chaining" is permitted; i.e., the selector channel can sequentially execute a series of block transfers.

## INPUT/OUTPUT SYSTEM OPERATION

### Word Transfers

Full word data transfer is performed through the parallel I/O or the Direct Memory Access (DMA) ports.

1. Parallel I/O – The parallel I/O interface is controlled by the Activate Parallel I/O (ACT), Read Parallel (RDP), and Write Parallel (WTP) instructions. (Refer to the section "Parallel I/O Instructions" of this chapter.)

2. DMA Ports – The DMA Port provides a high speed, 16-bit, word-oriented interface between main memory and an external device. Data is routed to and from memory directly between the external device and the memory gating module.

   Since the DMA is designed for use with special device configurations, the reference manual concerning the device in question should be consulted.

**Single Byte Transfers**

Both the multiplexor and selector channels accept data in the single byte (character) mode. There is no difference in the operation of either channel during data transfer; i.e., data rate is the same, multiple devices may be serviced, etc.

Data transfer is accomplished under program control, using the channel commands discussed previously, with data being output or input to or from the low order 8 bits of the accumulator. This mode of channel operation allows great flexibility to the programmer, placing the full capabilities of the channel under his complete control. The following illustrates a typical application of the single byte mode.

The device should first be tested for availability. If it is available, it may be selected with an SIO command and the device interrupt enabled, if operation under interrupt control is prefered. Next, an order is transmitted to the device with an EOA command to initiate the desired action.

If operation is not under interrupt control, the device is then tested to be ready. When it becomes ready, data is transmitted to or from it. This test for ready status and the subsequent data transmission sequence is repeated until the desired number of bytes have been transferred. The device operation is then terminated with a TWC command. The final status for the operation may then be tested when the device again becomes available.

If operation is under interrupt control, the interrupting unit may be determined by using the IIU command. The data is then transmitted after checking the device status if desired. When the data transfer is complete, the device is terminated with a TWC command. The device status is finally tested upon receiving its termination interrupt.

If there are multibyte orders, the first order byte is transmitted with an EOA command followed by a XMT command for each additional order byte. For example:

| M OR S CHANNEL CHARACTER MODE | ACCUMULATOR |
|---|---|
| SIO | |
| EOA | |
| XMT | |
| XMT | MULTIBYTE |
| XMT | ORDERS |
| . . . | . . . |
| XMT | DATA |
| XMT | DATA |
| XMT | DATA |
| . . . | . . . |
| XMT | DATA |

### Block Transfers

The block transfer process to or from memory can be performed through either the multiplexor or selector channel. Transfers of multiple bytes of data through the selector channel occur asynchronously and independently of the Central Processing Unit; however, memory cycles may be stolen by the selector channel if the same memory module is accessed simultaneously.

All block transfers through the multiplexor and selector channel are initiated by execution of the Start I/O channel command. Transfer of data then proceeds automatically until completion, without further attention from the program.

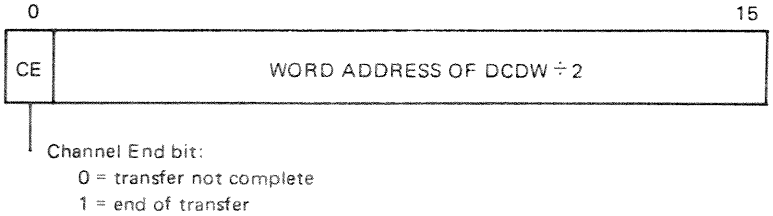
Both channels are programmed in exactly the same way; the only difference between the two channels occur in the manner the hardware handles the block transfer. Those to be considered are:

1. Faster data rate on selector channel.
2. Address and byte count are stored in core memory for the multiplexor channel, rather than in registers as with selector channel.
3. The microcode of the CPU handles the interrupt and data transfer which results in the multiplexor channel stealing some CPU time and memory cycles (the selector channel steals only memory cycles.)
4. Multiple devices can be active on the multiplexor channel at the same time, but only one device at a time can be active on the selector channel.

To perform a block transfer on either the multiplexor or selector channel, certain steps must be accomplished. (Refer to Figure 2-2.)

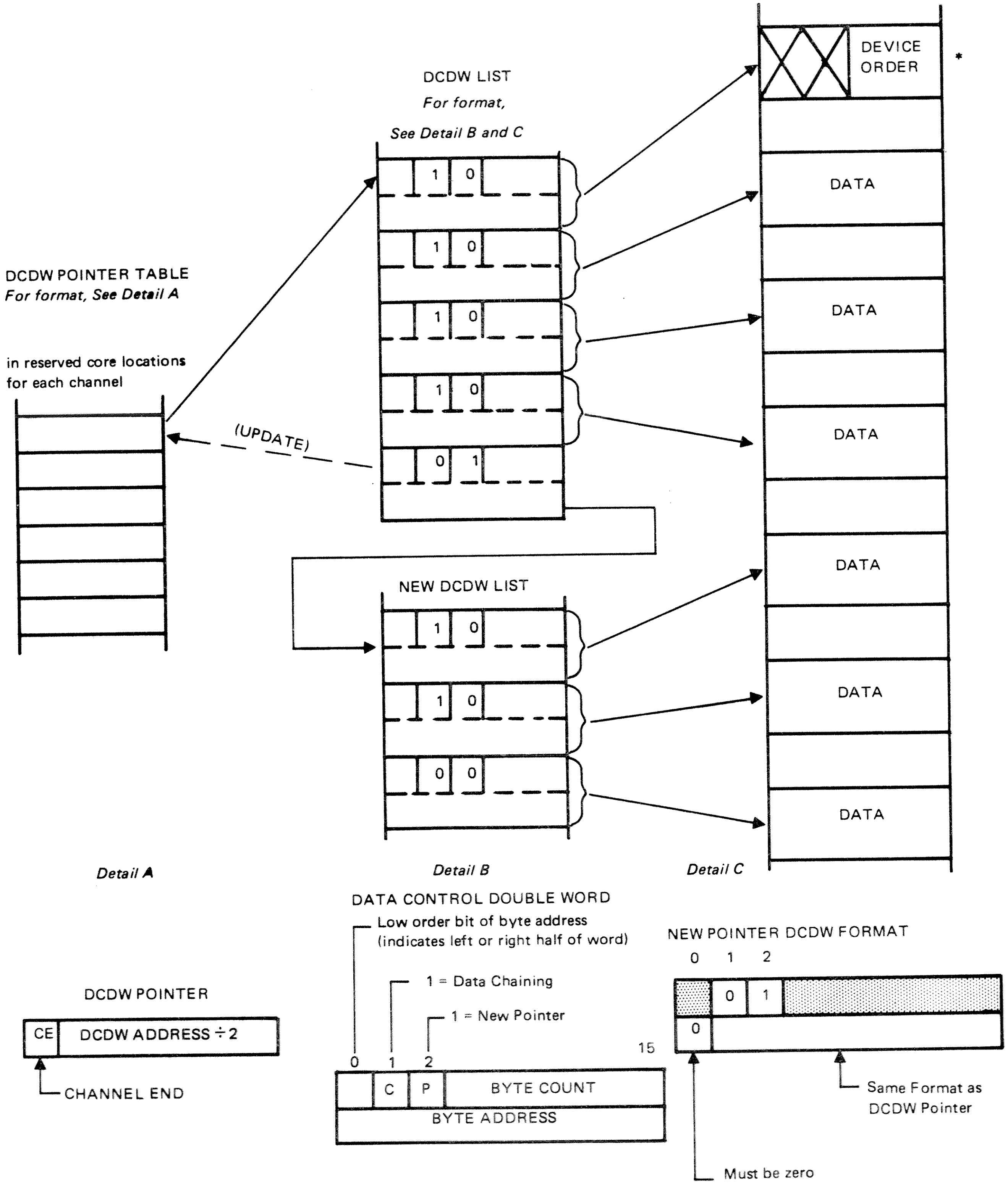
1. An input/output data area is reserved in memory.
2. A DCDW area is reserved in memory.
3. A pointer to the Data Control Double Word (DCDW) controlling this transfer is placed at the proper location in memory. The location for the DCDW pointer is calculated by: the channel number and device number (juxtapositioned as in the channel command) plus $28_{10}$.

   The format of the DCDW Pointer is:



   The entry for the DCDW pointer is obtained by performing a logical right shift of one place on the real core address of the first DCDW.

   The channel end bit of the pointer is set (=1) by the channel when transfer is complete. If the program logic uses this bit to determine completion of transfer, it must be reset before the same pointer is referenced again.



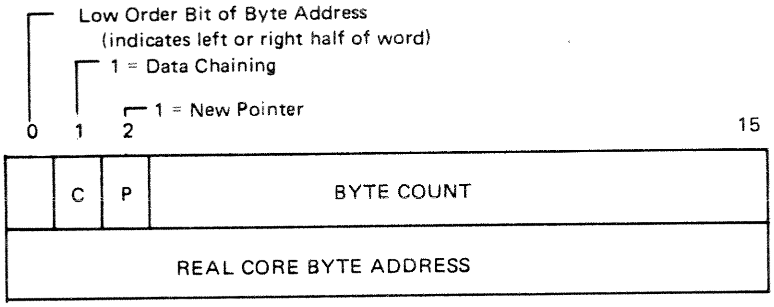
* If the device order is not placed with the data as shown above, separate DCDW's are used. If the device order occupies the first byte in the data field, only one DCDW is necessary.

*Figure 2-2. Typical Block Transfer Operation*

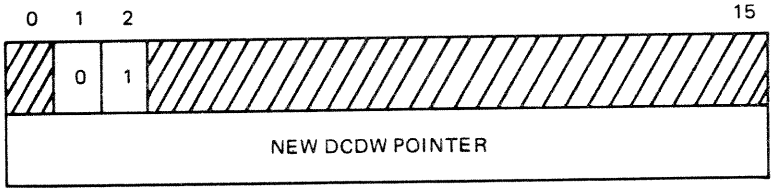
4. DCDW's are entered into memory, starting at the location specified by the DCDW pointer. The real core word address in the pointer is doubled to determine the word location of the DCDW; therefore, it is necessary that DCDW's always be located on even word boundaries.

*Note: The channels use only real core addresses; no virtual addresses are allowed.*

The Data Control Double Word contains the byte count and byte address of the data area(s) to/from which data is to be transferred. The format of the DCDW is:



Normal DCDW Format

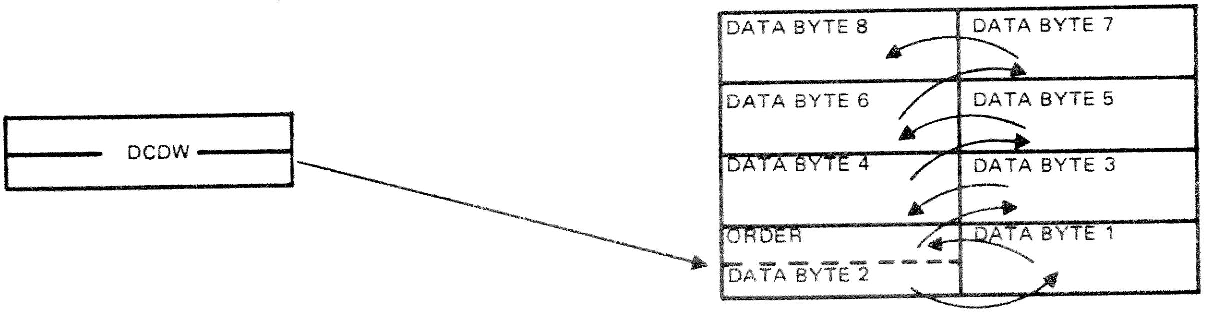


New Pointer DCDW Format

If bit 0 (the low order bit of the byte address) = 1, transfer starts with the right half-word (least significant half).

If bit 0 = 0, transfer starts with the left half word.

Note: If the device order is placed with the data to be transferred and the order causes the device to switch to a read backward mode, the following operation occurs: The DCDW specifies the location of the device order; after transferring the order, the byte address is incremented. On the next transfer, the machine is directed to this new location. After transferring the first data byte, the DCDW byte address is decremented and the machine is directed back to the byte containing the device order, overwriting the order byte with the next data byte. This condition is illustrated below.



If neither the data chaining bit (bit 1) or the new pointer bit (bit 2) are set, transfer ends at the completion of the DCDW. The channel end bit of the pointer is then set and the device terminates.

If the data chaining bit is set (=1), the channel, upon completion of the current DCDW, proceeds to the next DCDW in the list (which must be contiguous).

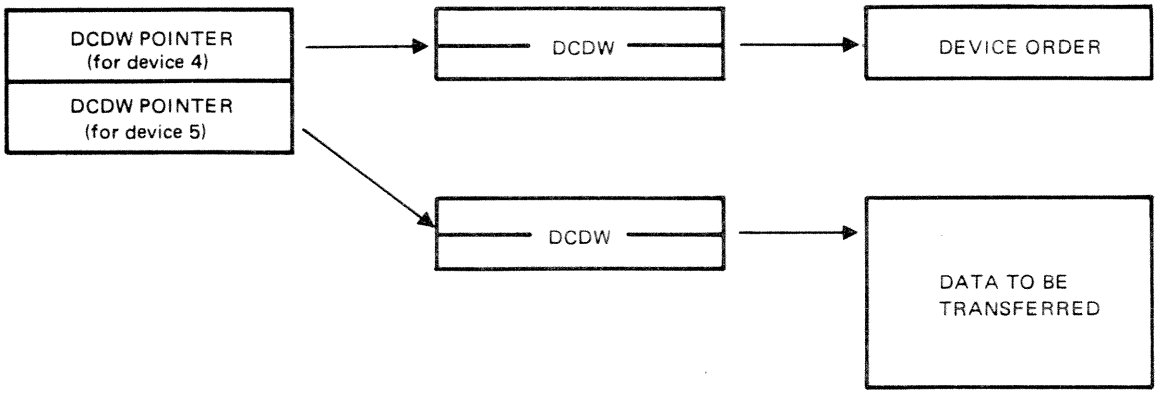
If, when proceeding to the next DCDW, the new pointer bit is set (=1), the second word of the DCDW becomes a new DCDW pointer. In the multiplexor channel this new DCDW pointer replaces the old DCDW pointer in memory. The selector channel does not modify memory, but maintains the new pointer in the channel hardware. (The count field of the new pointer DCDW is ignored.) Thus, the new pointer DCDW allows noncontiguous lists of DCDW's to be chained together.

The byte count field of the DCDW specifies the total number of bytes to be transferred under control of the current DCDW. It may never be specified as zero (except as a new pointer) and must not be large enough to allow the byte address to cross a block boundary in a mapped system (counts greater than 1024 are not recommended for general applications).

The second word of the DCDW contains a word address which, when combined with the low order bit in the first word, provides the byte address of the first byte to be transferred under control of the DCDW.

Some devices may reverse the direction of data transfer (input versus output) during certain operations (e.g., multibyte orders). On the selector channel this may occur only when chaining between DCDW's; however, operation will proceed normally using the new DCDW from the same list. On the multiplexor channel a change in direction may occur at any time (indicated by a change in device numbers) and will cause a different DCDW list to be executed. Thus, two DCDW lists must be prepared based on both the input and output device numbers of the device.

*Example:* Before performing the transfer operation, the multiplexor channel requests device identification. The device replys with a device number of 4, which indicates an input operation. The DCDW operation to transfer the device order on device number 4 (illustrated on next page) is performed. Upon completion of this DCDW operation, the channel again requests identification from the same device. This time the device will reply with a device number of 5, which indicates an output operation. As a result, the DCDW pointer will specify a new DCDW list to be used in transferring the proper data.



NOTE: The first byte of an order is always output regardless of the device number; however, additional bytes of a multibyte order must use an odd device number.

5. A device order is then written to activate the device controller. The order may be located separately from the data with its own DCDW or placed with the data to be transferred, but it must be the first byte to be output.

   For a multibyte order on the selector channel, a separate DCDW must be specified for the data.

6 After the above information has been prepared and assembled in the proper sequence and format and after verifying the device available, the desired data transfer is performed by executing an IOC (Input/Output Control) instruction followed by an SIO (Start I/O) channel command in the block mode.

   *Example:* IOC
   SIO 0E05

   On execution of the SIO command, the selector channel locates the data to be transferred by means of the DCDW pointer and DCDW's and performs the transfer asynchronously and independently of the CPU.

   For block transfers on the multiplexor channel, the CPU performs the necessary update to the DCDW's and DCDW pointer in memory and transfers each byte directly from memory, as each device becomes ready, asynchronously of the program. The programmable registers are not affected. The programmer may determine how far the multiplexor channel has progressed with the operation by testing the DCDW pointer and the DCDW's.

7. The transfer may be repeated on the selector channel by resetting the channel end bit (if desired) and executing another SIO.

   On the multiplexor channel the DCDW pointer and the DCDW's must all be re-initialized (new pointer DCDW's are not changed) before executing another SIO.

## SIO (Start Input/Output) Channel Command Operation in Block Mode

The SIO command performs the following operations when transferring data in the block mode on the multiplexor channel.

1. An IOC instruction is executed by the CPU and the associated command is transferred by the channel to the device controller. If the command was an SIO in the block mode, the DCDW pointer location is determined by adding $28_{10}$ to the channel number and device number at it appears in the SIO command.

2. The DCDW location is calculated by doubling the contents of the DCDW pointer location.

3. If the byte count of the DCDW is zero, the operation branches to step 12. If it is not zero, the count is decremented, and the byte specified by the byte address of the DCDW is transferred to the device with a hardware generated EOA command.

4. The DCDW byte address is incremented.

5. If the resultant byte count is zero and the chaining bit is also zero, the channel end bit in the DCDW pointer is set and a TWC command is sent to the device. The TWC stops the device from requesting more data transmission and causes the device to terminate when complete. Upon termination the device generates an interrupt if armed.

6. If a channel or memory error occurred during the preceding operations, the error bit (IUS bit 7) in the device is set by transmitting an OUS command to the device.

7. The CPU then proceeds to execute the next instruction, while the channel waits for a transfer request from the device. (A transfer request is a signal from a device indicating it is ready for a data transmission in the block mode.)

8. When the channel receives a transfer request, it performs an IDN command. (This is a special command used by the channel to determine the highest priority device signaling a transfer request.) $28_{10}$ is then added to the channel and device number response received to the IDN command to determine the DCDW pointer location.

9. The DCDW location is calculated by doubling the contents of the DCDW pointer location.

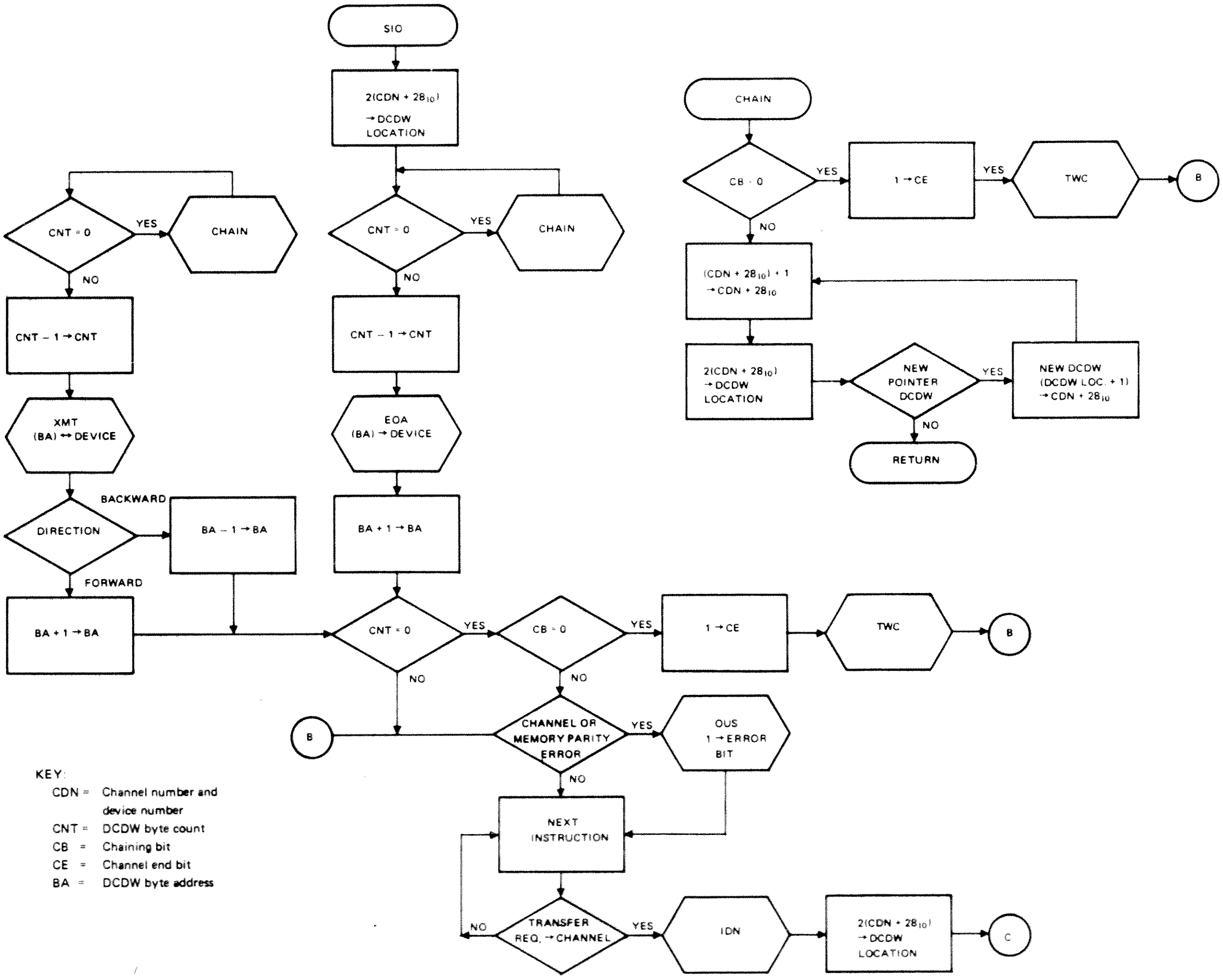
10. If the byte count of the DCDW is zero, the operation branches to step 12. If it is not zero, the count is decremented, and the byte specified by the byte address of the DCDW is transferred to/from the device with a hardware generated XMT command. The direction of data transfer is determined by the least significant bit of the device number.

11. The byte address is then incremented (decremented if a read backward indication was received from the device). The operation returns to step 5.

12. If the chaining bit in the DCDW is not set, an error condition has been detected. The channel end bit is set, a TWC command is sent to the device and the operation proceeds to step 6. If the chaining bit is a one, the DCDW pointer is incremented to the next consecutive DCDW and the operation proceeds using this new DCDW.

13. If the new pointer bit is not set, the operation returns to and repeats the step that branched to step 12. If the new pointer bit is set, the second word of the DCDW replaces the DCDW pointer and this step is repeated using the new DCDW specified by the updated DCDW pointer.

## SIO – Start Input/Output Operation in Block Mode



# SECTION III
# MULTIPLEX CHANNEL

## BASIC DESCRIPTION

The Model 4706-01 Multiplexor unit (M channel) is the basic input/output unit supplied with every SCC 4700 computer. It provides timing and data routing logic to interface the SCC 4700 series device controllers with the SCC 4700 CPU. The M channel interface configuration is shown in Figure 3-1. The M channel interfaces with the I/O module, which is a part of each device controller. The M channel can interface with from 1 to 64 peripheral devices and provides the capability to perform data transfer in background (microinterrupt logic) or foreground (program interrupt) logic.

## THEORY OF OPERATION

Interface operations through the M channel are under direct control of the CPU with data and control signals using CPU data and address paths to memory. Data transfer between the channel and device controller is 8-bit byte oriented; however, the channel can operate in character (single byte) or block transfer modes. Character mode is program controlled for every 8 bits. Transfer of data in block mode, once initiated, continues automatically until completed. Block mode operation requires referencing areas of core memory with pointer addresses as described in Section II, Input/Output Instructions.

Since the M channel is capable of simultaneous interleaved operation of many slow I/O devices, it handles each data request as a separate transaction, complete in itself, without any reference to those coming before or after. Thus, many devices may be active simultaneously with only brief conflicts when one device may have a short wait for service until the channel has completed a transmission for another device. The character buffer in the device controller must cover these occasional waits.

Since the M channel does not retain any information from one data request to the next (as opposed to the S channel), it must be furnished with the necessary addresses required to service each request. In the byte mode this is provided explicitly by the CPU program via the programmable registers. In the block mode, the necessary pointers and control information are kept in core, but are directly accessed and maintained by the microprogram without involving the programmable registers. Physically, the M channel is largely characterized by microcode and the only hardwired logic involves a relatively small interface and interrupt-control module.

The following is an example of M channel microcode:

1. Obtain device number.

2. Reference core location unique to device containing DCDW pointer and Channel End flag. DCDW pointer locates the two-word Data Control Double Word (DCDW) in core. The first of these two words contains a byte counter, a byte position bit, and a control field. The second word contains a data pointer.

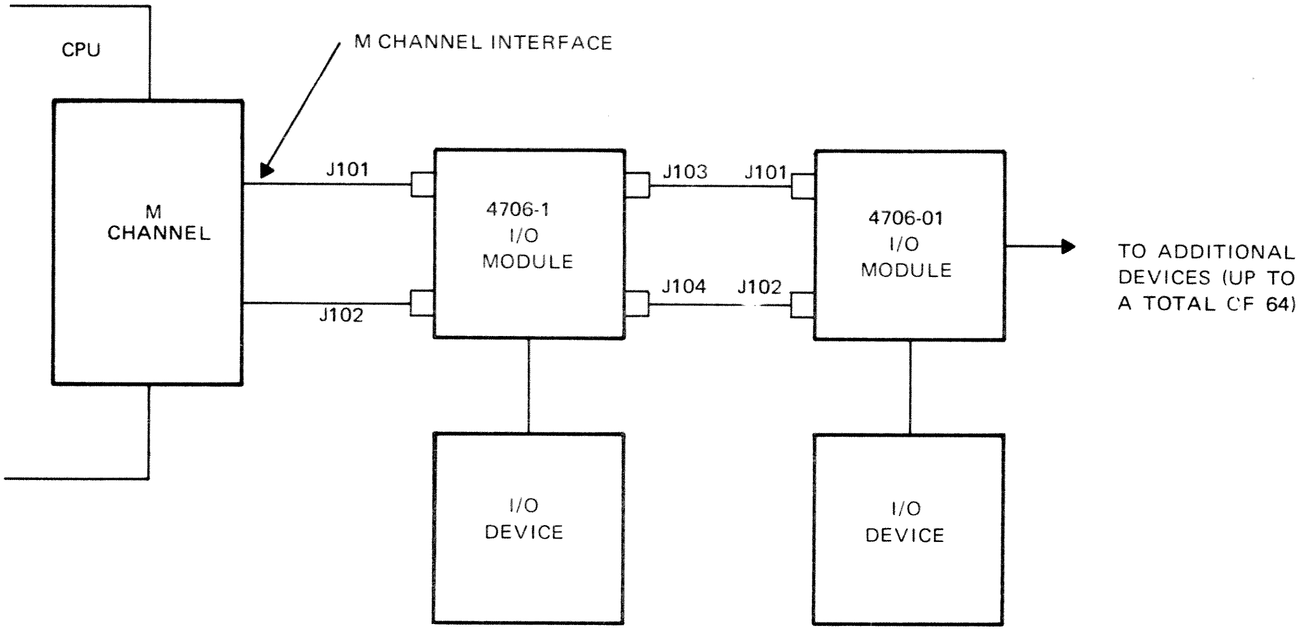


*Figure 3-1. Channel Interface*

3. Check byte counter for zero. If zero, jump to step 10; otherwise, continue.

4. Decrement byte counter, reverse byte-position bit, and restore first word of DCDW to core. Retain modified word.

5. Increment, decrement, or leave unchanged the data word pointer, depending upon the byte-position bit and the physical direction of movement in the device. Restore the second word of the DCDW to retain the data word pointer.

6. (Input only), Transmit the data byte from the specified device into the left or right half of the core location addressed by the data word position. Left or right is indicated by the byte-position bit.

6. (Output only), Transmit the left or right half of memory word addressed by the data word pointer to the specified device. Left or right is specified by the byte-position bit.

7. Examine previously decremented byte-counter and control field. If byte counter has reached zero and if control field does not specify chaining, proceed to step 8, otherwise, skip to step 9.

8. Set channel end flag in DCDW pointer and order device to terminate when complete.

10. (Jump from step 3), Examine chain bit in control field. If chaining is not specified, jump to step 8; otherwise, continue.

11. Increment DCDW by 2.

12. Obtain the first word of the DCDW.

13. Examine New Pointer bit of the control field. If new pointer is not specified, jump to step 3; otherwise, continue.

14. Obtain second word of new DCDW, replace old DCDW pointer with it. Jump to step 12.

The M channel may be functionally divided into the I/O data transfer logic, the microinterrupt logic, and the program interrupt logic. The channel provides interface for the CPU I/O strobe (IOS), channel commands, device addresses and orders, mode bits, interrupt arm/disarm master clear, and data byte signals to the device controller. Device controller data, address, and acknowledge signals are also routed to the CPU by the channel. The channel responds to channel commands from the CPU in one of two ways. Channel response occurs on a priority basis when the command is an Input Interrupting Unit (IIU) or an Input Device Number (IDN) channel command. At other times the channel responds when addressed.

Block transfer requests from the device controller cause the channel to generate a Transfer Interrupt Request (TIR) on a channel priority basis. As TIR does not interrupt the main program, it is referred to as a microinterrupt. Service requests from the device controllers cause the channel to generate a Program Interrupt Request (PIR). PIR interrupts the main program and is referred to as a program interrupt. Thus, the CPU must honor a PIR by executing an interrupt service routine entered from a core location unique to the channel number to which the interrupting device is connected.

A TIR to a M channel activates the microcode interrupt logic on a priority basis. This steals processing time from the CPU, but does not disturb the instruction sequence or the programmable registers.

M channel input and output signals are shown in Figures 3-2 and 3-3 and are described in the following paragraphs.

## SIGNALS

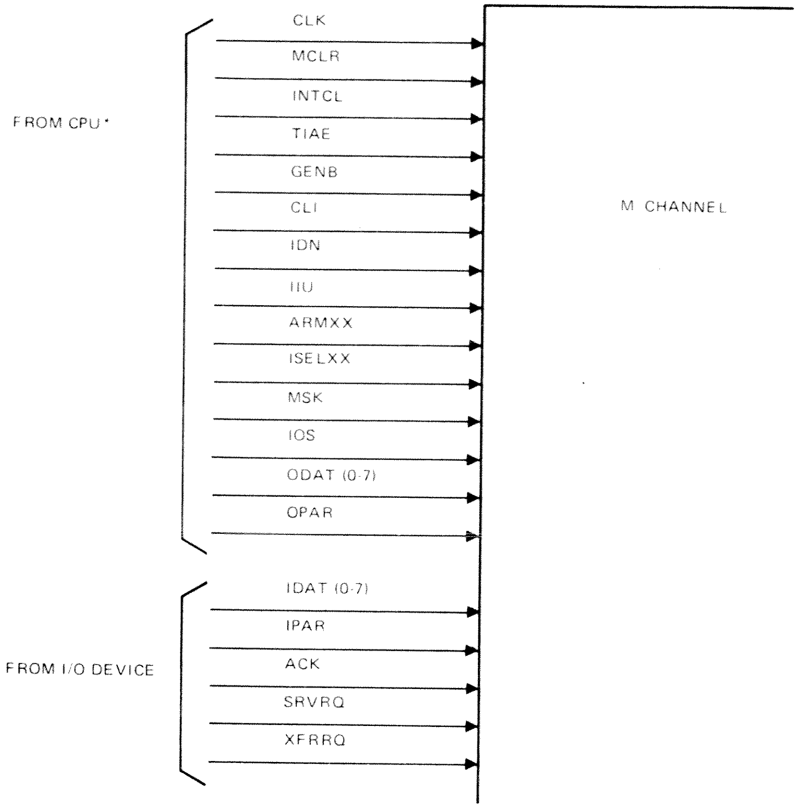
### Input

1. MCLR – Master Clear: This signal clears the channel logic on the logic 1 to logic 0 transition.*

2. CLK – System Clock.*

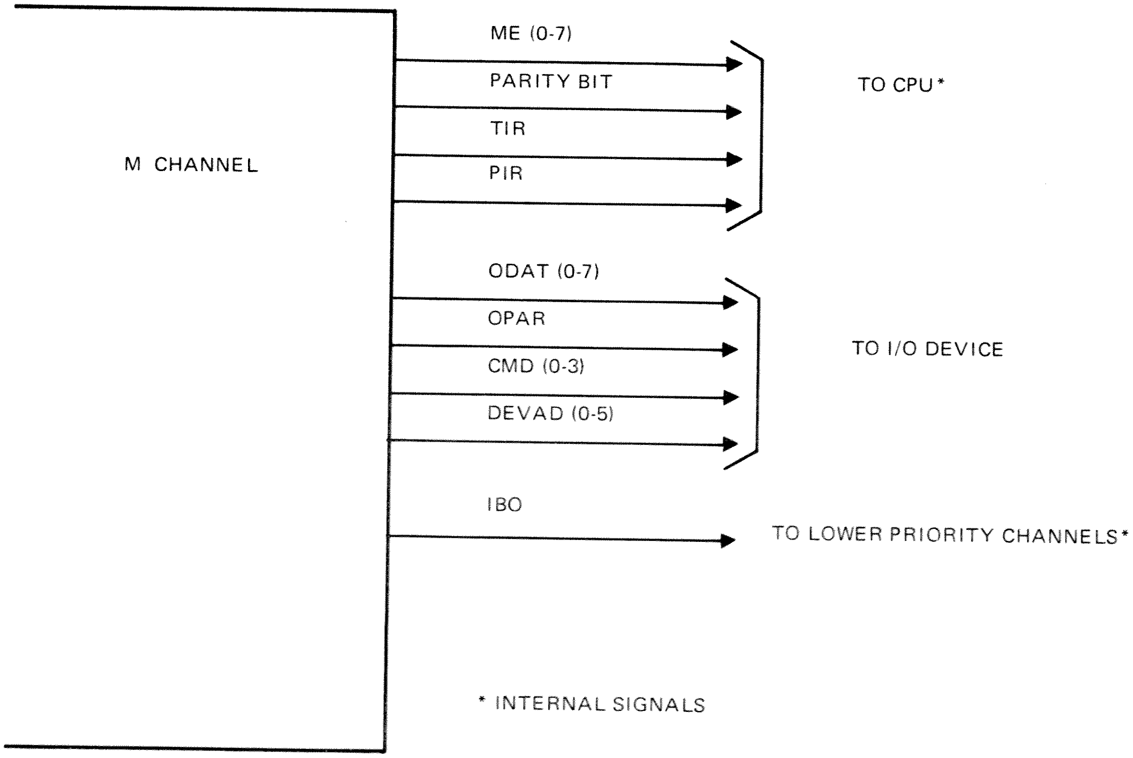
3. INTCL – Interrupt Control: This signal, when at logic 1, inhibits the channel interrupt logic from entering the waiting state and allows the priority logic to stabilize. The highest priority channel with an SRVRQ interrupt pending will enter the active state on the logic 1 to logic 0 transition of INTCL.*

* Internal signals, not appearing at the interface but listed for reference purposes.



*INTERNAL SIGNALS NOT AVAILABLE AT INTERFACE

*Figure 3-2. Input Signals*



* INTERNAL SIGNALS

*Figure 3-3. Output Signals*

4. TIAE – Transfer Interrupt Address to the E Bus: A positive pulse used to gate the 5-bit address of the highest priority channel in the waiting state onto the CPU E bus. The trailing edge of TIAE drops the INTCL line, causing the channel waiting logic to reset and the active logic to be set.*

5. GENB – Group Enable: A signal true at logic 1 used to enable the channel active interrupt logic.*

6. CLI – Clear Interrupt: This signal clears the active interrupt logic of the highest priority channel on the logic 0 to logic 1 transition.*

7. IDN – Input Device Number: This signal transfers the device number on the input data bus in response to a transfer request from the device. The signal is true at logic 1.*

8. IIU – Input Interrupting Unit: This signal transfers the device address number on the input data bus in response to a service request from the device. The signal is true at logic 1.*

9. ARMXX – Interrupt Arm Bit: A signal used to programmatically arm or disarm the waiting logic. This signal is a logic 1 bit and must be accompanied by the ISELXX and a mask (MSK) signal.*

10. MSK – Mask Bit: A logic 1 bit used with the ARMXX bit.*

11. XFRRQ – Transfer Request: A logic 0 signal from the I/O device to the channel indicating a block transfer mode request. This signal initiates a microinterrupt and request sequence of events.

12. SRVRQ – Service Request: A logic 0 service request signal from the I/O device. This signal initiates a program interrupt request sequence of events.

13. IDAT (0-7) – Eight Interrupt Lines: The input data should remain stable for the duration of the ACK signal. The data lines are true at logic 0.

14. ACK – Acknowledge: A strobe signal from the I/O device which validates the data or device address lines returning to the channel from the I/O device.

15. IOS – Input/Output Strobe: A strobe signal from the CPU to the channel which validates the address, command and data output lines.*

16. IPAR – Input Parity Line: An optional parity line which carries parity on the 8 bits of input data. This line is true at logic 0 and is valid for the duration of the ACK signal.

* Internal signals, not appearing at the interface but listed for reference purposes.

### Output

1. ODAT (0-7) – Eight Output Data Lines: The output data lines to the I/O devices are true at logic 0 and are valid for the duration of the IOS signal.

2. OPAR – Output Parity Line: An optional output parity line which carries parity for the 8 bits of output data. This line is true at logic 0 and is valid for the duration of the IOS signal. Channel parity is odd.

3. CMD (0-3) – Four Device Command Lines: These lines are used to transfer the ten device commands. The lines are true at logic 0 and are valid for the duration of the IOS signal.

4. DEVAD (0-5) – Six Device Address Lines: These lines are used to address up to a total of 64 I/O devices. The lines are true at logic 0 and are valid for the duration of the IOS signal.

5. TIR – Transfer Interrupt Request: A microinterrupt request signal sent to the CPU by the channel after receiving the XFRRQ signal from the I/O device. This signal is generated only if no higher priority channel is busy. The signal is true at logic 0.*

6. PIR – Program Interrupt Request: This signal is generated by the channel and sent to the CPU when the channel receives the SRVRQ signal from the I/O device. The signal is generated only if the channel is armed and no higher priority channel is busy. The PIR signal is true at logic 0.*

7. IBO – Interrupt Busy: A signal sent from a higher priority channel to a lower priority channel indicating the higher priority channel is busy. The signal is true at logic 0.*

8. ME (0-7) – M Channel Data Line to E Bus: This data line is the IDAT(0-7) data received from the I/O device. The data lines to the CPU E bus are true at logic 0. A parity bit is also sent with the data.*

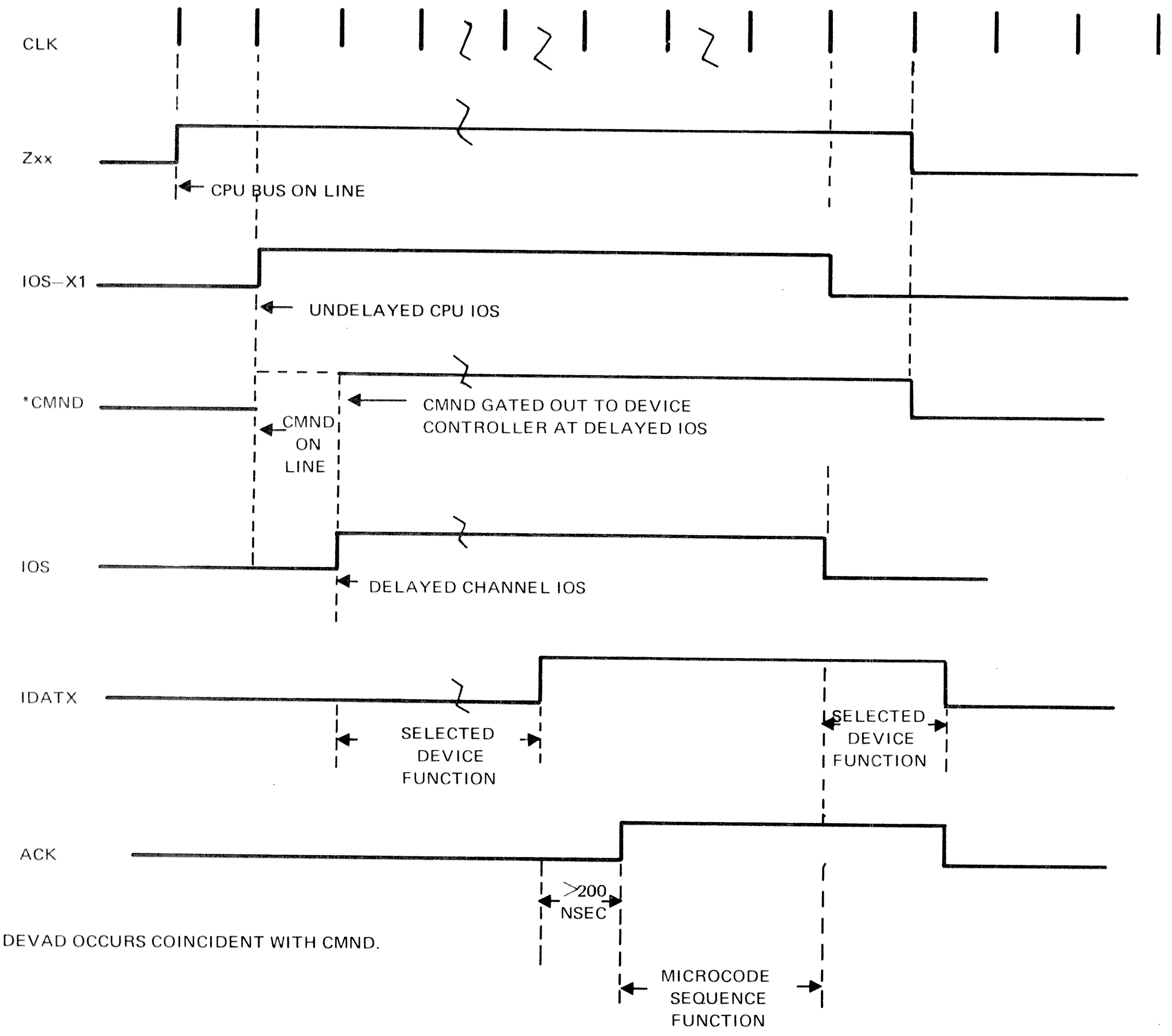
## TIMING DIAGRAMS

### Input and Output Data Transfer

All timing diagrams for the input data transfer (I/O device to CPU) and output data transfer (CPU to I/O device) operations are shown in Figures 3-4 and 3-5, respectively.

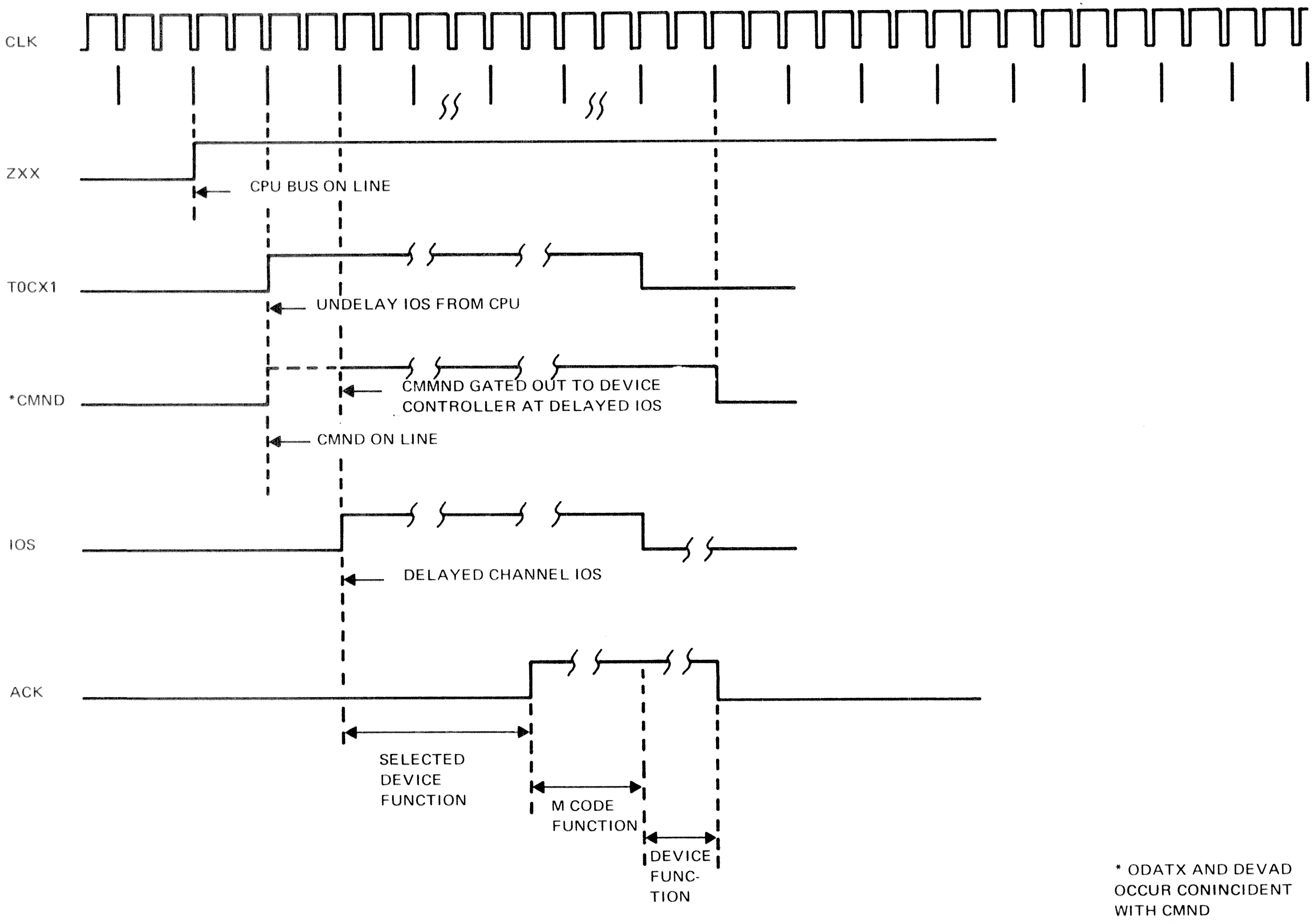
When the CPU bus is on the line, the IOS signal strobes out the device command and device address signals to the I/O device. The addressed I/O device then responds with the acknowledge (ACK) signal and data is strobed in or out of the device at that time.

* Internal signals, not appearing at the interface but listed for reference purposes.



*DEVAD OCCURS COINCIDENT WITH CMND.

*Figure 3-4. Input Data Transfer Timing Diagram*



* ODATX AND DEVAD OCCUR CONINCIDENT WITH CMND

*Figure 3-5. Output Data Transfer Timing Diagram*

### Microinterrupt Mode

The timing diagram for the microinterrupt mode is shown in Figure 3-6. The M channel microinterrupt logic initiates a block data transfer request to the CPU on receiving an XFRRQ signal from the I/O device. Signal XFRRQ sets the TIR logic in the channel on the following CLK pulse. With the TIR logic set, the channel generates a TIR signal to the CPU when no higher priority channel has an interrupt pending. The CPU acknowledges the request with an IDN command. TIR is reset when the device has been serviced with a transmit (XMT), or terminates when serviced with complete (TWC) command. The IOS signal strobes out the channel address to the CPU. A busy signal is generated by the channel during TIR to inhibit lower priority channels.

### Program Interrupt Mode

The timing diagram for the program interrupt mode is shown in Figure 3-7. The SRVRQ signal sets the channel WAIT logic if the channel is armed, INTCL is false, and the channel ACTIVE logic is not set. When the WAIT logic is set and no higher priority channel is waiting, the channel generates PIR and gates a 5-bit address onto the CPU E bus at TIAE strobe time. The trailing edge of TIAE drops INTCL, and with INTCL down, the ACTIVE logic is set and the WAIT logic is reset. The CPU clears the highest priority ACTIVE logic with the CLI signal. The program interrupt busy logic generates a signal to inhibit lower priority channels when the channel is waiting. The channel program interrupt input to the CPU is shared with the external interrupt subsystem.

## CONNECTORS

Connectors required for M channel and I/O module interface are numbered as shown in Figure 3-8. The J101 and J102 connectors connect the IOM to the M channel or to a preceding IOM. Connectors J103 and J104 connect the IOM to a succeeding IOM. A listing of the connectors and connector pin numbers with the associated signals follows. All signal signatures ending in the letters LO indicate the common side of the signal.

* Internal signals, not appearing at the interface but listed for reference purposes.

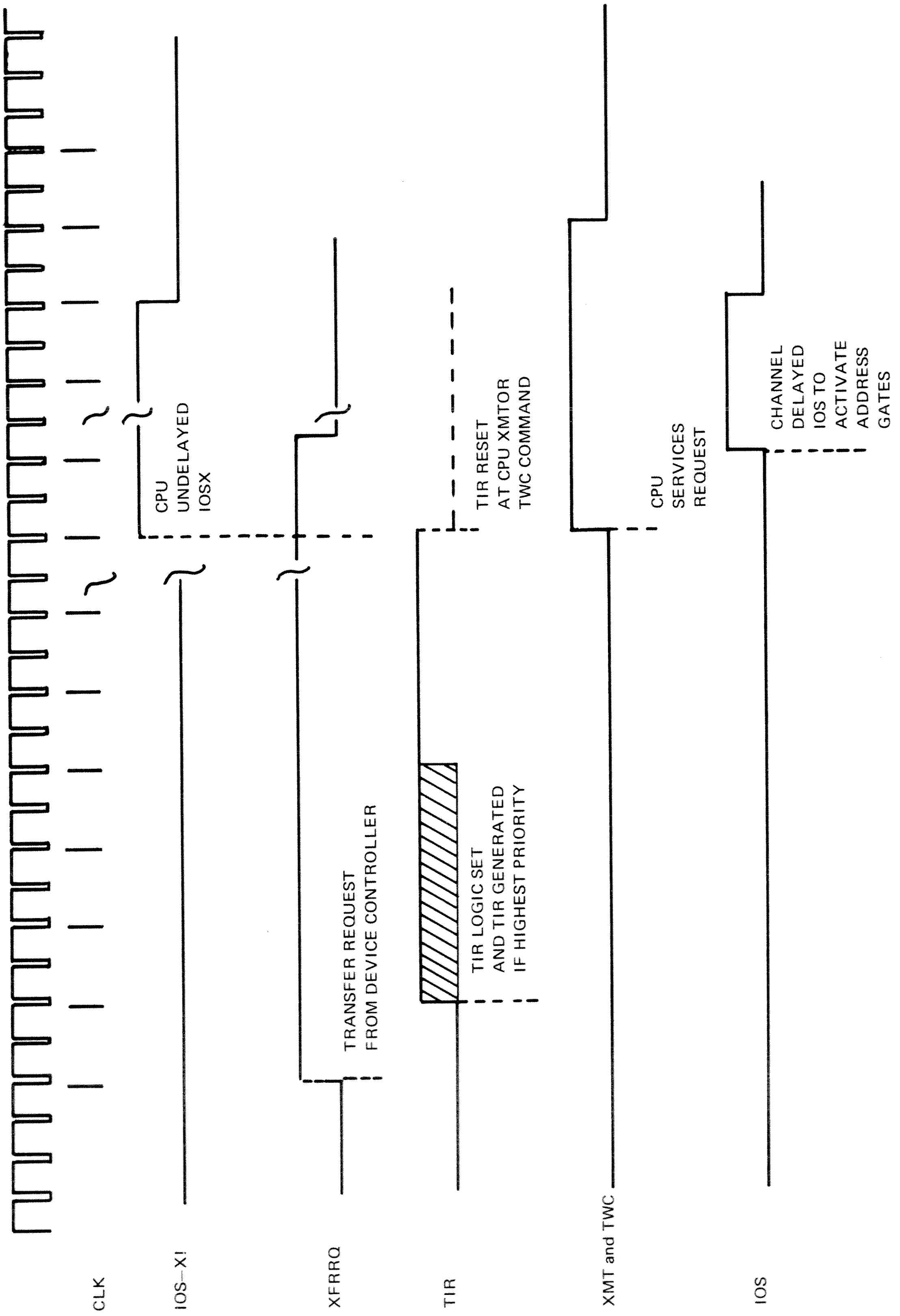

CLK
IOS—X!
XFRRQ
TIR
XMT and TWC
IOS
CPU UNDELAYED IOSX
TRANSFER REQUEST FROM DEVICE CONTROLLER
TIR RESET AT CPU XMTOR TWC COMMAND
TIR LOGIC SET AND TIR GENERATED IF HIGHEST PRIORITY
CPU SERVICES REQUEST
CHANNEL DELAYED IOS TO ACTIVATE ADDRESS GATES

*Figure 3-6. MicroInterrupt Mode-Timing Diagram*

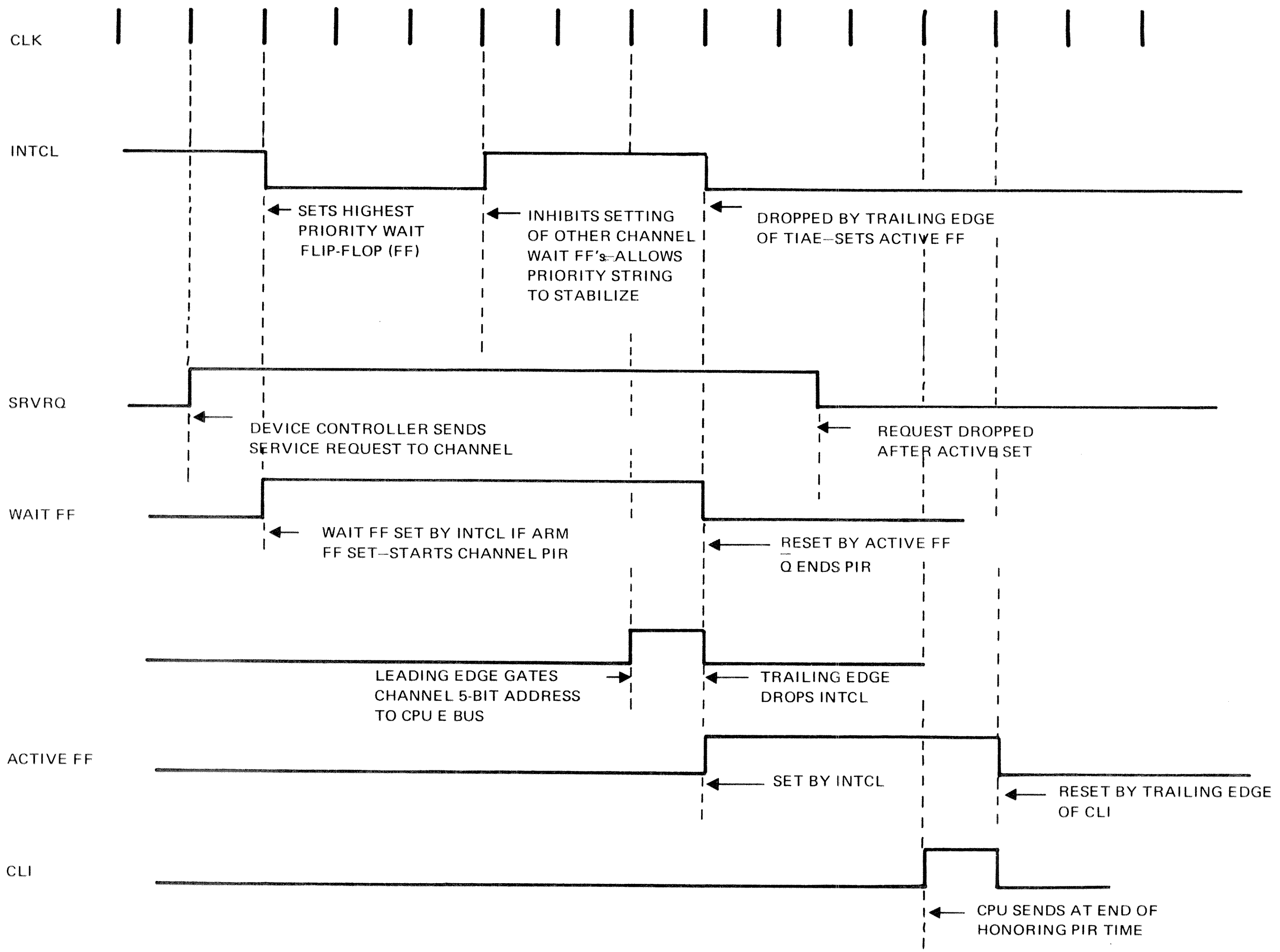

CLK
INTCL
SRVRQ
WAIT FF
ACTIVE FF
CLI
SETS HIGHEST PRIORITY WAIT FLIP-FLOP (FF)
INHIBITS SETTING OF OTHER CHANNEL WAIT FF's–ALLOWS PRIORITY STRING TO STABILIZE
DROPPED BY TRAILING EDGE OF TIAE–SETS ACTIVE FF
DEVICE CONTROLLER SENDS SERVICE REQUEST TO CHANNEL
REQUEST DROPPED AFTER ACTIVE SET
WAIT FF SET BY INTCL IF ARM FF SET–STARTS CHANNEL PIR
RESET BY ACTIVE FF
$\overline{Q}$ ENDS PIR
LEADING EDGE GATES CHANNEL 5-BIT ADDRESS TO CPU E BUS
TRAILING EDGE DROPS INTCL
SET BY INTCL
RESET BY TRAILING EDGE OF CLI
CPU SENDS AT END OF HONORING PIR TIME

*Figure 3-7. Program Interrupt Mode Timing Diagram*

## CHANNEL BOARD A/B

J101

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | CMDO | f | DVAD1 |
| B | CMDOLO | j | DVAD1LO |
| C | CMD1 | k | DVAD2 |
| D | CMD1LO | l | DVAD2LO |
| E | CMD2 | m | DVAD3 |
| F | CMD2LO | n | DVAD3LO |
| H | CMD3 | p | DVAD4 |
| J | CMD3LO | r | DVAD4LO |
| K | ODAT0 | s | DVAD5 |
| V | ODAT0LO | t | DVAD5LO |
| L | ODAT1 | u | IDAT0 |
| M | ODAT1LO | v | IDATLO |
| N | ODAT2 | w | IDAT1 |
| P | ODAT2LO | x | IDAT1LO |
| R | ODAT3 | y | IDAT2 |
| S | ODAT3LO | JJ | IDAT2LO |
| T | ODAT4 | z | IDAT3 |
| U | ODAT4LO | AA | IDAT3LO |
| W | ODAT5 | BB | IDAT4 |
| X | ODAT5LO | CC | IDAT4LO |
| Y | ODAT6 | DD | IDAT5 |
| Z | ODAT6LO | EE | IDAT5LO |
| a | ODAT7 | FF | IDAT6 |
| b | ODAT7LO | HH | IDAT6LO |
| c | OPAR | KK | IDAT7 |
| d | OPARLO | LL | IDAT7LO |
| e | DVAD0 | MM | |
| h | DVAD0LO | NN | |

*Figure 3-8 Connectors (Sheet 1 of 4)*

J102

| PIN | SIGNAL |
|---|---|
| A | IOS |
| B | IOSLO |
| C | MCLR |
| D | MCLRLO |
| E | ACK |
| F | ACKLO |
| H | SRVRQ |
| J | SRVRQLO |
| K | XFRRQ |
| V | XFRRQLO |
| L | IPAR |
| M | IPARLO |

*Figure 3-8 Cont'd (Sheet 2 of 4)*

CONNECTOR J103

| PIN | SIGNAL |
|---|---|
| A | CMD0 |
| B | CMD0LO |
| C | CMD1 |
| D | CMD1LO |
| E | CMD2 |
| F | CMD2LO |
| H | CMD3 |
| J | CMD3LO |
| K | ODAT0 |
| V | ODAT0LO |
| L | ODAT1 |
| M | ODAT1LO |
| N | ODAT2 |
| P | ODAT2LO |
| R | ODAT3 |
| S | ODAT3LO |
| T | ODAT4 |
| U | ODAT4LO |
| W | ODAT5 |
| X | ODATSLO |
| Y | ODAT6 |
| Z | ODAT6LO |
| a | ODAT7 |
| b | ODAT7LO |
| c | OPAR |
| d | OPARLO |
| e | DVAD0 |
| h | DVAD0LO |
| f | DVAD1 |
| j | DVAD1LO |
| k | DVAD2 |
| l | DVAD2LO |
| m | DVAD3 |
| n | DVAD3LO |
| p | DVAD4 |
| r | DVAD4LO |
| s | DVAD5 |
| t | DVAD5LO |
| u | IDAT0 |
| v | IDAT0LO |
| w | IDAT1 |
| x | IDAT1LO |
| y | IDAT2 |
| JJ | IDAT2LO |
| z | IDAT3 |
| AA | IDAT3LO |
| BB | IDAT4 |
| CC | IDAT4LO |
| DD | IDAT5LO |
| EE | IDAT5LO |
| FF | IDAT6 |
| HH | IDAT6LO |
| KK | IDAT7 |
| LL | IDAT7LO |
| MM | |
| NN | |

*Figure 3-8 Cont'd (Sheet 3 of 4)*

J104

| PIN | SIGNAL |
|---|---|
| A | IOS |
| B | IOSLO |
| C | MCLR |
| D | MCLRLO |
| E | ACK |
| F | ACKLO |
| H | SRVRQ |
| J | SRVRQLO |
| K | XFRRQ |
| V | XFRRQLO |
| L | IPAR |
| M | IPARLO |
| N | |

*Figure 3-8 Cont'd (Sheet 4 of 4)*

# SECTION IV

# SELECTOR CHANNEL

## BASIC DESCRIPTION

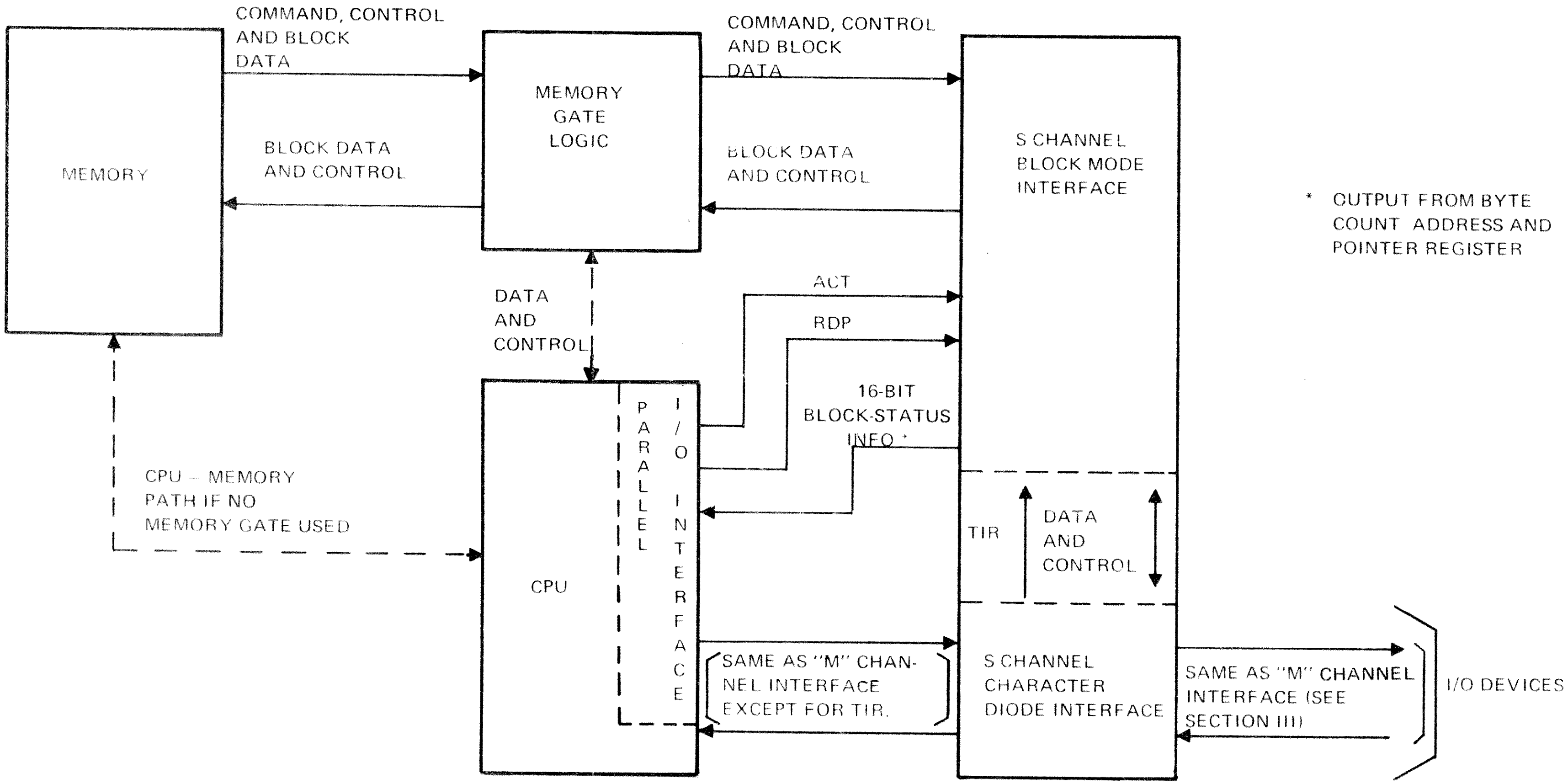
The Model 4706-01 Selector (S) channel is the high-speed byte-oriented input/output channel supplied for the SCC 4700 computer. Physically, the S channel resembles the M channel plus a set of logic hardware including registers to replace the microcode sequences used in M channel operation. The S channel provides the control and data routing logic to interface high-speed data transfer operations between the SCC 4700 computer and up to 64 byte-oriented I/O devices. Data may be transferred on the channel in character (single byte) or block modes. In the single byte mode the S channel functions in the same manner as described in Section III for the M channel byte mode. For block mode, the S channel functions in a different manner from the M channel with respect to the CPU and memory interfaces. A block diagram of the S channel interface configuration is shown in Figure 4-1. The data and control signal interface is shown in Figure 4-2.

The S channel uses wired logic to follow the same procedure as the M channel (see Section III) except that the initial value of the DCDW pointer and each new DCDW is obtained from core. The modified values are never restored to core with the exception of the Channel End flag. Thus, a block operation may be repeated without resetting all the pointers and counters in core.

## THEORY OF OPERATION

The S channel contains its own internal registers for storing device identification, type of transmission, data pointers, byte counter and data packing/unpacking information, thus enabling large blocks of data to be transferred to or from core with only one core cycle per word transferred. Since the S channel has direct access to memory, it need not steal any time from the CPU except when both reference the same memory module or when a new block of data is to be started. The S channel retains all the information internally required to process a data request from a single device. As a result, it cannot handle more than one device at a time.

When the S channel is activated by the CPU for block transfer mode, the S channel proceeds to access memory for command and control information. Access of memory in the block mode is accomplished using the DCDW pointers described in Section II. Using the start address and byte count specified by the memory information, the S channel performs the data transfer asynchronously and independently of the CPU. The S channel deactivates the operation when the specified data count has been achieved. Block transfers may be chained so that the channel can execute a series of block transfers before deactivating the mode. Three S channel registers store block transfer status information: one register stores the byte count, a second register stores the address, and a third register stores the pointer. An optional feature allows the programmer to read the registers. A flow diagram of the general sequence of events is shown in Figure 4-3. The initial location address for the DCDW that controls the transfer, is obtained by adding the channel number, the I/O device number,


MEMORY
COMMAND, CONTROL AND BLOCK DATA
BLOCK DATA AND CONTROL
MEMORY GATE LOGIC
COMMAND, CONTROL AND BLOCK DATA
BLOCK DATA AND CONTROL
S CHANNEL BLOCK MODE INTERFACE
DATA AND CONTROL
ACT
RDP
16-BIT BLOCK-STATUS INFO *
CPU – MEMORY PATH IF NO MEMORY GATE USED
CPU
PARALLEL I/O INTERFACE
TIR
DATA AND CONTROL
SAME AS "M" CHANNEL INTERFACE EXCEPT FOR TIR.
S CHANNEL CHARACTER DIODE INTERFACE
SAME AS "M" CHANNEL INTERFACE (SEE SECTION III)
I/O DEVICES

* OUTPUT FROM BYTE COUNT ADDRESS AND POINTER REGISTER

*Figure 4-1. Channel Interface Configuration Block Diagram*

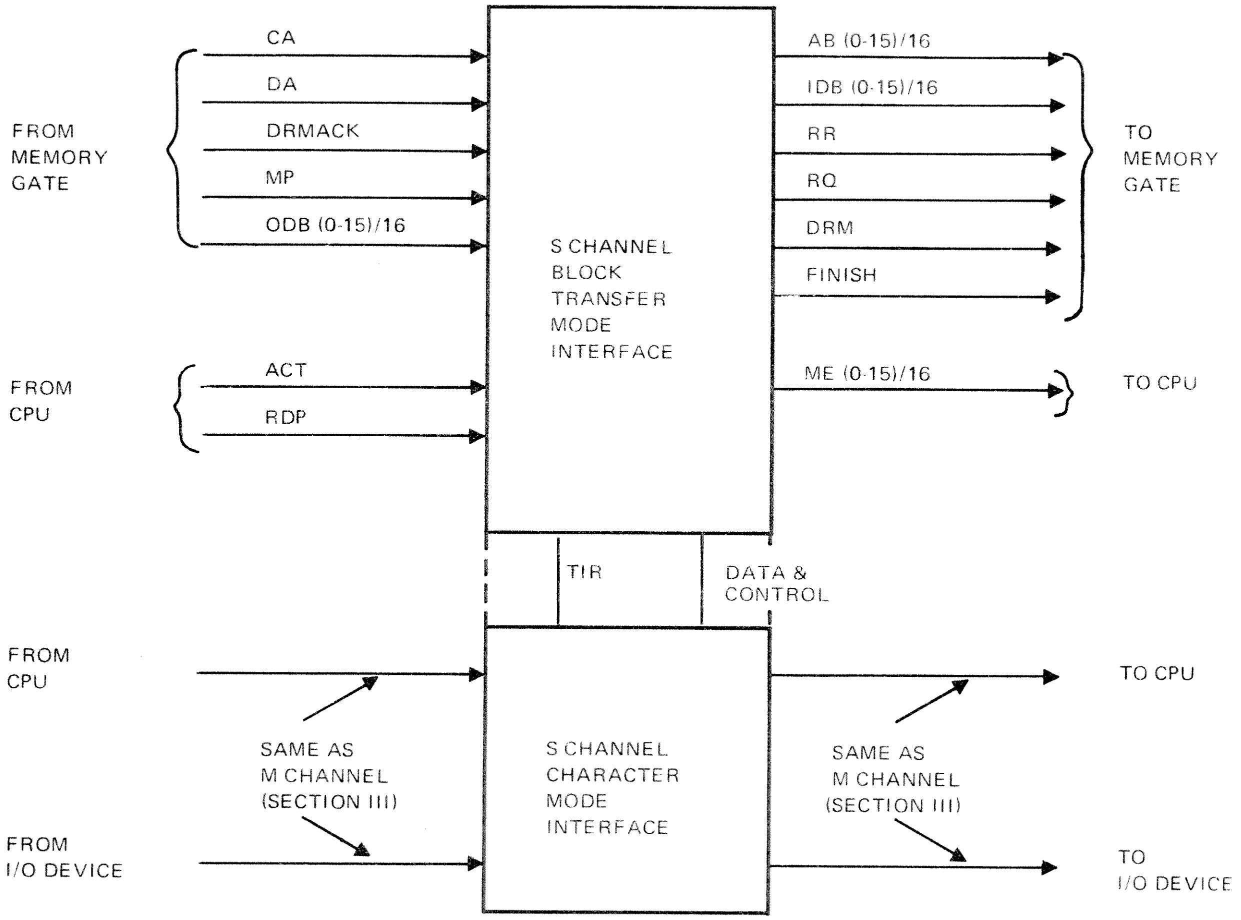


*Figure 4-2. Data and Control Signals*

and the base. The parallel ready (PRDY) is reset at this time to inhibit the read parallel operation while the channel registers are being updated. The location address is loaded into the address register and memory is addressed. Memory then loads the byte address in the pointer real core into the M register. The byte address must be doubled to determine the location of the DCDW. This is accomplished by shifting the address one place to the left as the address is loaded into the channel address and pointer registers. The DCDW byte count is loaded into the channel count register. At this time a channel decoder detects if a new pointer (NP) is required, or if order chaining (OCH) and data chaining (DCH) are to be performed. Data is then transferred from memory to the I/O device or loaded from the device into memory. The S channel issues the commands required to transfer the data characters read or output status plus terminate when complete (TWC). In addition, the programmer may issue commands to the device when the channel is transferring; however, a program command interrupts the channel operation and data may be lost.

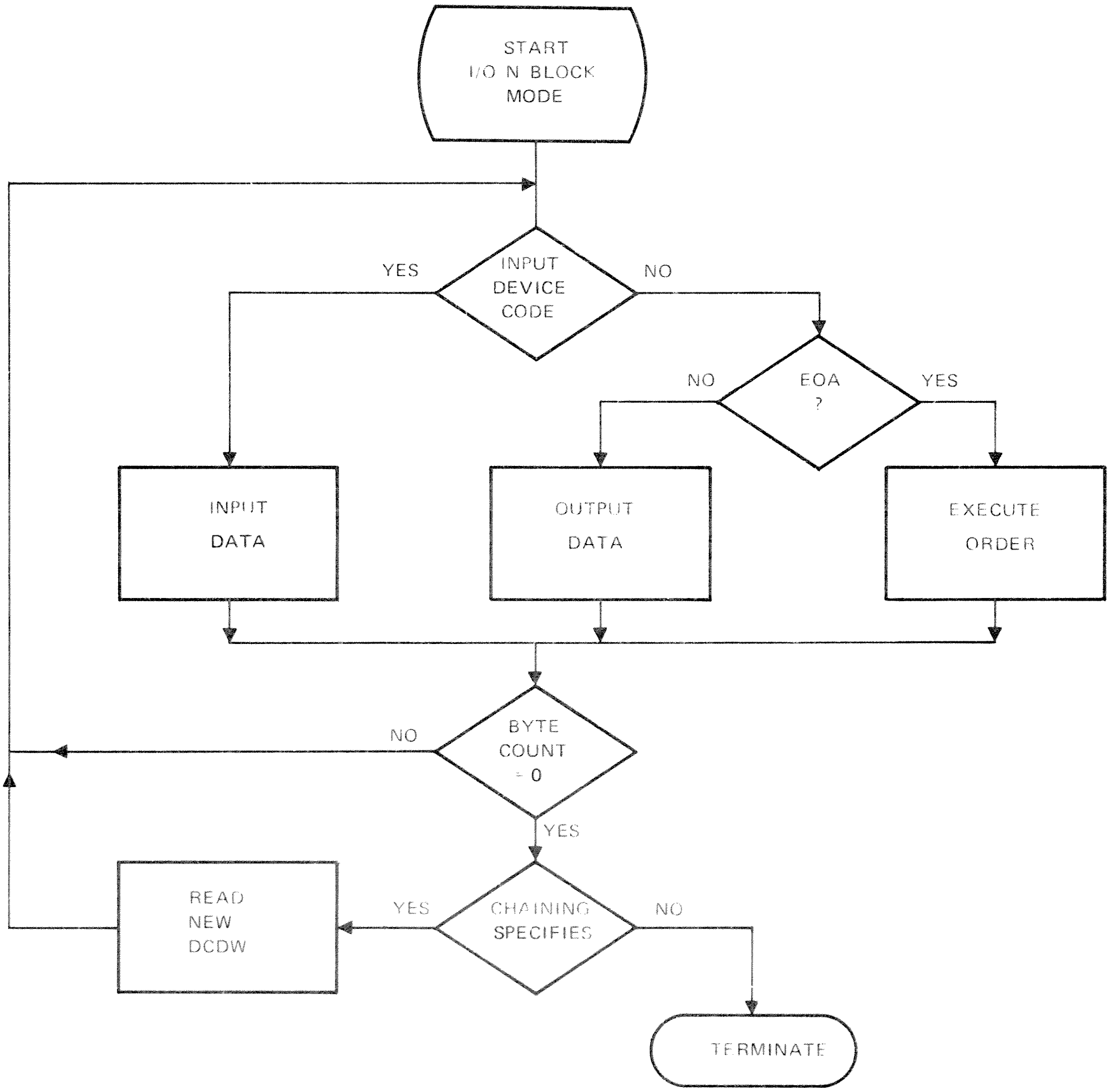


*Figure 4-3. S Channel Sequence of Operation Flow Diagram*

## SIGNALS

1. Channel — Device Signals: The signal interface between the I/O device and the S channel is the same as described for the M channel in Section III.

2. Channel — CPU Signals: The signal interface between the CPU and the S channel is the same as described for the M channel in Section III except for the Transfer Interrupt Request (TIR) signal.

3. Channel — Parallel I/O Signals:

   a. ACT: Parallel I/O activate signal to S channel, used by the programmers to activate the status registers of the S channel on the logic 1 to logic 0 transition.

   b. RDP: Parallel I/O read parallel signal to S channel, used by the programmer to read the channel status registers on the logic 1 to logic 0 transition.

   c. ME00-ME15: 16-bit status information and two 8-bit bytes of data from the S channel to the parallel I/O for transfer to the CPU E bus.

4. Channel — Memory Gate Signals: The signal interface between the S channel and the memory gate includes all the signals described in Section VIII, Memory Gate, for data bus 1, 2 or 3.

## CONNECTORS

All connectors are 56-pin ELCO 8016-056-000-007 connectors. The cables are 28 twisted-pair cables. Two cables are required between the S channel and the I/O device; two cables are required between the S and M channels; four cables are required between the S channel and the memory gate; and three cables are required between the S channel and parallel I/O.

## PHYSICAL AND ELECTRICAL CHARACTERISTICS

The S channel consists of four standard card files (SCC 365004-1), eleven cable connectors, nine cables, and a central +5-volt, 20-ampere power supply.

## ELECTRICAL CHARACTERISTICS

Power requirements for the S channel consist of 115 volts ac, at approximately 5 amperes per channel. Logic signal levels are +2.4 volts minimum to +5.0 volts maximum for a logic 1 and +0.4 volt maximum to -1.0 volt minimum for a logic 0. Input signals are terminated into 180 ohms to collector voltage (Vcc) and 270 ohms to ground. The channel output drivers are capable of driving 120-ohm twisted-pair cable terminated into matching impedance.

## INPUT/OUTPUT CONTROL

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PDS | f | |
| B | PDS | j | |
| C | | k | |
| D | | l | |
| E | | m | ACT |
| F | | n | ACT |
| H | | p | |
| J | | r | |
| K | ICS | s | |
| V | ICS | t | |
| L | | u | |
| M | | v | |
| N | | w | |
| P | | x | |
| R | | y | |
| S | | JJ | |
| T | RDPC | z | |
| U | RDPC | AA | |
| W | | BB | |
| X | | CC | |
| Y | | DD | |
| Z | | EE | |
| a | | FF | |
| b | | HH | |
| c | WTPC | KK | |
| d | WTPC | LL | |
| e | | MM | DRDY |
| h | | NN | DRDY |

*Figure 4-4. Connectors (Sheet 1 of 11)*

## PARALLEL INPUT

J02

| PIN | SIGNAL |
|---|---|
| A | PDI00 |
| B | PDIOO |
| C | PDI01 |
| D | PDI01 |
| E | PDI02 |
| F | PDI02 |
| H | PDI03 |
| J | PDI03 |
| K | PDI04 |
| V | PDIO4 |
| L | PDI05 |
| M | PDIO5 |
| N | PDI06 |
| P | PDI06 |
| R | PDI07 |
| S | PDI07 |
| T | PDI08 |
| U | PDI08 |
| W | PDI09 |
| X | PDI09 |
| Y | PDI10 |
| Z | PDI10 |
| a | PDI11 |
| b | PDI11 |
| c | PDI12 |
| d | PDI12 |
| e | PDI13 |
| h | PDI13 |

| PIN | SIGNAL |
|---|---|
| f | PDI14 |
| j | PDI14 |
| k | PDI15 |
| l | PDI15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 4-4. Cont'd (Sheet 2 of 11)*

## PARALLEL OUTPUT

J03

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PD00 | f | PD14 |
| B | PD00 | j | PD14 |
| C | PD01 | k | PD15 |
| D | PD01 | l | PD15 |
| E | PD02 | m | |
| F | PD02 | n | |
| H | PD03 | p | |
| J | PD03 | r | |
| K | PD04 | s | |
| V | PD04 | t | |
| L | PD05 | u | |
| M | PD05 | v | |
| N | PD06 | w | |
| P | PD06 | x | |
| R | PD07 | y | |
| S | PD07 | JJ | |
| T | PD08 | z | |
| U | PD08 | AA | |
| W | PD09 | BB | |
| X | PD09 | CC | |
| Y | PD10 | DD | |
| Z | PD10 | EE | |
| a | PD11 | FF | |
| b | PD11 | HH | |
| c | PD12 | KK | |
| d | PD12 | LL | |
| e | PD13 | MM | |
| h | PD13 | NN | |

*Figure 4-4. Cont'd (Sheet 3 of 11)*

INPUT/OUTPUT DATA & ADDRESS
FROM CPU OR PRIOR I/O MODULE

J09

| PIN | SIGNAL |
|---|---|
| A | CMD0 |
| B | CMD0 |
| C | CMD1 |
| D | CMD1 |
| E | CMD2 |
| F | CMD2 |
| H | CMD3 |
| J | CMD3 |
| K | ODAT0 |
| V | ODAT0 |
| L | ODAT1 |
| M | ODAT1 |
| N | ODAT2 |
| P | ODAT2 |
| R | ODAT3 |
| S | ODAT3 |
| T | ODAT4 |
| U | ODAT4 |
| W | ODAT5 |
| X | ODAT5 |
| Y | ODAT6 |
| Z | ODAT6 |
| a | ODAT7 |
| b | ODAT7 |
| c | OPAR |
| d | OPAR |
| e | DVAD0 |
| h | DVAD0 |

| PIN | SIGNAL |
|---|---|
| f | DVAD1 |
| j | DVAD1 |
| k | DVAD2 |
| l | DVAD2 |
| m | DVAD3 |
| n | DVAD3 |
| p | DVAD4 |
| r | DVAD4 |
| s | DVAD5 |
| t | DVAD5 |
| u | IDAT0 |
| v | IDAT0 |
| w | IDAT1 |
| x | IDAT1 |
| y | IDAT2 |
| JJ | IDAT2 |
| z | IDAT3 |
| AA | IDAT3 |
| BB | IDAT4 |
| CC | IDAT4 |
| DD | IDAT5 |
| EE | IDAT5 |
| FF | IDAT6 |
| HH | IDAT6 |
| KK | IDAT7 |
| LL | IDAT7 |
| MM |  |
| NN |  |

*Figure 4-4. Cont'd (Sheet 4 of 11)*

INPUT/OUTPUT CONTROLS
FROM CPU OR PRIOR I/O MODULE

J10

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | IOS | f | |
| B | IOS | j | |
| C | MCLR | k | |
| D | MCLR | l | |
| E | ACK | m | |
| F | ACK | n | |
| H | SRVRQ | p | |
| J | SRVRQ | r | |
| K | | s | |
| V | | t | |
| L | IPAR | u | |
| M | IPAR | v | |
| N | | w | |
| P | | x | |
| R | | y | |
| S | | JJ | |
| T | | z | |
| U | | AA | |
| W | | BB | |
| X | | CC | |
| Y | | DD | |
| Z | | EE | |
| a | | FF | |
| b | | HH | |
| c | | KK | |
| d | | LL | |
| e | | MM | |
| h | | NN | |

*Figure 4-4. Cont'd (Sheet 5 of 11)*

## INPUT/OUTPUT DATA & ADDRESS
## TO NEXT I/O MODULE

J11

| PIN | SIGNAL |
|---|---|
| A | CMD0 |
| B | CMD0 |
| C | CMD1 |
| D | CMD1 |
| E | CMD2 |
| F | CMD2 |
| H | CMD3 |
| J | CMD3 |
| K | ODAT0 |
| V | ODAT0 |
| L | ODAT1 |
| M | ODAT1 |
| N | ODAT2 |
| P | ODAT2 |
| R | ODAT3 |
| S | ODAT3 |
| T | ODAT4 |
| U | ODAT4 |
| W | ODAT5 |
| X | ODAT5 |
| Y | ODAT6 |
| Z | ODAT6 |
| a | ODAT7 |
| b | ODAT7 |
| c | OPAR |
| d | OPAR |
| e | DVAD0 |
| h | DVAD0 |

| PIN | SIGNAL |
|---|---|
| f | DVAD1 |
| j | DVAD1 |
| k | DVAD2 |
| l | DVAD2 |
| m | DVAD3 |
| n | DVAD3 |
| p | DVAD4 |
| r | DVAD4 |
| s | DVAD5 |
| t | DVAD5 |
| u | IDAT0 |
| v | IDAT0 |
| w | IDAT1 |
| x | IDAT1 |
| y | IDAT2 |
| JJ | IDAT2 |
| z | IDAT3 |
| AA | IDAT3 |
| BB | IDAT4 |
| CC | IDAT4 |
| DD | IDAT5 |
| EE | IDAT5 |
| FF | IDAT6 |
| HH | IDAT6 |
| KK | IDAT7 |
| LL | IDAT7 |
| MM | |
| NN | |

*Figure 4-4. Cont'd (Sheet 6 of 11)*

## INPUT/OUTPUT CONTROLS
## TO NEXT I/O MODULE

J12

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | IOS | f | |
| B | IOS | j | |
| C | MCLR | k | |
| D | MCLR | l | |
| E | ACK | m | |
| F | ACK | n | |
| H | SRVRQ | p | |
| J | SRVRQ | r | |
| K | XFRRQ | s | |
| V | XFRRQ | t | |
| L | IPAR | u | |
| M | IPAR | v | |
| N | | w | |
| P | | x | |
| R | | y | |
| S | | JJ | |
| T | | z | |
| U | | AA | |
| W | | BB | |
| X | | CC | |
| Y | | DD | |
| Z | | EE | |
| a | | FF | |
| b | | HH | |
| c | | KK | |
| d | | LL | |
| e | | MM | |
| h | | NN | |

*Figure 4-4. Cont'd (Sheet 7 of 11)*

## MEMORY GATE ADDRESS

J13

| PIN | SIGNAL |
|---|---|
| A | RS00 |
| B | RS00 |
| C | RS01 |
| D | RS01 |
| E | RS02 |
| F | RS02 |
| H | RS03 |
| J | RS03 |
| K | RS04 |
| V | RS04 |
| L | RS05 |
| M | RS05 |
| N | RS06 |
| P | RS06 |
| R | RS07 |
| S | RS07 |
| T | RS08 |
| U | RS08 |
| W | RS09 |
| X | RS09 |
| Y | RS10 |
| Z | RS10 |
| a | RS11 |
| b | RS11 |
| c | RS12 |
| d | RS12 |
| e | RS13 |
| h | RS13 |
| f | RS14 |
| j | RS14 |
| k | RS15 |
| l | RS15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 4-4. Cont'd (Sheet 8 of 11)*

MEMORY GATE
OUTPUT DATA

J14

| PIN | SIGNAL |
|---|---|
| A | MMS00 |
| B | MMS00 |
| C | MMS01 |
| D | MMS01 |
| E | MMS02 |
| F | MMS02 |
| H | MMS03 |
| J | MMS03 |
| K | MMS04 |
| V | MMS04 |
| L | MMS05 |
| M | MMS05 |
| N | MMS06 |
| P | MMS06 |
| R | MMS07 |
| S | MMS07 |
| T | MMS08 |
| U | MMS08 |
| W | MMS09 |
| X | MMS09 |
| Y | MMS10 |
| Z | MMS10 |
| a | MMS11 |
| b | MMS11 |
| c | MMS12 |
| d | MMS12 |
| e | MMS13 |
| h | MMS13 |

| PIN | SIGNAL |
|---|---|
| f | MMS14 |
| j | MMS14 |
| k | MMS15 |
| l | MMS15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 4-4. Cont'd (Sheet 9 of 11)*

MEMORY GATE
INPUT DATA

J15

| PIN | SIGNAL |
|---|---|
| A | MDS00 |
| B | MDS00 |
| C | MDS01 |
| D | MDS01 |
| E | MDS02 |
| F | MDS02 |
| H | MDS03 |
| J | MDS03 |
| K | MDS04 |
| V | MDS04 |
| L | MDS05 |
| M | MDS05 |
| N | MDS06 |
| P | MDS06 |
| R | MDS07 |
| S | MDS07 |
| T | MDS08 |
| U | MDS08 |
| W | MDS09 |
| X | MDS09 |
| Y | MDS10 |
| Z | MDS10 |
| a | MDS11 |
| b | MDS11 |
| c | MDS12 |
| d | MDS12 |
| e | MDS13 |
| h | MDS13 |
| f | MDS14 |
| j | MDS14 |
| k | MDS15 |
| l | MDS15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 4-4. Cont'd (Sheet 10 of 11)*

MEMORY GATE
CONTROL

J16

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | CA | f | |
| B | CA | j | |
| C | DA | k | |
| D | DA | l | |
| E | DRMACK | m | |
| F | DRMACK | n | |
| H | RQ | p | |
| J | RQ | r | |
| K | RR | s | |
| V | RR | t | |
| L | DRM | u | |
| M | DRM | v | |
| N | | w | |
| P | | x | |
| R | | y | |
| S | | JJ | |
| T | | z | |
| U | | AA | |
| W | | BB | |
| X | | CC | |
| Y | FIN | DD | |
| Z | FIN | EE | |
| a | | FF | |
| b | | HH | |
| c | | KK | |
| d | | LL | |
| e | | MM | |
| h | | NN | |

*Figure 4-4 Cont'd (Sheet 11 of 11)*

# SECTION V

# PARALLEL INPUT/OUTPUT INTERFACE

## BASIC DESCRIPTION

The parallel I/O interface provides the SCC 4700 system with the capability of transferring full word (16-bit) data between memory and an I/O device. The capability is expanded to include communication with several devices by adding the optional digital multiplex unit described in Section VI. A general block diagram of the parallel I/O Interface is shown in Figure 5-1. Data from the CPU M register is tranferred to the optional memory map or to the I/O device (or devices) under control of the CPU microcontroller logic and the CPU Z register control output. A Device Ready (DRDY) control signal from the I/O device to the CPU microcontroller must enable the CPU condition logic before data transfer can occur. If the CPU condition logic is enabled by the DRDY control signal from the I/O device, data transfer occurs from the CPU M register to the I/O device on a Read Parallel (RDP) instruction, and from the I/O device to the CPU E bus on a Write Parallel (WTP) instruction. Systems with more than one parallel I/O device require the Activate (ACT) instruction to address and activate a specific device.

## THEORY OF OPERATION

The parallel I/O data signals and parallel I/O control signals are shown in Figures 5-2 and 5-3, respectively. A typical data and control interface (CPU to I/O device) configuration is shown in Figure 5-6 at the end of this section. As necessary, refer to the timing diagrams in Figure 5-4 and Figure 5-5.

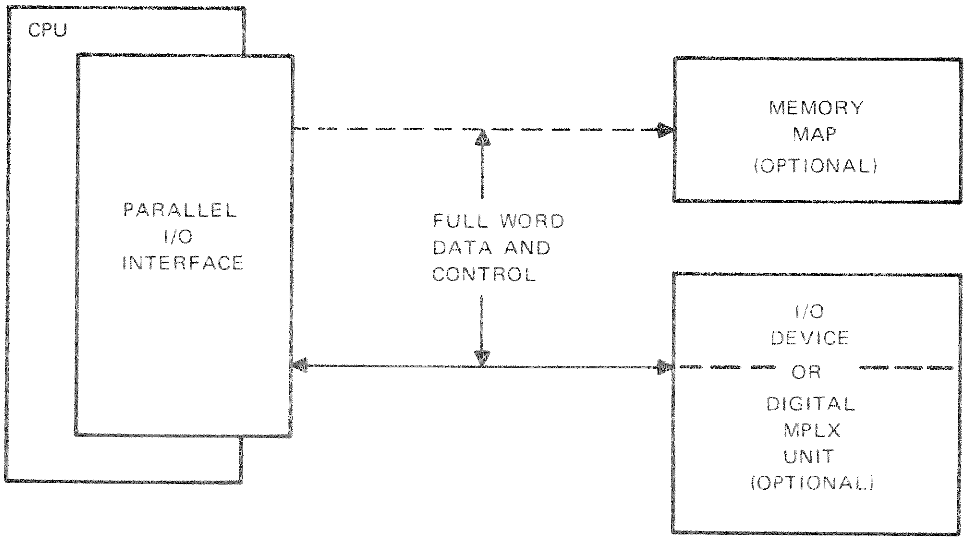


*Figure 5-1. Parallel I/O Interface Configuration*

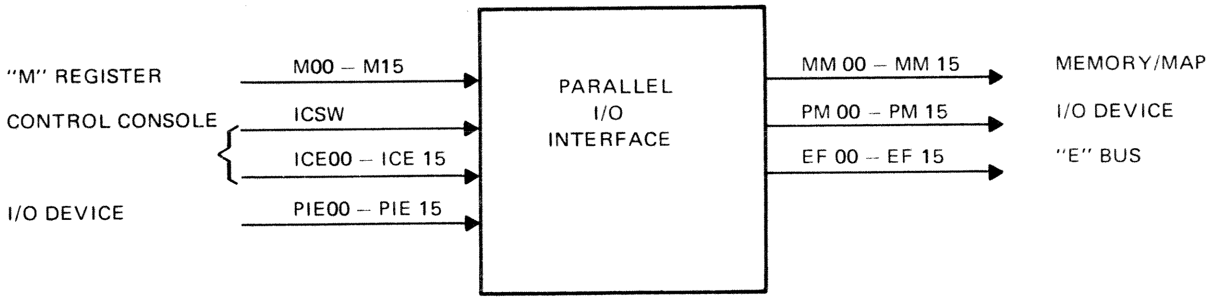


*Figure 5-2. Data Signals*

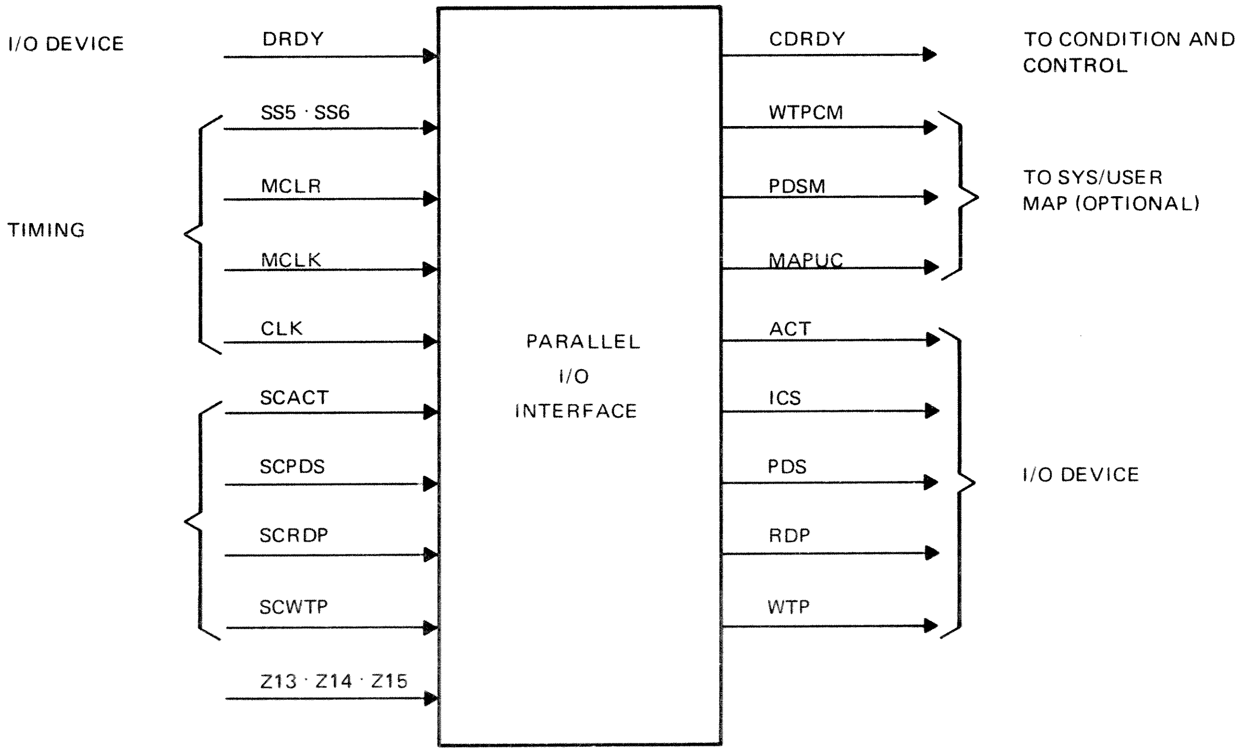


*Figure 5-3. Control Signals*

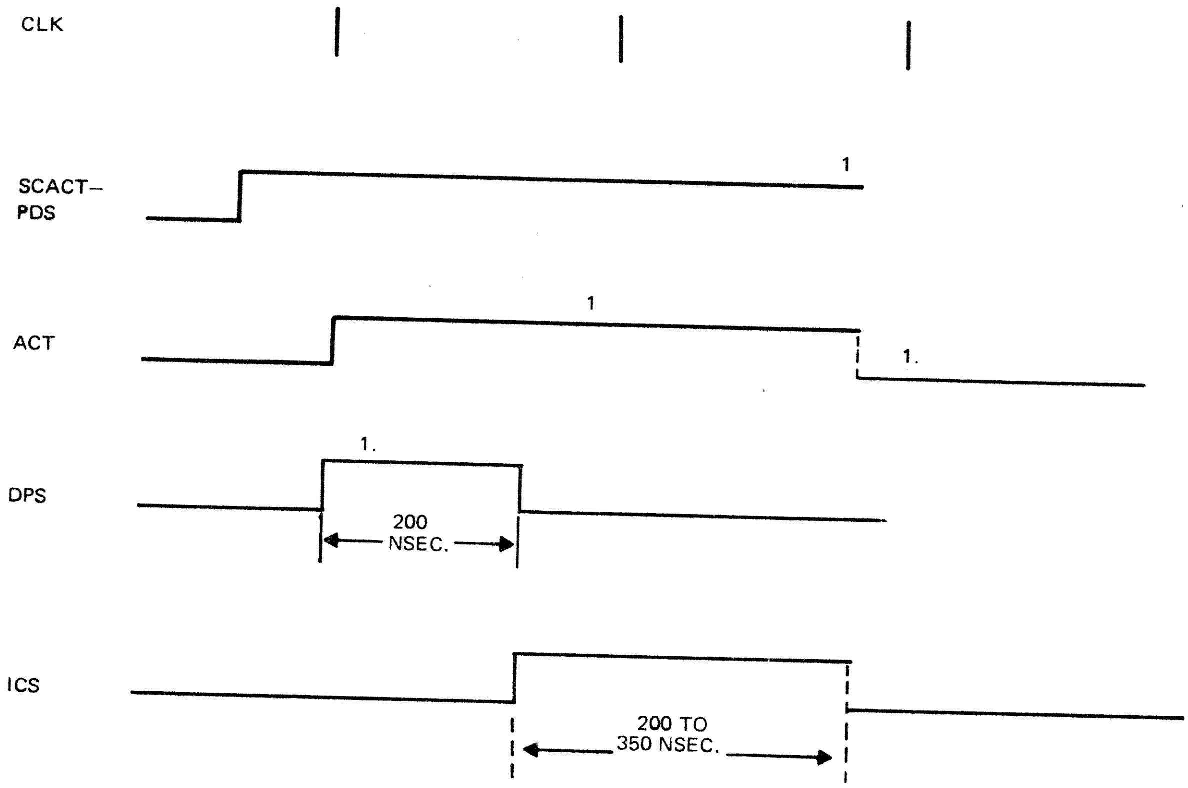
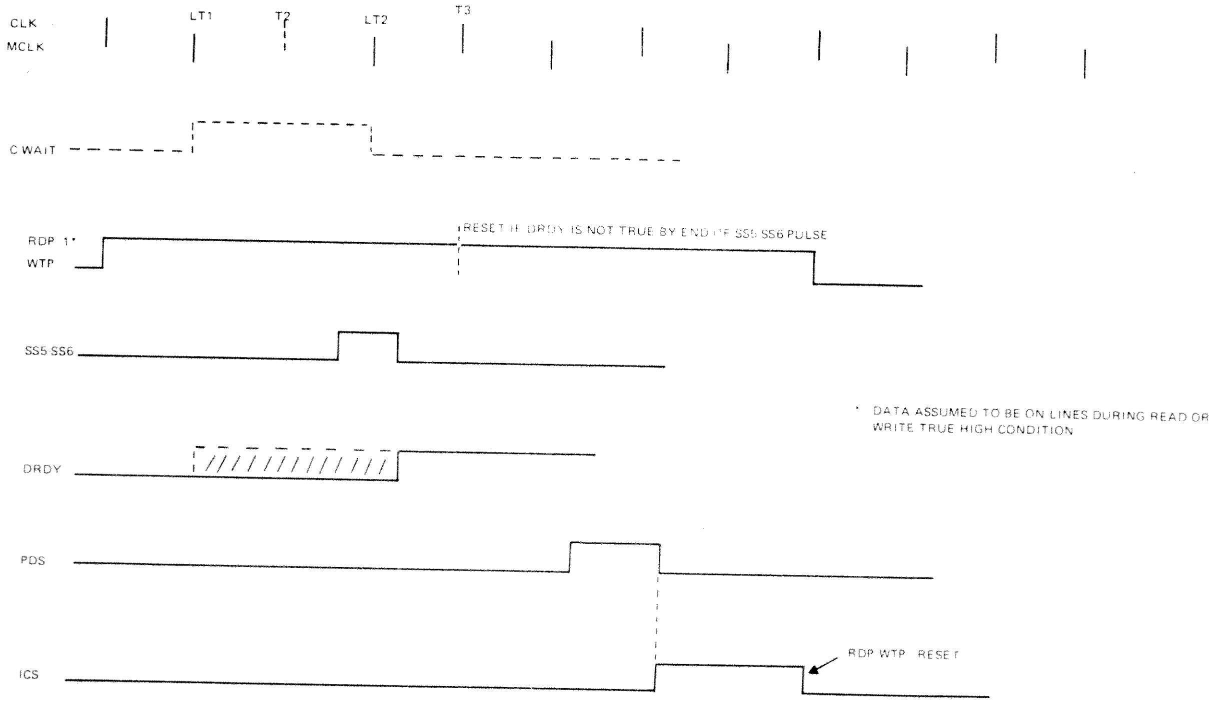


*Figure 5-4. Activate*



* DATA ASSUMED TO BE ON LINES DURING READ OR WRITE TRUE HIGH CONDITION

*Figure 5-5. Read/Write*

To activate a device, the CPU enables the interface activate and data strobe control flip-flops with the SCACT and SCPDS signals. At the next CLK pulse the two flip-flops are set. The CPU then drops all the Z bus logic signals to true low, allowing the ACT and PDS control signals to be gated to the I/O device. At the next CLK pulse PDS is reset, setting ICS. The activate flip-flop is reset at the end of ICS. During read or write operations, the respective flip-flops are set as previously described. However, the Device Ready (DRDY) signal must be present at the time signals SS5·SS6 appear; otherwise, the set read or write flip-flop will be reset by the DRDY flip-flop output at the next clock pulse. The RDPC and WTPC control signals are also gated to the I/O device by the low Z bus logic. The WTPC and RDP control signals may be routed to the optional memory map section by the Z bus control logic. Under those conditions, logic signal Z15 is used as the USER/SYSTEM control signal. During a Parallel I/O Write (WTPC) operation, the 16-bit data in the CPU M register (M00 through M15) is strobed out at PDSC time and routed out to the I/O device (PM00 through PM15) by the Z bus low true logic. During the optional Memory Map Write (WTPM) operation, the M register data is strobed out by the PDSM signal onto the memory map lines (M00 through MM15). The I/O device data (PIE00 through PIE15) is strobed out of the device by the PDSC signal during a Parallel I/O Read (RDPC) operation. An RDPC output is taken from the read parallel control flip-flop to enable the PIE output gates. The data is routed by the output gates onto the CPU E bus lines (EF00 through EF15). These output gates are time shared with the channel I/O device outputs (MEO through ME07), the console switch outputs (ICE00 through ICE15), and the optional map table pointers (MPTE00 through MPTE15).

## SIGNAL

### Input

1. M00-M15 – Input Data Bits: Input data from the M register should remain stable while SC RDPC or SC WTPC is at logic 1.*

2. ICE00-ICE15 – Console Input Bits: Manually inserted data from the Control Console; must be enabled by the ISCW console switch input.*

3. PIE00-PIE15 – I/O Data Bits: Input data from the I/O devices must be gated by Z13 · Z14 · Z15 control logic signals.

4. MCLR – Master Clear: A clock pulse used to reset all parallel I/O logic.*

5. CLK – Clock: Main I/O timing reference pulses (approximately 300 nanoseconds).*

6. MCLK – Matrix Clock: A late clock pulse occurring approximately 150 nanoseconds after main clock. The CWAIT signal, used to inhibit the main clock pulse, lasts from one late clock pulse to the succeeding late clock pulse.*

7. SCACT – Special Control Activate: Sets the ACT logic of the parallel I/O on the logic 0 to logic 1 transition.*

* Internal signals listed for reference.

8. SCPDS – Special Control Parallel Data Strobe: Sets the PDS logic of the parallel I/O and strobes the data to/from the I/O device on the logic 0 to logic 1 transition.*

9. SCRDP – Special Control Read Parallel: Sets the RDP logic of the parallel I/O to initiate a read parallel operation on the logic 0 to logic 1 transition.*

10. SCWTP – Special Control Write Parallel: Sets the WTP logic of the parallel I/O to initiate a write parallel operation on the logic 0 to logic 1 transition.*

11. DRDY – Device Ready: Input control signal from the I/O device indicating that the device is ready on the logic 0 to logic 1 transition.

12. Z13 · Z14 · Z15 – Z Bus Logic: Input logic control signals. These signals gate memory data through the parallel I/O when all three signals are at logic 0. The signals are also used to gate optional memory map control signals.*

**Outputs**

1. MM00-MM15 – Output Data Bits: Data signals originating at the CPU M register, which are buffered and passed along ungated by the interface to the optional memory map section. The signals are true at logic 1.*

2. PM00-PM15 – Output Data Bits: Output data signals to the I/O device from memory. The signals are true at logic 0 and are gated out by the Z13 · Z14 · Z15 control condition.

3. EF00-EF15 – Output Data Bits: Output data signals to the CPU E bus. The signals are true at logic 1.*

4. CDRDY – Condition Device Ready: Output device ready logic 1 signal used by the CPU condition logic.*

5. WTP – Write Parallel: Output write control signal to I/O device initiates the write operation on the logic 1 to logic 0 transition.

6. RDP – Read Parallel: Output read control signal to I/O device initiates the read operation on the logic 1 to logic 0 transition.

7. ACT – Activate: Output activate control signal to I/O device activates the device on the logic 1 to logic 0 transition, indicating a read or write operation will follow.

8. PDS – Parallel Data Strobe: Output data strobe control signal to I/O device strobes data into or out of the I/O device on the logic 1 to logic 0 transition. This signal is present for 200 nanoseconds.

* Internal signals listed for reference.

9. ICS – Instruction Complete Strobe: Output instruction complete signal to I/O device makes a logic 1 to logic 0 transition at the end of PDS, lasts 200 to 300 nanoseconds and resets ACT, WTP or RDP on the logic 0 to logic 1 transition.

10. WTPCM – WTP Control: Control signal to map goes true on the logic 0 to logic 1 transition.*

11. PDSM – PDS Control: Control signal to map goes true on the logic 0 to logic 1 transition.*

12. MAPUC – Map User Control: Control signal to map, used to select system or user mode. The user mode is selected when the signal is a logic 1 and system mode is selected when the signal is a logic 0.*

## TIMING DIAGRAM

The timing relationship of the Parallel I/O interface signals during the ACT instruction is shown in Figure 5-4. Signal timing relationship during RDP and WTP operations is shown in Figure 5-5. The microcontroller CWAIT signal is not an interface signal, but is included to show CPU control of the timing pulses.

## CABLES AND CONNECTORS

Three cables are required to connect the parallel I/O Interface to the I/O device or digital multiplex units. The cable used to transfer CPU data from the parallel I/O Interface to the I/O device is designated cable No. 1. The cable used to transfer data from the I/O device to the parallel I/O is designated cable No. 2. Control signals are transferred between the I/O device and the Interface on cable No. 3. All three cables shall be made up of twisted pairs, and the maximum length of the cables shall not exceed 30 feet. A listing of the interface connectors and connector pin numbers with the associated signals is given in figure 5-6. All signal signatures ending in LO indicate the ground side of the signal.

* Internal signals listed for reference.

PARALLEL INPUT/OUTPUT

PARALLEL OUTPUT

CABLE 1

J15

| PIN | SIGNAL |
|---|---|
| A | PM00 |
| B | PM00 |
| C | PM01 |
| D | PM01 |
| E | PM02 |
| F | PM02 |
| H | PM03 |
| J | PM03 |
| K | PM04 |
| V | PM04 |
| L | PM05 |
| M | PM05 |
| N | PM06 |
| P | PM06 |
| R | PM07 |
| S | PM07 |
| T | PM08 |
| U | PM08 |
| W | PM09 |
| X | PM09 |
| Y | PM10 |
| Z | PM10 |
| a | PM11 |
| b | PM11 |
| c | PM12 |
| d | PM12 |
| e | PM13 |
| h | PM13 |

| PIN | SIGNAL |
|---|---|
| f | PM14 |
| j | PM14 |
| k | PM15 |
| l | PM15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 5-6. Connectors (Sheet 1 of 3)*

PARALLEL INPUT/OUTPUT
PARALLEL INPUT
CABLE 2

J13

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PIE00 | f | PIE14 |
| B | PIE00 | j | PIE14 |
| C | PIE01 | k | PIE15 |
| D | PIE01 | l | PIE15 |
| E | PIE02 | m | |
| F | PIE02 | n | |
| H | PIE03 | p | |
| J | PIE03 | r | |
| K | PIE04 | s | |
| V | PIE04 | t | |
| L | PIE05 | u | |
| M | PIE05 | v | |
| N | PIE06 | w | |
| P | PIE06 | x | |
| R | PIE07 | y | |
| S | PIE07 | JJ | |
| T | PIE08 | z | |
| U | PIE08 | AA | |
| W | PIE09 | BB | |
| X | PIE09 | CC | |
| Y | PIE10 | DD | |
| Z | PIE10 | EE | |
| a | PIE11 | FF | |
| b | PIE11 | HH | |
| c | PIE12 | KK | |
| d | PIE12 | LL | |
| e | PIE13 | MM | |
| h | PIE13 | NN | |

*Figure 5-6. Cont'd (Sheet 2 of 3)*

PARALLEL INPUT/OUTPUT
INPUT/OUTPUT CONTROL
CABLE 3

J11

| PIN | SIGNAL |
|---|---|
| A | PDS |
| B | PDS |
| C | |
| D | |
| E | |
| F | |
| H | |
| J | |
| K | ICS |
| V | ICS |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | RDPC |
| U | RDPC |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | WTPC |
| d | WTPC |
| e | |
| h | |

| PIN | SIGNAL |
|---|---|
| f | |
| j | |
| k | |
| l | |
| m | ACT |
| n | ACT |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | DRDY |
| NN | DRDY |

*Figure 5-6. Cont'd (Sheet 3 of 3)*

# SECTION VI
# DIGITAL MULTIPLEX INTERFACE

## OPERATION

The digital multiplex interface provides a means of expanding the I/O capability of the CPU via the parallel I/O section. A digital multiplex unit expands the one peripheral device connection of the parallel I/O to four connections. Several multiplex units may be connected in a pass-along configuration, tree configuration, or in combinations of both to expand the I/O connections for handling up to 64 devices. A typical expanded configuration is shown in Figure 6-1. The pass-along feature of the units allows control information and data to be passed through units with nonactive devices to or from units with an active device. Only one device may be active at any one time. Devices are activated by priority or when addressed if a no priority condition exists. The digital multiplex unit consists of an input and an output section. Input and output data flow are shown in Figure 6-2. The output section receives CPU output data either from the CPU or from a preceding unit. If the data is addressed to one of its devices, it is passed to the device; otherwise, the data is passed along to the succeeding unit. In the output section data is received from one of the devices, if a device is active, or from a preceding unit. The received data is passed to the CPU parallel I/O directly or through a succeeding unit. The multiplex unit decodes 6 address bits (included in the second word of the ACT instruction) and uses the decoded address to select one of four outputs. A block diagram of the address decode logic is shown in Figure 6-3. The Two LSB are hardwired in a fixed pattern, while the four MSB are selected through the four MSB decode logic to generate a decode enable signal for the LSB pattern. Each of the four MSB may be selected to generate the decode enble signal on a zero, one, or "don't care" state of each bit.

A control signal flow diagram of the multiplex unit is shown in Figure 6-4. Signals ACT, ICS, and PDS, originating at the CPU, are present at all multiplex units whenever the CPU sends these signals. WTP and RDP signals are selectively gated to the I/O devices, if the gating is set up by the six address bits, each time ACT and PDS strobes are received by the multiplex unit. Device Ready (DROC) signals and all data from the devices are selectively gated to the CPU. All inputs to the multiplex unit are terminated in a resistor combination of 180 ohms to +5 volts dc and 270 ohms to ground. In addition to this termination, each input shall be connected to two TTL loads. Multiplex unit input and output circuits shall be driven by Sylvania SG 132 or Fairchild 9009. Some delay is introduced to the control and data signals as indicated in the single flow block diagrams. The delays indicated are based on an output capacity equivalent to 6 to 12 feet of cable. When using several multiplex units, the combined total delay must be considered to prevent system malfunction.

## THEORY OF OPERATION

A typical digital multiplex data and control signal interface is shown in Figure 6-6 at the end of this section. (Reference may be made, as necessary, to the timing diagram shown in Figure 6-5 and to the parallel I/O timing diagram in Section V.) The 16-bit ICMP (0-15) data input from the CPU is strobed into the multiplex unit by the ACT and PDS control signals. The data and control signals are sent ungated, though delayed, to the unit I/O devices and also to succeeding multiplex

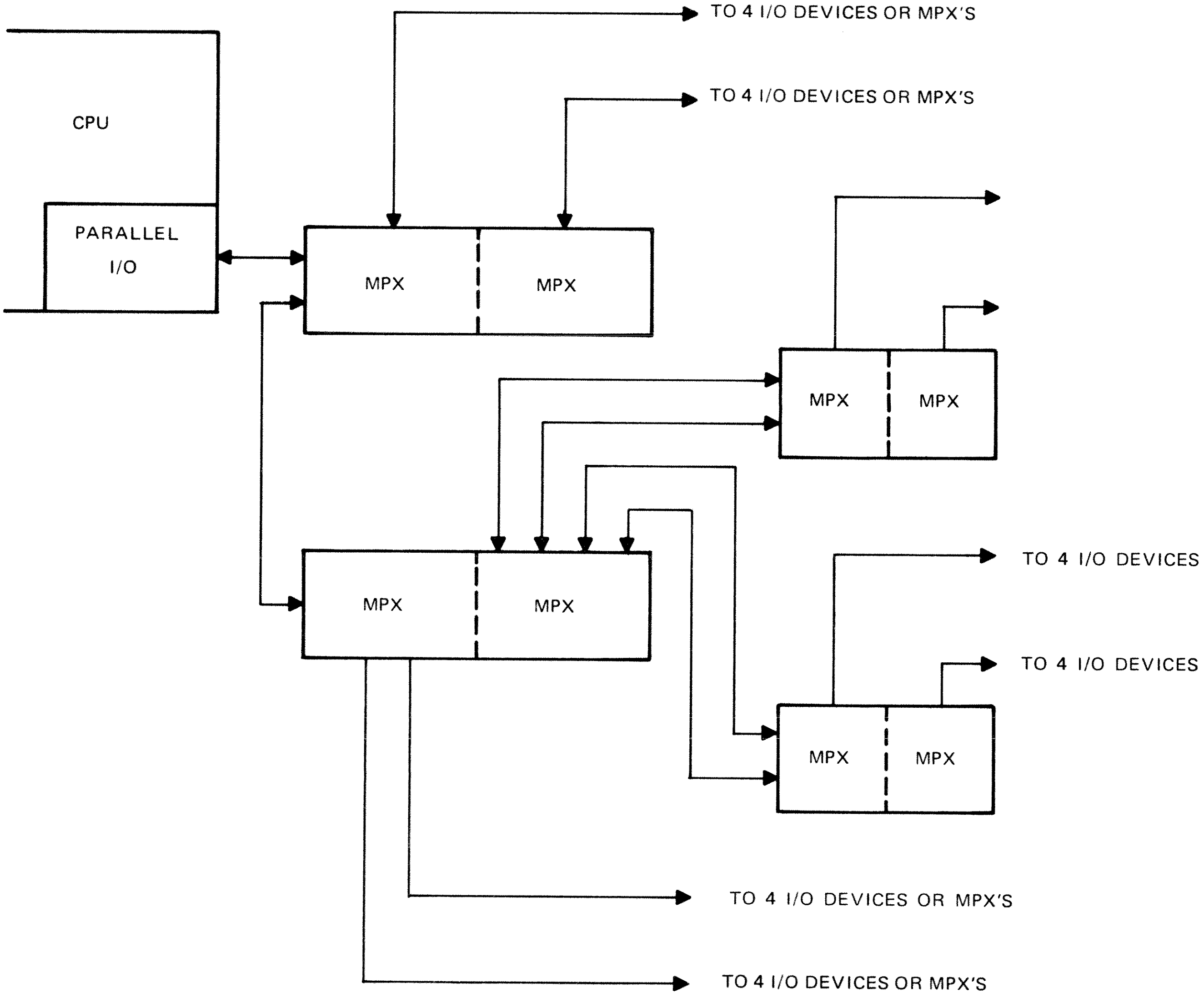

TO 4 I/O DEVICES OR MPX'S
TO 4 I/O DEVICES OR MPX'S
CPU
PARALLEL I/O
MPX
MPX
MPX
MPX
MPX
MPX
TO 4 I/O DEVICES
TO 4 I/O DEVICES
MPX
MPX
TO 4 I/O DEVICES OR MPX'S
TO 4 I/O DEVICES OR MPX'S

*Figure 6-1. Typical Expander Parallel I/O Capability — Configuration*

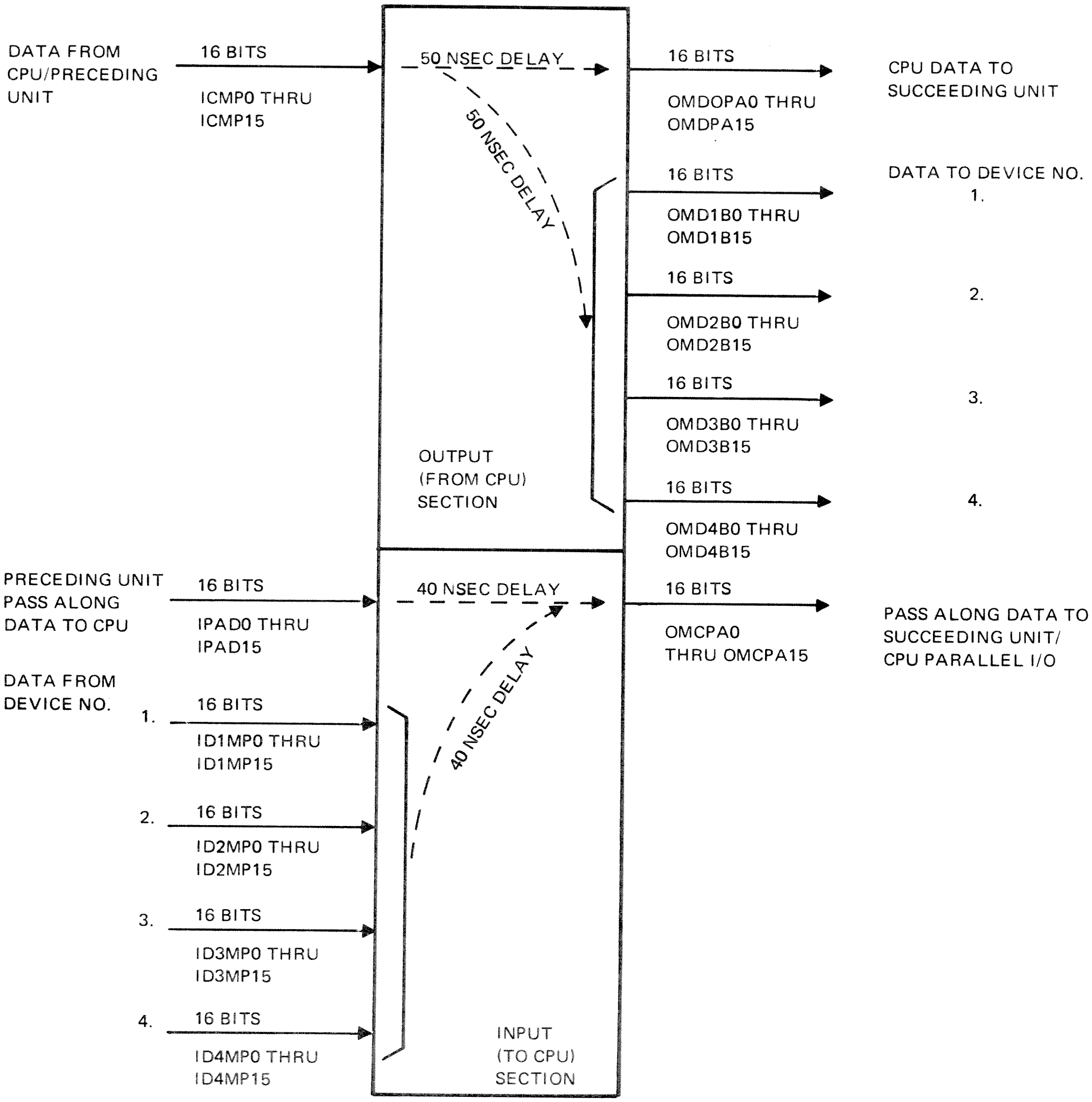


*Figure 6-2. Digital Multiplex Unit Data Flow*

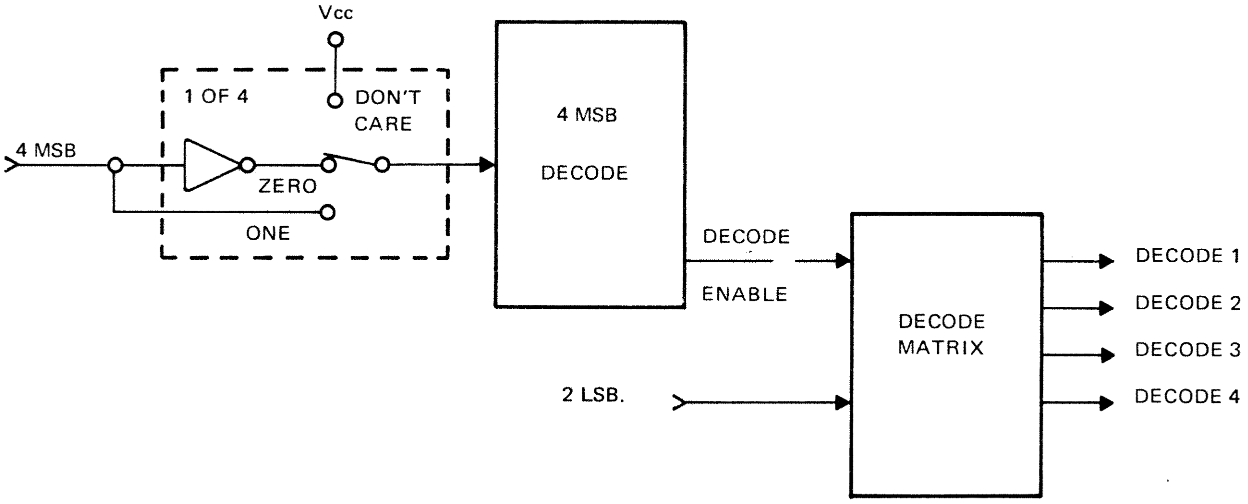


*Figure 6-3. Multiplex Unit Address Decode Block Diagram*

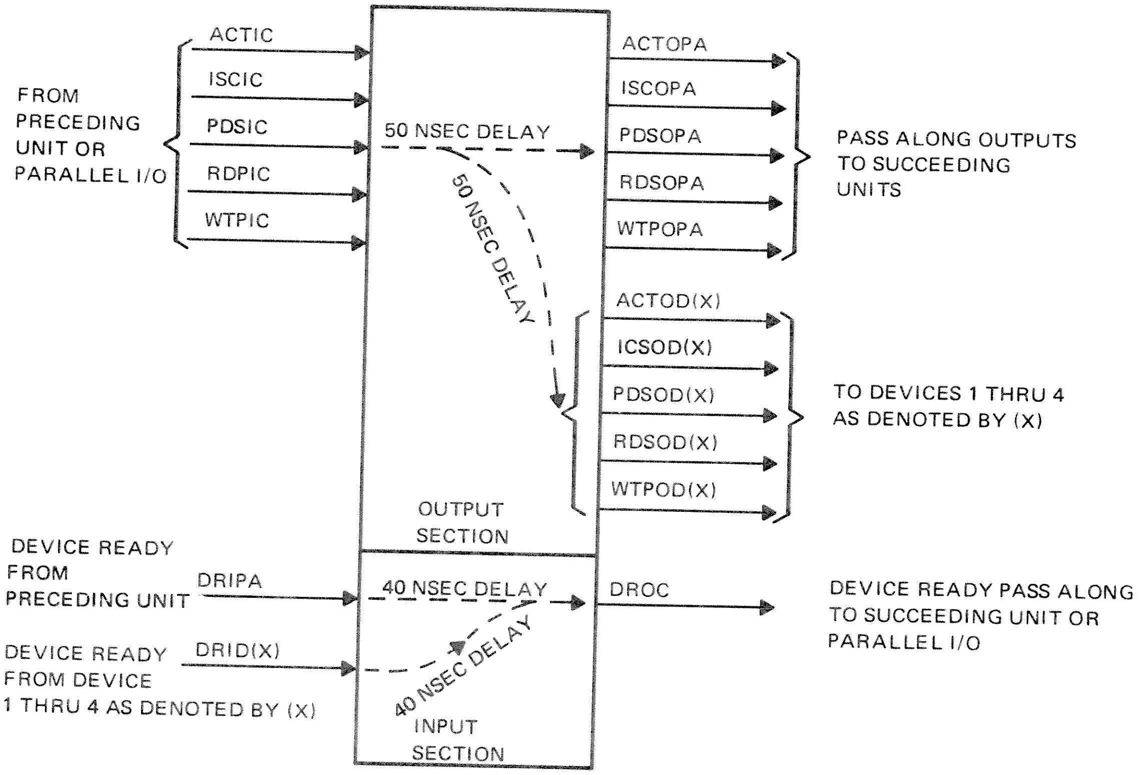

FROM PRECEDING UNIT OR PARALLEL I/O
ACTIC
ISCIC
PDSIC
RDPIC
WTPIC
50 NSEC DELAY
50 NSEC DELAY
ACTOPA
ISCOPA
PDSOPA
RDSOPA
WTPOPA
PASS ALONG OUTPUTS TO SUCCEEDING UNITS
ACTOD(X)
ICSOD(X)
PDSOD(X)
RDSOD(X)
WTPOD(X)
TO DEVICES 1 THRU 4 AS DENOTED BY (X)
OUTPUT SECTION
DEVICE READY FROM PRECEDING UNIT
DRIPA
40 NSEC DELAY
DROC
DEVICE READY PASS ALONG TO SUCCEEDING UNIT OR PARALLEL I/O
DEVICE READY FROM DEVICE 1 THRU 4 AS DENOTED BY (X)
DRID(X)
40 NSEC DELAY
INPUT SECTION

*Figure 6-4. Digital Multiplex Unit – Control Signal Flow Diagram*

units (each succeeding unit will introduce additional signal delay) until the next ACT and DPS strobes. The WTP and RDP signals are also sent ungated as pass-along signals to the next unit. When the decode enable logic has been addressed, the WTP and RDP signals are gated to the addressed unit device. When addressed, the decode logic also enables the device ready and device data gate to allow device outputs to be passed on to the CPU or to the next unit. When not addressed, the unit will pass along the device ready and device data signals received from the preceding unit.

## TIMING DIAGRAM

The timing relationship between the multiplex unit input and output control and data signals is shown in Figure 6-5.

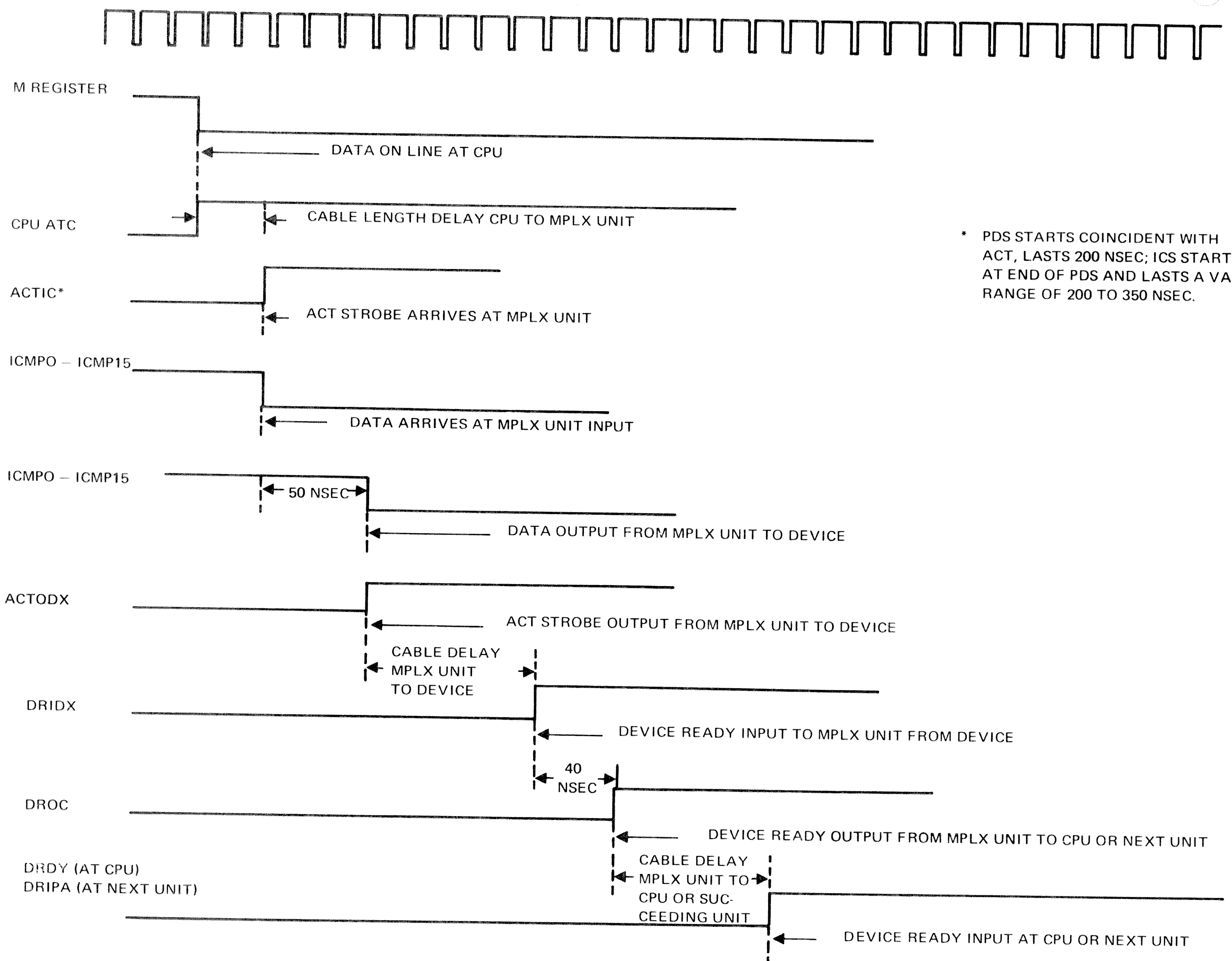
## DIGITAL MULTIPLEX INTERFACE SIGNALS

### Input

1. ICMP (0-15) – 16 Input Data Lines: Carry data bits 0-15 from the CPU or from a preceding multiplex unit. Low is true.
2. IPAD (0-15) – 16 Pass-Along Lines: 16 bits of pass-along data, bits 0 through 15 from a preceding unit to be passed to the CPU parallel I/O section or to a succeeding unit. Low is true.
3. ID (0-3) MP (0-15) – 64 Device Data Bits: 64 bits of device data to the multiplex unit (16 bits per device). The ID number is the device number, while the MP number denotes the bit number.
4. ACTIC – Activate Control Line: Activate Parallel I/O strobe from the CPU. True is high.
5. ICSIC – Instruction Complete Control Line: Instruction complete strobe from the CPU.
6. PDSIC – Parallel Data Control Line: Parallel data strobe from the CPU.
7. RDPIC – Read Parrallel Control Line: Read parallel strobe from the CPU.
8. WTPIC – Write Parallel Control Line: Write parallel strobe from the CPU.
9. DRID(0-4) – Four Device Ready Lines: Device ready input to multiplex unit from device (1-4) indicated. True is high.
10. DRIPA – Device Ready Strobe Line: Strobe input to multiplex unit from pass-along section of a preceding unit.

### Output

1. OMD (0-3) B (0-15) – 64 Data Lines: 64 bits of data from the multiplex unit to the devices (16 bits to each device). The OMD number is the device number (0-3) and the B number is the bit (0-15). Low is true.


M REGISTER
DATA ON LINE AT CPU
CPU ATC
CABLE LENGTH DELAY CPU TO MPLX UNIT
ACTIC*
ACT STROBE ARRIVES AT MPLX UNIT
ICMPO – ICMP15
DATA ARRIVES AT MPLX UNIT INPUT
ICMPO – ICMP15
50 NSEC
DATA OUTPUT FROM MPLX UNIT TO DEVICE
ACTODX
ACT STROBE OUTPUT FROM MPLX UNIT TO DEVICE
CABLE DELAY MPLX UNIT TO DEVICE
DRIDX
DEVICE READY INPUT TO MPLX UNIT FROM DEVICE
40 NSEC
DROC
DEVICE READY OUTPUT FROM MPLX UNIT TO CPU OR NEXT UNIT
CABLE DELAY MPLX UNIT TO CPU OR SUC-CEEDING UNIT
DRDY (AT CPU)
DRIPA (AT NEXT UNIT)
DEVICE READY INPUT AT CPU OR NEXT UNIT

* PDS STARTS COINCIDENT WITH ACT, LASTS 200 NSEC; ICS STARTS AT END OF PDS AND LASTS A VARIABLE RANGE OF 200 TO 350 NSEC.

*Figure 6-5. Multiplex Unit Signals – Timing Diagram*

2. OMDPA (0-15) – 16 CPU Data Lines: 16 bits of CPU data (0-15) output from the multiplex unit to a succeeding unit. These lines will connect to the ICMPX input of a succeeding unit. Low is true.

3. OMCPA (0-15) – 16 Output Data Lines: 16 bits of data (0-15) output from the multiplex unit to a succeeding unit or to the parallel I/O. These lines will connect to the IPADX input lines of a succeeding unit. Low is true.

4. ACTOD (0-3) – ACT Control Line: ACT output strobe from multiplex unit to device (1-4) indicated. High is true.

5. ICSOD (0-3) – ICS Control Line: ICS Output strobe from multiplex unit to device (1-4) indicated.

6. PDSOD (0-3) – PDS Control Line: PDS output strobe from multiplex unit to device (1-4) indicated.

7. WTPOD (0-3) – RDP Control Line: RDP output strobe from multiplex unit of device (1-4) indicated.

8. WTPOD (0-3) – WTP Control Line: WTP output strobe from multiplex unit to device (1-4) indicated.

9. DROC – Device Ready Strobe Line: DRDY strobe output from multiplex unit to the CPU or a succeeding unit. High is true.

10. ACTOPA – ACT Strobe Output. ACT pass-along output strobe from the multiplex unit to a succeeding unit.

11. ICSOPA – ICS Strobe Output: ICS pass-along output strobe from the multiplex unit to a succeeding unit.

12. PDSOPA – PDS Strobe Output: PDS pass-along output strobe form the multiplex unit to a succeeding unit.

13. RDSOPA – RDP Strobe Output: RDP pass-along output strobe from the multiplex unit to a succeeding unit.

14. WTPOPA – WTP Strobe Output: WTP pass-along output strobe from the multiplex unit to a succeeding unit.

## CONNECTORS

Signals are listed as they appear on the input and output connectors in Figure 6-6.

## INPUT/OUTPUT CONTROL FROM CPU

J05

| PIN | SIGNAL |
|---|---|
| A | PDS |
| B | PDS |
| C | |
| D | |
| E | |
| F | |
| H | |
| J | |
| K | ICS |
| V | ICS |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | RDPC |
| U | RDPC |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | WTPC |
| d | WTPC |
| e | |
| h | |

| PIN | SIGNAL |
|---|---|
| f | |
| j | |
| k | |
| l | |
| m | ACT |
| n | ACT |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | DRDY |
| NN | DRDY |

*Figure 6-6. Signal Listings (Sheet 1 of 18)*

PARALLEL INPUT
FROM CPU

J06

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PM00 | f | PM14 |
| B | PM00 | j | PM14 |
| C | PM01 | k | PM15 |
| D | PM01 | l | PM15 |
| E | PM02 | m | |
| F | PM02 | n | |
| H | PM03 | p | |
| J | PM03 | r | |
| K | PM04 | s | |
| V | PM04 | t | |
| L | PM05 | u | |
| M | PM05 | v | |
| N | PM06 | w | |
| P | PM06 | x | |
| R | PM07 | y | |
| S | PM07 | JJ | |
| T | PM08 | z | |
| U | PM08 | AA | |
| W | PM09 | BB | |
| X | PM09 | CC | |
| Y | PM10 | DD | |
| Z | PM10 | EE | |
| a | PM11 | FF | |
| b | PM11 | HH | |
| c | PM12 | KK | |
| d | PM12 | LL | |
| e | PM13 | MM | |
| h | PM13 | NN | |

*Figure 6-6 Cont'd (Sheet 2 of 18)*

## PARALLEL INPUT
## FROM PRECEDING DEVICE

J07

| PIN | SIGNAL |
|---|---|
| A | PIE00 |
| B | PIE00 |
| C | PIE01 |
| D | PIE01 |
| E | PIE02 |
| F | PIE02 |
| H | PIE03 |
| J | PIE03 |
| K | PIE04 |
| V | PIE04 |
| L | PIE05 |
| M | PIE05 |
| N | PIE06 |
| P | PIE06 |
| R | PIE07 |
| S | PIE07 |
| T | PIE08 |
| U | PIE08 |
| W | PIE09 |
| X | PIE09 |
| Y | PIE10 |
| Z | PIE10 |
| a | PIE11 |
| b | PIE11 |
| c | PIE12 |
| d | PIE12 |
| e | PIE13 |
| h | PIE13 |

| PIN | SIGNAL |
|---|---|
| f | PIE14 |
| j | PIE14 |
| k | PIE15 |
| l | PIE15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 3 of 18)*

INPUT/OUTPUT CONTROL
TO NEXT DEVICE

J08

| PIN | SIGNAL |
|---|---|
| A | PDS |
| B | PDS |
| C | |
| D | |
| E | |
| F | |
| H | |
| J | |
| K | ICS |
| V | ICS |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | RDPC |
| U | RDPC |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | WTPC |
| d | WTPC |
| e | |
| h | |
| f | |
| j | |
| k | |
| l | |
| m | ACT |
| n | ACT |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | DRDY |
| NN | DRDY |

*Figure 6-6 Cont'd (Sheet 4 of 18)*

PARALLEL OUTPUT
TO NEXT DEVICE

J09

| PIN | SIGNAL |
|---|---|
| A | PM00 |
| B | PM00 |
| C | PM01 |
| D | PM01 |
| E | PM02 |
| F | PM02 |
| H | PM03 |
| J | PM03 |
| K | PM04 |
| V | PM04 |
| L | PM05 |
| M | PM05 |
| N | PM06 |
| P | PM06 |
| R | PM07 |
| S | PM07 |
| T | PM08 |
| U | PM08 |
| W | PM09 |
| X | PM09 |
| Y | PM10 |
| Z | PM10 |
| a | PM11 |
| b | PM11 |
| c | PM12 |
| d | PM12 |
| e | PM13 |
| h | PM13 |

| PIN | SIGNAL |
|---|---|
| f | PM14 |
| j | PM14 |
| k | PM15 |
| l | PM15 |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 5 of 18)*

## PARALLEL OUTPUT TO CPU

J10

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PIE00 | f | PIE14 |
| B | PIE00 | j | PIE14 |
| C | PIE01 | k | PIE15 |
| D | PIE01 | l | PIE15 |
| E | PIE02 | m | |
| F | PIE02 | n | |
| H | PIE03 | p | |
| J | PIE03 | r | |
| K | PIE04 | s | |
| V | PIE04 | t | |
| L | PIE05 | u | |
| M | PIE05 | v | |
| N | PIE06 | w | |
| P | PIE06 | x | |
| R | PIE07 | y | |
| S | PIE07 | JJ | |
| T | PIE08 | z | |
| U | PIE08 | AA | |
| W | PIE09 | BB | |
| X | PIE09 | CC | |
| Y | PIE10 | DD | |
| Z | PIE10 | EE | |
| a | PIE11 | FF | |
| b | PIE11 | HH | |
| c | PIE12 | KK | |
| d | PIE12 | LL | |
| e | PIE13 | MM | |
| h | PIE13 | NN | |

*Figure 6-6 Cont'd (Sheet 6 of 18)*

DEVICE 1
INPUT/OUTPUT CONTROL

J23

| PIN | SIGNAL |
|---|---|
| A | PDSA |
| B | PDSA |
| C | |
| D | |
| E | |
| F | |
| H | |
| J | |
| K | ICSA |
| V | ICSA |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | RDPCA |
| U | RDPCA |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | WTPCA |
| d | WTPCA |
| e | |
| h | |

| PIN | SIGNAL |
|---|---|
| f | |
| j | |
| k | |
| l | |
| m | ACTA |
| n | ACTA |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | DRDYA |
| NN | DRDYA |

*Figure 6-6 Cont'd (Sheet 7 of 18)*

## DEVICE 1
## PARALLEL OUTPUT

J24

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PM00A | f | PM14A |
| B | PM00A | j | PM14A |
| C | PM01A | k | PM15A |
| D | PM01A | l | PM15A |
| E | PM02A | m | |
| F | PM02A | n | |
| H | PM03A | p | |
| J | PM03A | r | |
| K | PM04A | s | |
| V | PM04A | t | |
| L | PM05A | u | |
| M | PM05A | v | |
| N | PM06A | w | |
| P | PM06A | x | |
| R | PM07A | y | |
| S | PM07A | JJ | |
| T | PM08A | z | |
| U | PM08A | AA | |
| W | PM09A | BB | |
| X | PM09A | CC | |
| Y | PM10A | DD | |
| Z | PM10A | EE | |
| a | PM11A | FF | |
| b | PM11A | HH | |
| c | PM12A | KK | |
| d | PM12A | LL | |
| e | PM13A | MM | |
| h | PM13A | NN | |

*Figure 6-6 Cont'd (Sheet 8 of 18)*

DEVICE 1
PARALLEL INPUT

J25

| PIN | SIGNAL |
|---|---|
| A | PIE00A |
| B | PIE00A |
| C | PIE01A |
| D | PIE01A |
| E | PIE02A |
| F | PIE02A |
| H | PIE03A |
| J | PIE03A |
| K | PIE04A |
| V | PIE04A |
| L | PIE05A |
| M | PIE05A |
| N | PIE06A |
| P | PIE06A |
| R | PIE07A |
| S | PIE07A |
| T | PIE08A |
| U | PIE08A |
| W | PIE09A |
| X | PIE09A |
| Y | PIE10A |
| Z | PIE10A |
| a | PIE11A |
| b | PIE11A |
| c | PIE12A |
| d | PIE12A |
| e | PIE13A |
| h | PIE13A |

| PIN | SIGNAL |
|---|---|
| f | PIE14A |
| j | PIE14A |
| k | PIE15A |
| l | PIE15A |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 9 of 18)*

## DEVICE 2
## INPUT/OUTPUT CONTROL

J26

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PDSB | f | |
| B | PDSB | j | |
| C | | k | |
| D | | l | |
| E | | m | ACTB |
| F | | n | ACTB |
| H | | p | |
| J | | r | |
| K | ICSB | s | |
| V | ICSB | t | |
| L | | u | |
| M | | v | |
| N | | w | |
| P | | x | |
| R | | y | |
| S | | JJ | |
| T | RDPCB | z | |
| U | RDPCB | AA | |
| W | | BB | |
| X | | CC | |
| Y | | DD | |
| Z | | EE | |
| a | | FF | |
| b | | HH | |
| c | WTPCB | KK | |
| d | WTPCB | LL | |
| e | | MM | DRDYB |
| h | | NN | DRDYB |

*Figure 6-6 Cont'd (Sheet 10 of 18)*

## DEVICE 2
## PARALLEL OUTPUT

J27

| PIN | SIGNAL |
|---|---|
| A | PM00B |
| B | PM00B |
| C | PM01B |
| D | PM01B |
| E | PM02B |
| F | PM02B |
| H | PM03B |
| J | PM03B |
| K | PM04B |
| V | PM04B |
| L | PM05B |
| M | PM05B |
| N | PM06B |
| P | PM06B |
| R | PM07B |
| S | PM07B |
| T | PM08B |
| U | PM08B |
| W | PM09B |
| X | PM09B |
| Y | PM10B |
| Z | PM10B |
| a | PM11B |
| b | PM11B |
| c | PM12B |
| d | PM12B |
| e | PM13B |
| h | PM13B |
| f | PM14B |
| j | PM14B |
| k | PM15B |
| l | PM15B |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 11 of 18)*

## DEVICE 2
## PARALLEL INPUT

J28

| PIN | SIGNAL |
|---|---|
| A | PIE00B |
| B | PIE00B |
| C | PIE01B |
| D | PIE01B |
| E | PIE02B |
| F | PIE02B |
| H | PIE03B |
| J | PIE03B |
| K | PIE04B |
| V | PIE04B |
| L | PIE05B |
| M | PIE05B |
| N | PIE06B |
| P | PIE06B |
| R | PIE07B |
| S | PIE07B |
| T | PIE08B |
| U | PIE08B |
| W | PIE09B |
| X | PIE09B |
| Y | PIE10B |
| Z | PIE10B |
| a | PIE11B |
| b | PIE11B |
| c | PIE12B |
| d | PIE12B |
| e | PIE13B |
| h | PIE13B |

| PIN | SIGNAL |
|---|---|
| f | PIE14B |
| j | PIE14B |
| k | PIE15B |
| l | PIE15B |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 12 of 18)*

DEVICE 3
INPUT/OUTPUT CONTROL

J29

| PIN | SIGNAL |
|---|---|
| A | PDSC |
| B | PDSC |
| C | |
| D | |
| E | |
| F | |
| H | |
| J | |
| K | ICSC |
| V | ICSC |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | RDPCC |
| U | RDPCC |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | WTPCC |
| d | WTPCC |
| e | |
| h | |

| PIN | SIGNAL |
|---|---|
| f | |
| j | |
| k | |
| l | |
| m | ACTC |
| n | ACTC |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | DRDYC |
| NN | DRDYC |

*Figure 6-6 Cont'd (Sheet 13 of 18)*

## DEVICE 3
## PARALLEL OUTPUT

J30

| PIN | SIGNAL |
|---|---|
| A | PM00C |
| B | PM00C |
| C | PM01C |
| D | PM01C |
| E | PM02C |
| F | PM02C |
| H | PM03C |
| J | PM03C |
| K | PM04C |
| V | PM04C |
| L | PM05C |
| M | PM05C |
| N | PM06C |
| P | PM06C |
| R | PM07C |
| S | PM07C |
| T | PM08C |
| U | PM08C |
| W | PM09C |
| X | PM09C |
| Y | PM10C |
| Z | PM10C |
| a | PM11C |
| b | PM11C |
| c | PM12C |
| d | PM12C |
| e | PM13C |
| h | PM13C |

| PIN | SIGNAL |
|---|---|
| f | PM14C |
| j | PM14C |
| k | PM15C |
| l | PM15C |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 14 of 18)*

DEVICE 3
PARALLEL INPUT

J31

| PIN | SIGNAL |
|---|---|
| A | PIE00C |
| B | PIE00C |
| C | PIE01C |
| D | PIE01C |
| E | PIE02C |
| F | PIE02C |
| H | PIE03C |
| J | PIE03C |
| K | PIE04C |
| V | PIE04C |
| L | PIE05C |
| M | PIE05C |
| N | PIE06C |
| P | PIE06C |
| R | PIE07C |
| S | PIE07C |
| T | PIE08C |
| U | PIE08C |
| W | PIE09C |
| X | PIE09C |
| Y | PIE10C |
| Z | PIE10C |
| a | PIE11C |
| b | PIE11C |
| c | PIE12C |
| d | PIE12C |
| e | PIE13C |
| h | PIE13C |

| PIN | SIGNAL |
|---|---|
| f | PIE14C |
| j | PIE14C |
| k | PIE15C |
| l | PIE15C |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 15 of 18)*

DEVICE 4
INPUT/OUTPUT CONTROL

J32

| PIN | SIGNAL | PIN | SIGNAL |
|---|---|---|---|
| A | PDSD | f | |
| B | PDSD | j | |
| C | | k | |
| D | | l | |
| E | | m | ACTD |
| F | | n | ACTD |
| H | | p | |
| J | | r | |
| K | ICSD | s | |
| V | ICSD | t | |
| L | | u | |
| M | | v | |
| N | | w | |
| P | | x | |
| R | | y | |
| S | | JJ | |
| T | RDPCD | z | |
| U | RDPCD | AA | |
| W | | BB | |
| X | | CC | |
| Y | | DD | |
| Z | | EE | |
| a | | FF | |
| b | | HH | |
| c | WTPCD | KK | |
| d | WTPCD | LL | |
| e | | MM | DRDYD |
| h | | NN | DRDYD |

*Figure 6-6 Cont'd (Sheet 16 of 18)*

DEVICE 4
PARALLEL OUTPUT

J33

| PIN | SIGNAL |
|---|---|
| A | PM00D |
| B | PM00D |
| C | PM01D |
| D | PM01D |
| E | PM02D |
| F | PM02D |
| H | PM03D |
| J | PM03D |
| K | PM04D |
| V | PM04D |
| L | PM05D |
| M | PM05D |
| N | PM06D |
| P | PM06D |
| R | PM07D |
| S | PM07D |
| T | PM08D |
| U | PM08D |
| W | PM09D |
| X | PM09D |
| Y | PM10D |
| Z | PM10D |
| a | PM11D |
| b | PM11D |
| c | PM12D |
| d | PM12D |
| e | PM13D |
| h | PM13D |

| PIN | SIGNAL |
|---|---|
| f | PM14D |
| j | PM14D |
| k | PM15D |
| l | PM15D |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 17 of 18)*

DEVICE 4
PARALLEL INPUT

J34

| PIN | SIGNAL |
|---|---|
| A | PIE00D |
| B | PIE00D |
| C | PIE01D |
| D | PIE01D |
| E | PIE02D |
| F | PIE02D |
| H | PIE03D |
| J | PIE03D |
| K | PIE04D |
| V | PIE04D |
| L | PIE05D |
| M | PIE05D |
| N | PIE06D |
| P | PIE06D |
| R | PIE07D |
| S | PIE07D |
| T | PIE08D |
| U | PIE08D |
| W | PIE09D |
| X | PIE09D |
| Y | PIE10D |
| Z | PIE10D |
| a | PIE11D |
| b | PIE11D |
| c | PIE12D |
| d | PIE12D |
| e | PIE13D |
| h | PIE13D |

| PIN | SIGNAL |
|---|---|
| f | PIE14D |
| j | PIE14D |
| k | PIE15D |
| l | PIE15D |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 6-6 Cont'd (Sheet 18 of 18)*

# SECTION VII
# INTERRUPT SYSTEM

## BASIC DESCRIPTION

The SCC 4700 interrupt capability allows the normal execution of a program to be interrupted in order to process a program of higher priority. Situations which may interrupt program execution are referred to as interrupt conditions.

Upon receiving notification of an interrupt condition, the SCC 4700 interrupt system instructs the central processor to process a program of high priority. The central processor stops execution of the current program and honors the request for an interrupt at the first available time. When the interrupt request is honored, the interrupt system supplies a memory location to the central processor. The central processor then generates in hardware and executes a System Call (SYCL) on the reserved location assigned for the interrupt (unless the interrupt is for power on). Contained in the reserved location is the effective address of the interrupt service subroutine. Hence, execution of SYCL on the interrupt location transfers control to the interrupt service routine. (This subroutine can be interrupted by the occurrence of an interrupt of higher priority, but not a lower until the current interrupt is cleared.)

On completion of the interrupt service subroutine (which resets the source of the interrupt), Clear Interrupt (CLI) and System Return (SRT) instructions are executed so that lower priority interrupts may be serviced. The interrupt service subroutine then returns control to the interrupted program and execution of the interrupted program continues. An example of this concept is illustrated in Figure 7-1.

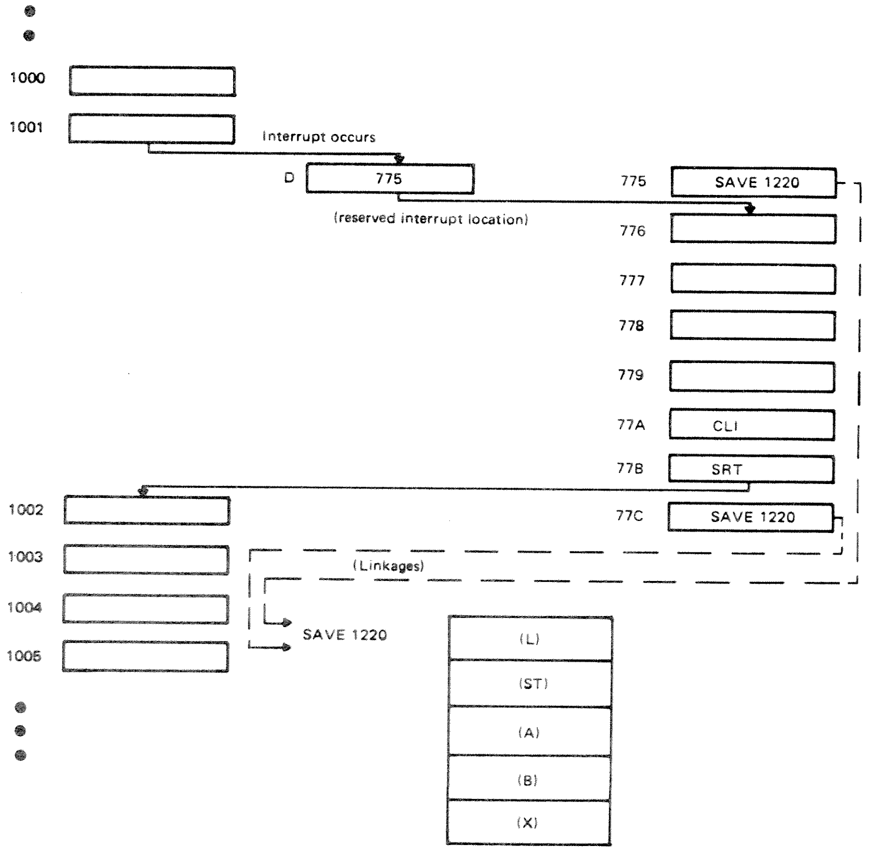


*Figure 7-1. Interrupt Sequence*

## THEORY OF OPERATION

### Enabling the Interrupt System

The interrupt system, except power up, power down, and system trap interrupts, is activated by the Enable Interrupts (ENA) instruction and deactivated by the Disable Interrupts (DIS) instruction. The ENA and DIS instructions affect all of the following interrupts as a group:

Interrupts for Channels 1 through 4

External Interrupts

Real Time Clock

Console Interrupt

Power up, power down, and system trap are enabled (i.e., ready to be activated) at all times and cannot be disarmed.

### Arming Interrupts

Arming/disarming of the various interrupts in the basic SCC 4700 system is as follows:

External interrupts are always armed (selective arming/disarming is available as an option).

Individual device interrupts are armed/disarmed by an SIO command.

Channel interrupts, console interrupt, and real time clock interrupt are armed/disarmed by the ARM instruction.

A Master Clear or power on disarms and disables all interrupts.

### Servicing Interrupts

An interrupt request from the interrupt system to the central processor is honored following completion of the instruction under execution unless the instruction is a non-interruptable instruction (ARM, ENA, DIS, CLI, SYCL, or ACT). If one of these instructions is being executed when the interrupt request occurs, the interrupt request is not granted until completion of an instruction which allows interrupts.

Interrupts and traps which are honored immediately (i.e., the instruction under execution is not completed) consist of the following:

Power On

$\left.\begin{array}{l}\text{Unimplemented Instruction}\\ \text{Protection Key Violation}\\ \text{Privileged Instruction Violation}\end{array}\right\}$ System Traps

Memory Parity

The program interrupts contained in channels 1 through 4 and the external interrupt subsystem follow a two-state sequence when an interrupt condition occurs. If an interrupt is armed when an interrupt condition occurs, it immediately enters the waiting state. The waiting state is maintained until no higher priority interrupt is waiting or being serviced. An interrupt request and an identifying address to the CPU is then generated. When the CPU honors this interrupt request, the above interrupt switches from the waiting state to the active state. The interrupt remains in the active state until a CLI (Clear Interrupt) instruction, which clears the highest active state, is executed.

An interrupt in the active state may be interrupted by a higher priority interrupt. If several interrupts enter the waiting state, the CPU will honor each interrupt request individually according to priority.

Interrupt requests in the waiting state are honored by the CPU only when the interrupts have been group enabled by the ENA instruction.

## PRIORITY STRUCTURE

The promptness with which an interrupt request is serviced is dependent on its priority; i.e., the highest priority interrupt is always serviced first, followed by the next highest, etc. After an interrupt has been serviced and cleared by the CLI instruction, the interrupt system prepares to receive the next highest priority interrupt waiting for service. Any lower priority interrupts which occur are remembered and serviced after each of the higher priority interrupts have been cleared.

The priority structure of the SCC 4700 interrupt system performs the following functions:

1. Resolves contention problems arising from simultaneous occurrence of interrupt conditions.
2. Permits an interrupt program to be interrupted by a request to service an interrupt of higher priority.
3. Permits lower priority interrupts which occur to be remembered and serviced at a later time.

Table 7-1 provides the priority level, core assignment, and timing of the various traps and interrupts which the SCC 4700 may service.

Figure 7-2 illustrates the Interrupt System Flow.

*Table 1. SCC 4700 Traps and Interrupts*

| RESERVED CORE ASSIGNMENTS | | | PRIORITY LEVELS | | TIMING | |
|---|---|---|---|---|---|---|
| DECIMALS | HEXADECIMAL | DESCRIPTION | SEQUENCE | REMARKS | μsec | MULTIPLE FUNCTIONS |
| 0 | 0 | Power Up Interrupt * | 1 | | 5.62 | |
| 1 | 1 | Power Down Interrupt * | 5 | | 7.36 | |
| 2 | 2 | Real Time Clock (location being decremented) | 27 | | 2.59 | Decrement Location** |
| 3 | 3 | Real Time Clock Trap Location | | | 8.73 | Decrement, Test and Trap |
| 4 | 4 | Console Interrupt | 28 | | 7.36 | |
| 5 | 5 | System Trap Indicator | 4 | | 7.76 | Unimplemented Instruction ①<br>Floating Point Over/Underflow * |
| 6 | 6 | System Trap Location | 2 | System Protection | 8.26 | Protection Key Violation *<br>Privileged Instruction *<br>Memory Parity * ② |
| 7 | 7 | Unassigned | ▨ | ▨ | | |
| 8 | 8 | Channel 1 Interrupt | 7-26 | Service Request (lowest location has highest priority) | 7.36 | Service Request (I/O or external interrupts) |
| 9 | 9 | Channel 2 Interrupt * | | | | |
| 10 | A | Channel 3 Interrupt * | | | | |
| 11 | B | Channel 4 Interrupt * | | | | |
| 12–27 | C–1B | External Interrupts * | | | | |
| 28–283 | 1C–11B | DCDW Pointers (64/channel) | 6 | Transfer** Request | 7.94 ***<br>10.33<br>12.47 | Normal Transfer (block mode)<br>Transfer with data chaining<br>Transfer with chaining and new pointer. |
| 284–347 | 11C–15B | System Calls (64 monitor entry points) | ▨ | ▨ | 7.64 | Storage of register plus transfer |

* Optional.

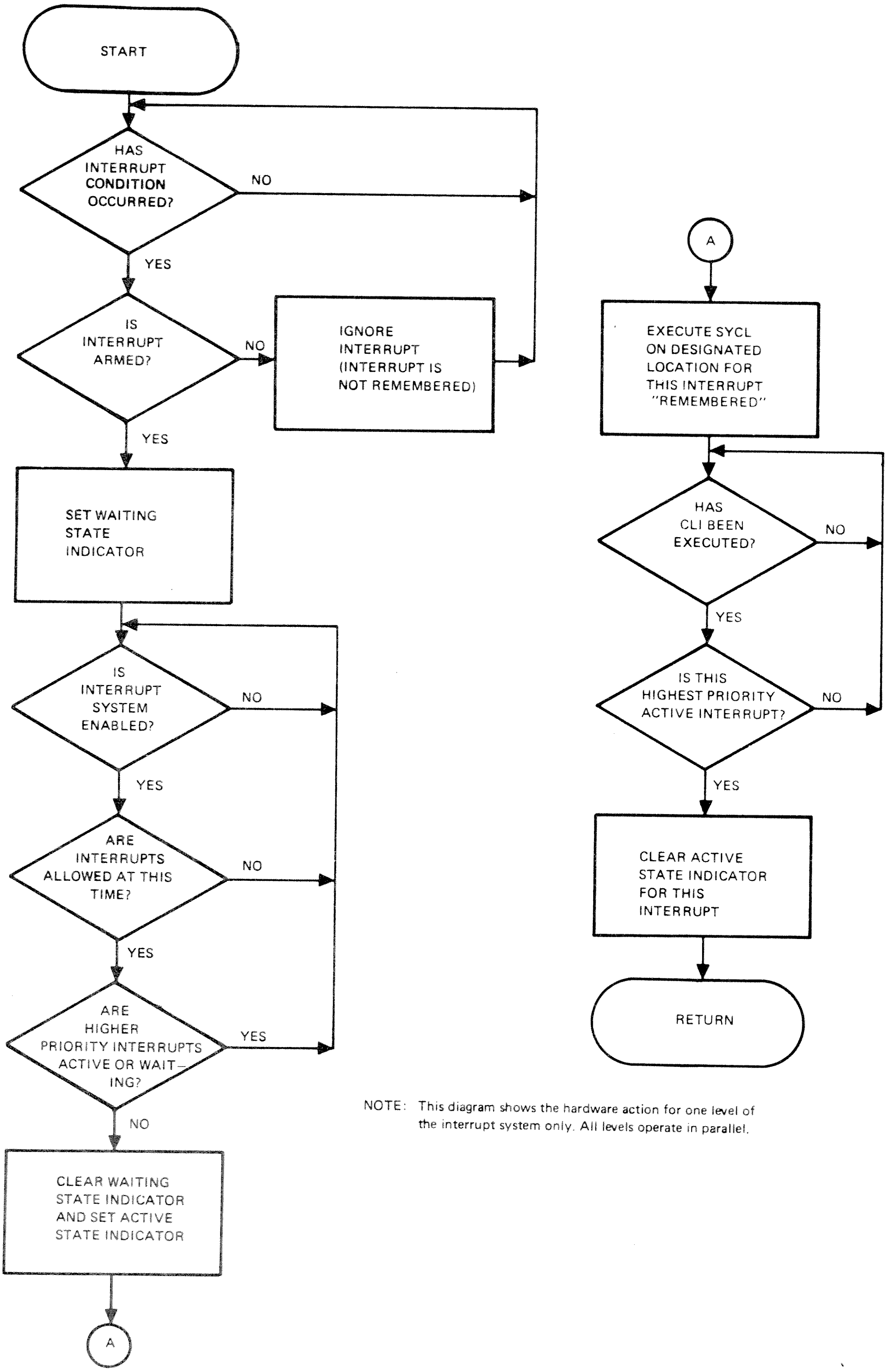
** Microinterrupts which do not affect the program.

***Minimum value, actual value depends upon device response time.

NOTES:

① In basic machine, only the unimplemented instruction function of the system trap is operative.

② Another microinterrupt is implemented with memory mapping. Called "Map No-match" it is 3rd in priority sequence (inhibits Protection Key Violation). **


START
HAS INTERRUPT CONDITION OCCURRED?
NO
YES
IS INTERRUPT ARMED?
NO
IGNORE INTERRUPT (INTERRUPT IS NOT REMEMBERED)
YES
SET WAITING STATE INDICATOR
IS INTERRUPT SYSTEM ENABLED?
NO
YES
ARE INTERRUPTS ALLOWED AT THIS TIME?
NO
YES
ARE HIGHER PRIORITY INTERRUPTS ACTIVE OR WAIT–ING?
YES
NO
CLEAR WAITING STATE INDICATOR AND SET ACTIVE STATE INDICATOR
A
A
EXECUTE SYCL ON DESIGNATED LOCATION FOR THIS INTERRUPT "REMEMBERED"
HAS CLI BEEN EXECUTED?
NO
YES
IS THIS HIGHEST PRIORITY ACTIVE INTERRUPT?
NO
YES
CLEAR ACTIVE STATE INDICATOR FOR THIS INTERRUPT
RETURN

NOTE: This diagram shows the hardware action for one level of the interrupt system only. All levels operate in parallel.

*Figure 7-2. Interrupt System Flow Diagram*

## CHANNEL INTERRUPTS

One interrupt level is provided for each of the four channels (multiplexor and selector). There is no individual enabling of the channel interrupts; however, they may be group enabled or disabled by the ENA or DIS instruction. Individual arming or disarming of the channel interrupts is accomplished through the use of the ARM instruction.

If an interrupt occurs on a channel, the program must identify the interrupting device by executing an IIU command. Each device is individually armed or disarmed with an SIO command. A system clear disarms device and channel interrupts.

## EXTERNAL INTERRUPT SUBSYSTEM

The basic external interrupt subsystem (Figure 7-3) consists of the Priority Interrupt Control Unit which provides an interface to the CPU and the 1 to 16 external priority interrupts. Each interrupt provides one priority interrupt level. Individual arming/disarming, enabling/disabling, and triggering of the external interrupts is available as an option.

The optional Extended Control Unit interfaces with the parallel I/O and adds the capability to individually arm/disarm, enable/disable, and trigger each of the 16 external priority interrupts. Through the use of an Activate (ACT) instruction followed by a Write Parallel (WTP) instruction, the desired interrupt function may be specified. The format for ACT and WTP is as follows.

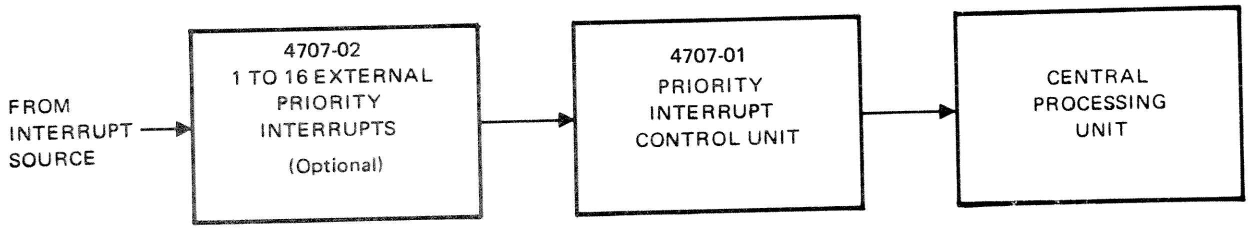
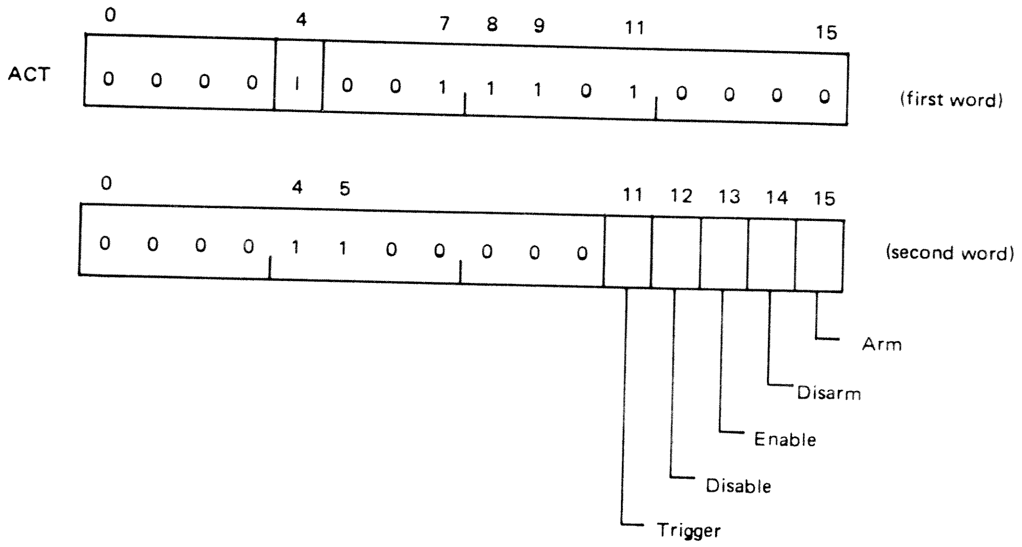
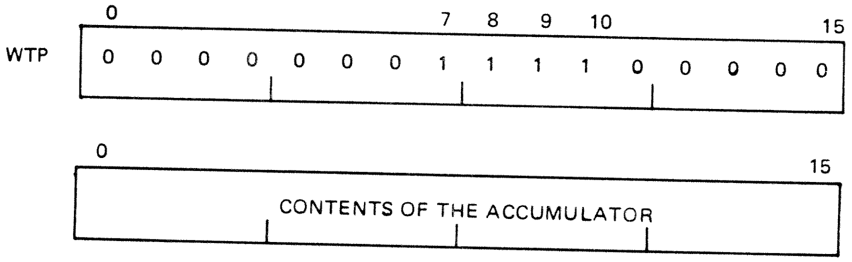


*Figure 7-3. External Interrupt Subsystem*



By setting (=1) the proper bit of the second word of ACT, the desired interrupt functions (to a maximum of three) may be specified.

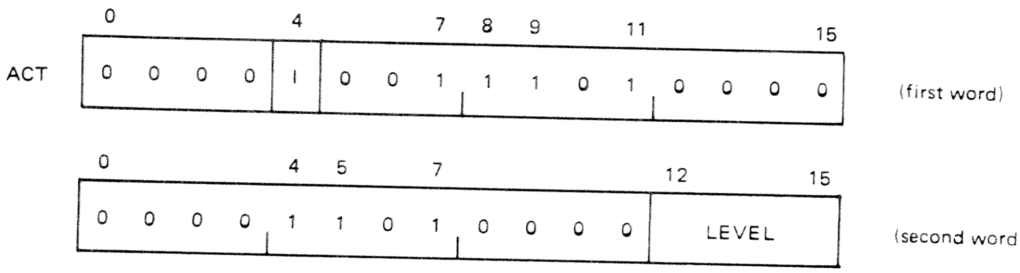


Each of the 16 bits transferred from the accumulator by the WTP instruction specifies a level to be armed, disarmed, enabled, disabled, or triggered.

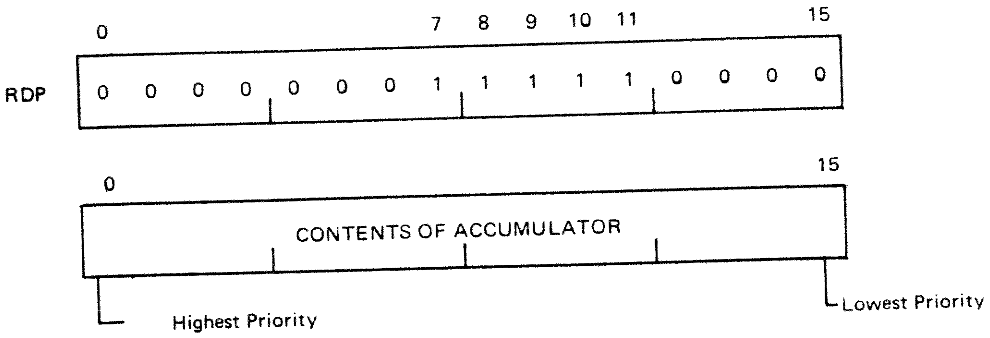
1 = arms, disarms, enables, disables, or triggers (as indicated by ACT).

0 = no action.

The Expander Unit is an optional unit which also interfaces with the parallel I/O. The addition of this unit adds the capability of each interrupt level to service up to 16 devices rather than one. When an interrupt occurs, the programmer identifies the device causing the interrupt by using an ACT and RDP (Read Parallel I/O) pair of instructions. The format for ACT and RDP is as follows.



Bits 12 through 15 of the second word of ACT designate (in binary) the interrupt level to be read.



The indicated interrupt is specified by setting the proper bit of the accumulator.

The LLO (Locate Leading One) instruction may be used to determine in priority order which interrupts on a given level are active.

## REAL TIME CLOCK

The Real Time Clock (RTC) decrements location 2 at the local power line frequency if the interrupt system is enabled and the RTC is armed. When location 2 reaches zero, a trap will be made to location 3.

The Real Time Clock may be disarmed or disabled for one-half cycle of the power line frequency without losing clock counts. Disarming the RTC will inhibit the update of displays on the control console.

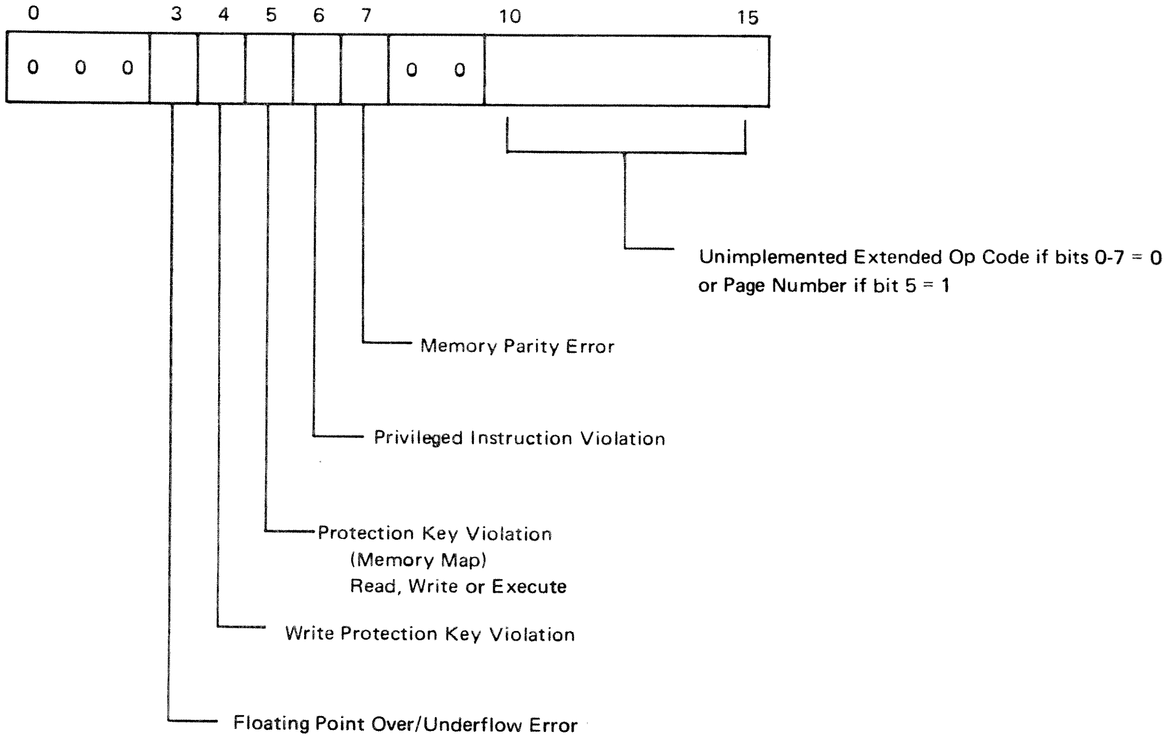
## POWER ON INTERRUPT

The Power On interrupt operates differently from other interrupts so as to restore the registers. Instead of simulating the System Call, it generates in hardware and executes System Return (SRT). Interrupt location zero corresponds to the second word of an SRT instruction and should contain the address of the registers.

Without the Power Fail Safe option the machine will halt following the interrupt.

## SYSTEM TRAP INDICATOR

Location 5 in core memory is reserved for the system trap indicator. If a system trap occurs, a system call will be executed on location 6. At that time the system trap indicator will be set to show what condition caused the trap.


0
3
4
5
6
7
10
15
0 0 0
0 0
Unimplemented Extended Op Code if bits 0-7 = 0
or Page Number if bit 5 = 1
Memory Parity Error
Privileged Instruction Violation
Protection Key Violation
(Memory Map)
Read, Write or Execute
Write Protection Key Violation
Floating Point Over/Underflow Error

If bits 3, 4, 5, 6, or 7 are set (=1), it will indicate the condition shown on the diagram. If more than one error condition occurs simultaneously, all of the corresponding bits in the trap indicator will be set.

If the system trap occurs and bits 3, 4, 5, 6, and 7 are all equal to zero, this will indicate that an attempt was made to execute an unimplemented instruction.

Since all 16 possible combinations are utilized in the standard machine for basic operation codes, any unimplemented instruction must be of the extended non-privileged op code set.

The extended non-privileged operation code is contained in bits 7 through 12 of the instruction. When the trap occurs, bits 0 through 7 of the system trap indicator are reset to zero and bits 7 through 12 of the instruction which caused the trap are stored in bits 10 through 15 of the indicator word.

In the basic machine (without memory parity, memory mapping unit, or optional instruction sets) only the unimplemented instruction trap is operative.

The location counter will always contain the address of the instruction causing the system trap, except for floating point over/underflow. In this case, the location counter will contain the address of the next instruction. If a protection key violation occurs (bit 5 = 1), bits 10 through 15 will contain the page number causing the violation.

Memory parity errors may cause loss of data in the programmable registers when the error occurs. Floating point under/overflow causes the contents of the A, B, and E registers to be undefined. Protection key violation, privileged instruction violation, or unimplemented instruction system traps cause loss of data.

## EXTERNAL INTERRUPTS – IMPLEMENTATION

Interrupts external to the CPU include the channel interrupts and the sixteen external interrupt capability provided by the External Interrupt Subsystem. The channel interrupts consist of the microinterrupt and the program interrupt. These channel interrupts are described in detail in the M channel description given in Section III. The External Interrupt Subsystem basically consists of the Model 4707-02 External Interrupt Unit and the Model 4707-01 Control Unit. A block diagram of the signal interface of the External Interrupt Subsystem is shown in Figure 7-4. The Model 4707-02 Interrupt Unit provides 16 interrupt levels, each interrupt level is on a different card. Input requests (IREQ) are received at the Model 4707-02 cards, one request per card. The card with the highest priority will generate an interrupt request (INTR) and an acknowledge (IACK) signal. The INTR signal is used in the Model 4707-01 Control Unit to generate an address (IAX), a priority signal (HPIIS) and the interrupt request INTR 1. The IACK signal from the card is sent to the device requesting the interrupt. An additional interrupt request input, provided by the interrupt subsystem, may be used from a second group of interrupts to generate the INTR 2 signal.

## THEORY OF OPERATION

Control signals from the CPU are passed to the Model 4707-02 Unit through the Model 4707-01 Control Unit. The INTR (1 through 16) signals from the Model 4707-02 Unit are used in the Model 4707-01 Unit to generate a unique 4-bit address IA (2 through 5) for each one of the 16 INTR signals. A priority signal HP115 is also generated by the Model 4707-01 Control Unit from the INTR signals.

The Model 4707-02 Interrupt Unit interface logic generates an INTR signal and an IACK signal upon receiving an interrupt request (IREQ) from an external location, provided the required gating signals are present. Refer to the timing diagram in Figure 7-5. The IREQ signal sets the WAIT flip-flop if the flip-flop is armed (ARM high or no input), INTCL is false, and the ACTIVE flip-flop is not set. When the WAIT flip-flop is set an acknowledge IACK signal is sent to the external location and if no higher priority interrupt (HIPI) is active or waiting, an INTR signal is generated. The WAIT flip-flop also enables the set input gate to the PREACTIVE flip-flop and generates the interrupt busy (IBO) signal if no higher priority interrupt is waiting. The IBO signal is sent to lower priority interrupt levels. If GENB and IENB enable signals are high (or no inputs), INTCL is true high and no higher priority interrupt is waiting, the PREACTIVE flip-flop is set by the WAIT flip-flop. As the PREACTIVE flip-flop is set, the input gate to the PREACTIVE flip-flop is enabled. When the CPU logic drops the INTCL signal to the false state, the ACTIVE flip-flop is set.

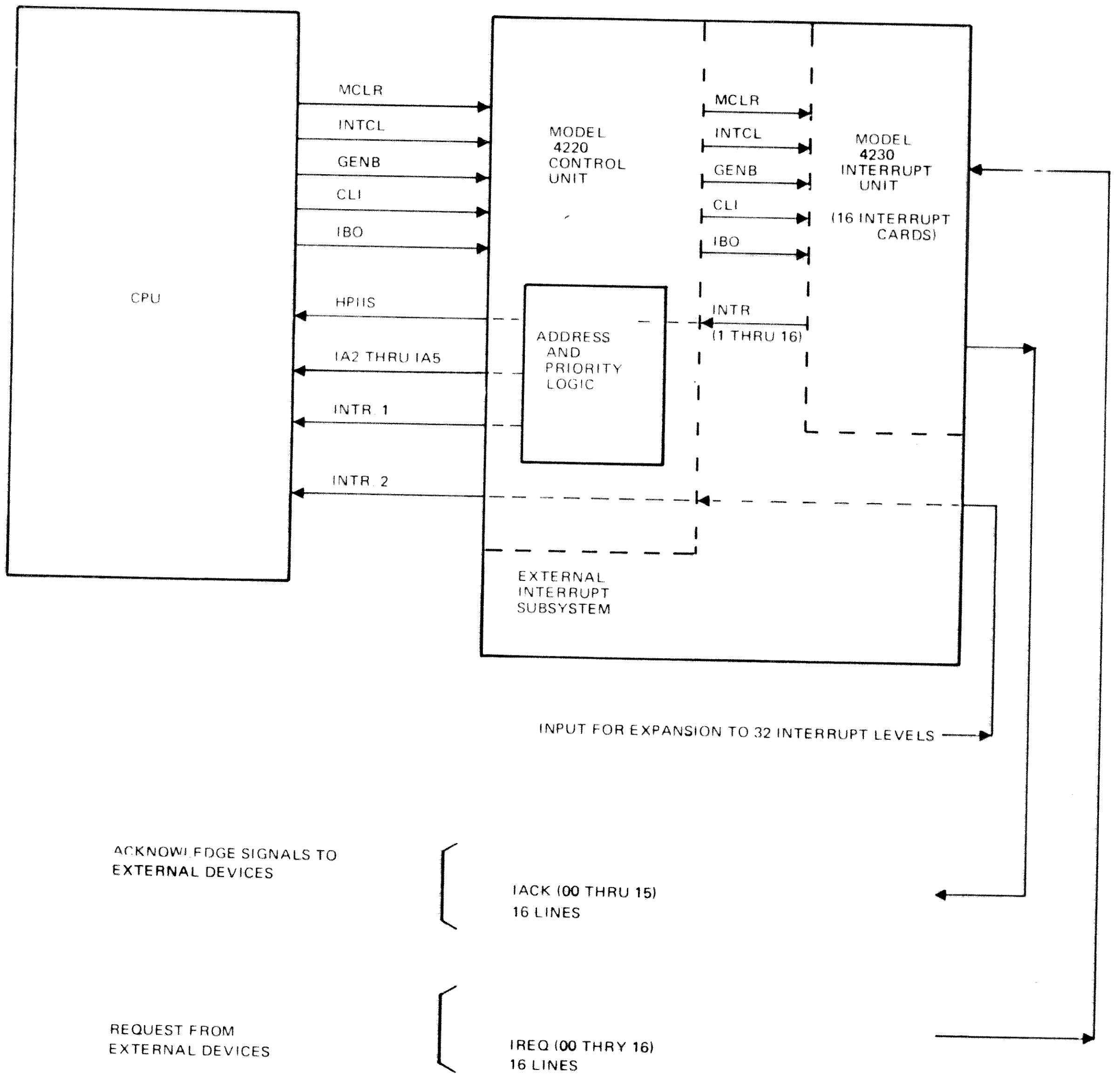


*Figure 7-4. External Interrupt System Signal Interface*

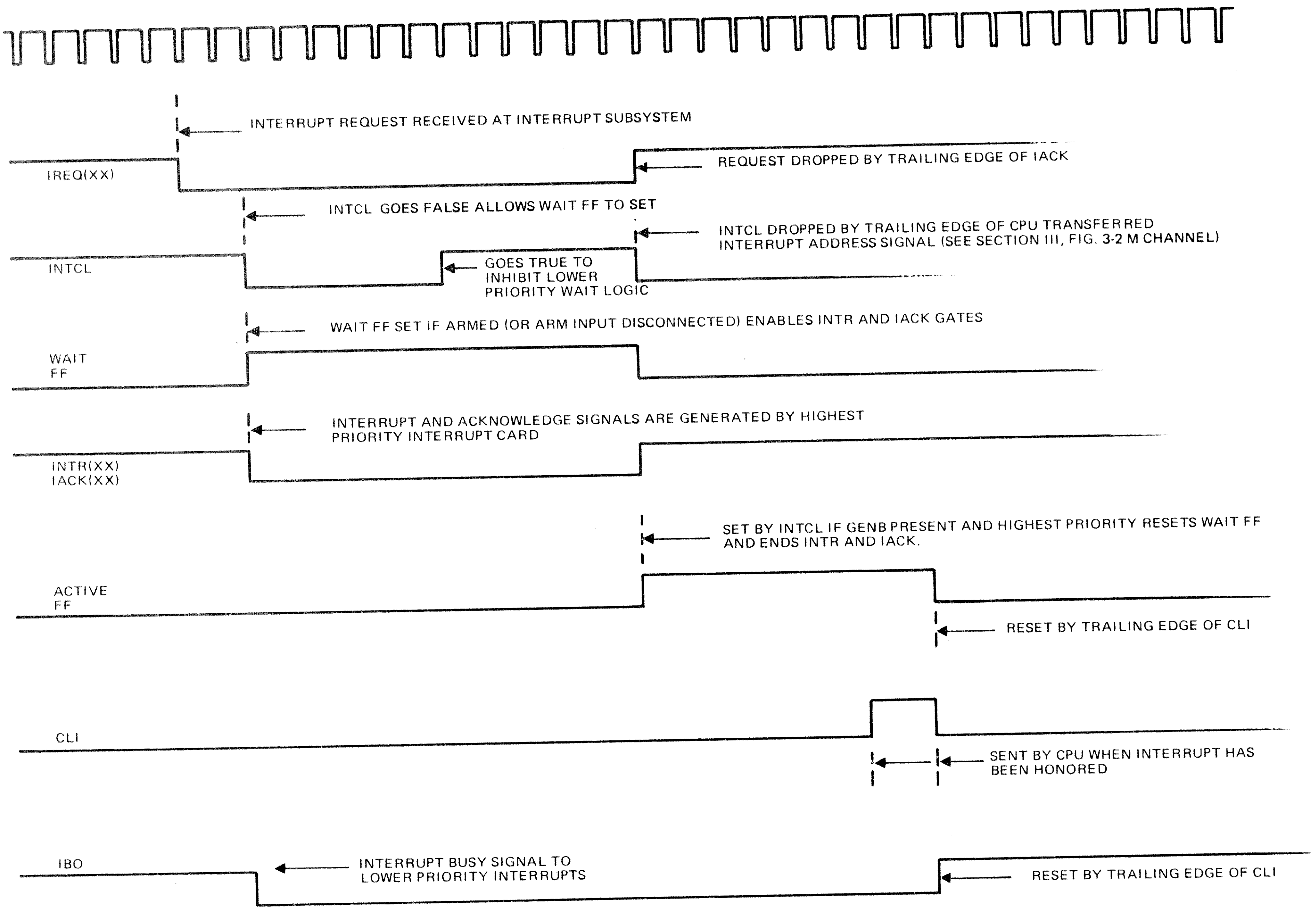

INTERRUPT REQUEST RECEIVED AT INTERRUPT SUBSYSTEM
IREQ(XX)
REQUEST DROPPED BY TRAILING EDGE OF IACK
INTCL GOES FALSE ALLOWS WAIT FF TO SET
INTCL DROPPED BY TRAILING EDGE OF CPU TRANSFERRED INTERRUPT ADDRESS SIGNAL (SEE SECTION III, FIG. 3-2 M CHANNEL)
INTCL
GOES TRUE TO INHIBIT LOWER PRIORITY WAIT LOGIC
WAIT FF SET IF ARMED (OR ARM INPUT DISCONNECTED) ENABLES INTR AND IACK GATES
WAIT FF
INTERRUPT AND ACKNOWLEDGE SIGNALS ARE GENERATED BY HIGHEST PRIORITY INTERRUPT CARD
INTR(XX) IACK(XX)
SET BY INTCL IF GENB PRESENT AND HIGHEST PRIORITY RESETS WAIT FF AND ENDS INTR AND IACK.
ACTIVE FF
RESET BY TRAILING EDGE OF CLI
CLI
SENT BY CPU WHEN INTERRUPT HAS BEEN HONORED
IBO
INTERRUPT BUSY SIGNAL TO LOWER PRIORITY INTERRUPTS
RESET BY TRAILING EDGE OF CLI

*Figure 7-5. External Interrupt Subsystem Timing*

The $\overline{Q}$ output of the ACTIVE flip-flop resets the WAIT and IBO logic. When the interrupt has been honored the CPU sends the CLI signal to clear the highest priority ACTIVE flip-flop. IENB and ARM inputs may be used to selectively arm or disarm interrupt levels. The 16 interrupt levels may be disabled as a group by the GENB signal.

## EXTERNAL INTERRUPT SUBSYSTEM

### SIGNALS

Input

1. MCLR – Master Clear: A high true signal from the CPU. The signal is used to clear all interrupt logic, waiting or active.
2. INTCL – Interrupt Control: A low true signal from the CPU, the trailing edge of which gates the highest priority waiting logic to the active state.
3. GENB – Group Enable: A low true signal used to enable all active logic.
4. CLI – Clear Interrupt: A low true signal used to clear the highest priority active interrupt.
5. IBO – Interrupt Busy: A high true signal used to indicate that an M or S channel is active and waiting.
6. IREQ (0-15) – Sixteen Interrupt Request Lines: Low true signals from the external devices requesting interrupt service. The request is dropped by the trailing edge of the signal.

Output

1. HPIIS – Higher Priority Internal Interrupt Signal: A high true signal indicating an interrupt is waiting or active.
2. IA2-IA5 – Four Interrupt Address Lines: Four lines, IA2 through IA5, of interrupt address for a total of 16 addresses.
3. INTR 1 – Interrupt 1: A high true interrupt request signal to the CPU when an interrupt is in the waiting state and no higher interrupt is active.
4. INTR 2 – Interrupt 2: Same as INTR 1, except generated by a second group of interrupts. It provides for expansion to 32 interrupt levels.
5. IACK (0-15) – Interrupt Acknowledge: Sixteen low true interrupt acknowledge signals sent to the external interrupting device.

### CONNECTORS

The External Interrupt Subsystem connectors, with the associated pin numbers and signals, are listed in Figure 7-6. Signatures ending with the suffix LO indicate the ground side of the signal. The connectors are ELCO 8016-0566-000-007 type of connectors.

J11

| PIN | SIGNAL |
|---|---|
| A | ARM 1 |
| B | ARM1LO |
| C | ARM2 |
| D | ARM2LO |
| E | ARM3 |
| F | ARM3LO |
| H | |
| J | |
| K | |
| V | |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | |
| U | |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | |
| d | |
| e | |
| h | |
| f | |
| j | |
| k | ARM16 |
| l | ARM16LO |

J12

| PIN | SIGNAL |
|---|---|
| A | IENB1 |
| B | IENB1LO |
| C | IENB2 |
| D | IENB2LO |
| E | IENB3 |
| F | IENB3LO |
| H | |
| J | |
| K | |
| V | |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | |
| U | |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | |
| d | |
| e | |
| h | |
| f | |
| j | |
| k | IENB16 |
| l | IRNB16LO |

J13

| PIN | SIGNAL |
|---|---|
| A | IACK1 |
| B | IACK1LO |
| C | IACK2 |
| D | IACK2LO |
| E | IACK3 |
| F | IACK3LO |
| H | |
| J | |
| K | |
| V | |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | |
| U | |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | |
| d | |
| e | |
| h | |
| f | |
| j | |
| k | IACK16 |
| l | IACK16LO |

J14

| PIN | SIGNAL |
|---|---|
| A | IREQ1 |
| B | IREQ1LO |
| C | IREQ2 |
| D | IREQ2LO |
| E | IREQ3 |
| F | IREQ3LO |
| H | |
| J | |
| K | |
| V | |
| L | |
| M | |
| N | |
| P | |
| R | |
| S | |
| T | |
| U | |
| W | |
| X | |
| Y | |
| Z | |
| a | |
| b | |
| c | |
| d | |
| e | |
| h | |
| f | |
| j | |
| k | IREQ16 |
| l | IREQ16LO |

*Figure 7-6. External Interrupt Subsystem Connectors (Sheet 1 of 2)*

J16

| PIN | SIGNAL |
|---|---|
| A | MCLR |
| B | MCLRLO |
| C | INTCL |
| D | INTCLLO |
| E | GENB |
| F | GENBLO |
| H | CLI |
| J | CLILO |
| K | IBO |
| V | IBOLO |
| L | INTRI |
| M | INTRILO |
| N | INTR2 |
| P | INTR2LO |
| R | IA5 |
| S | IA5LO |
| T | IA4 |
| U | IA4LO |
| W | IA3 |
| X | IA3LO |
| Y | IA2 |
| Z | IA2LO |

*Figure 7-6. External Interrupt Subsystem Connectors (Sheet 2 of 2)*

## PHYSICAL CHARACTERISTICS

A line drawing of the External Interrupt Subsystem is shown in Figure 7-7. The Model 4707-01 and Model 4707-02 Units are mounted in a standard SCC 365004-1 card file. The Model 4707-01 includes the card file, the SCC Power Supply 600153, connector bracket, five ELCO 8016-0566-000-007 connectors, a 28-pair cable and four interface cards (one 401047, two 401052, and one 400715).The Model 4707-02 Interrupt Unit is contained on a 3-inch by 6-inch printed circuit card, SCC 400966; one interrupt level per card with a maximum of 16 cards available.

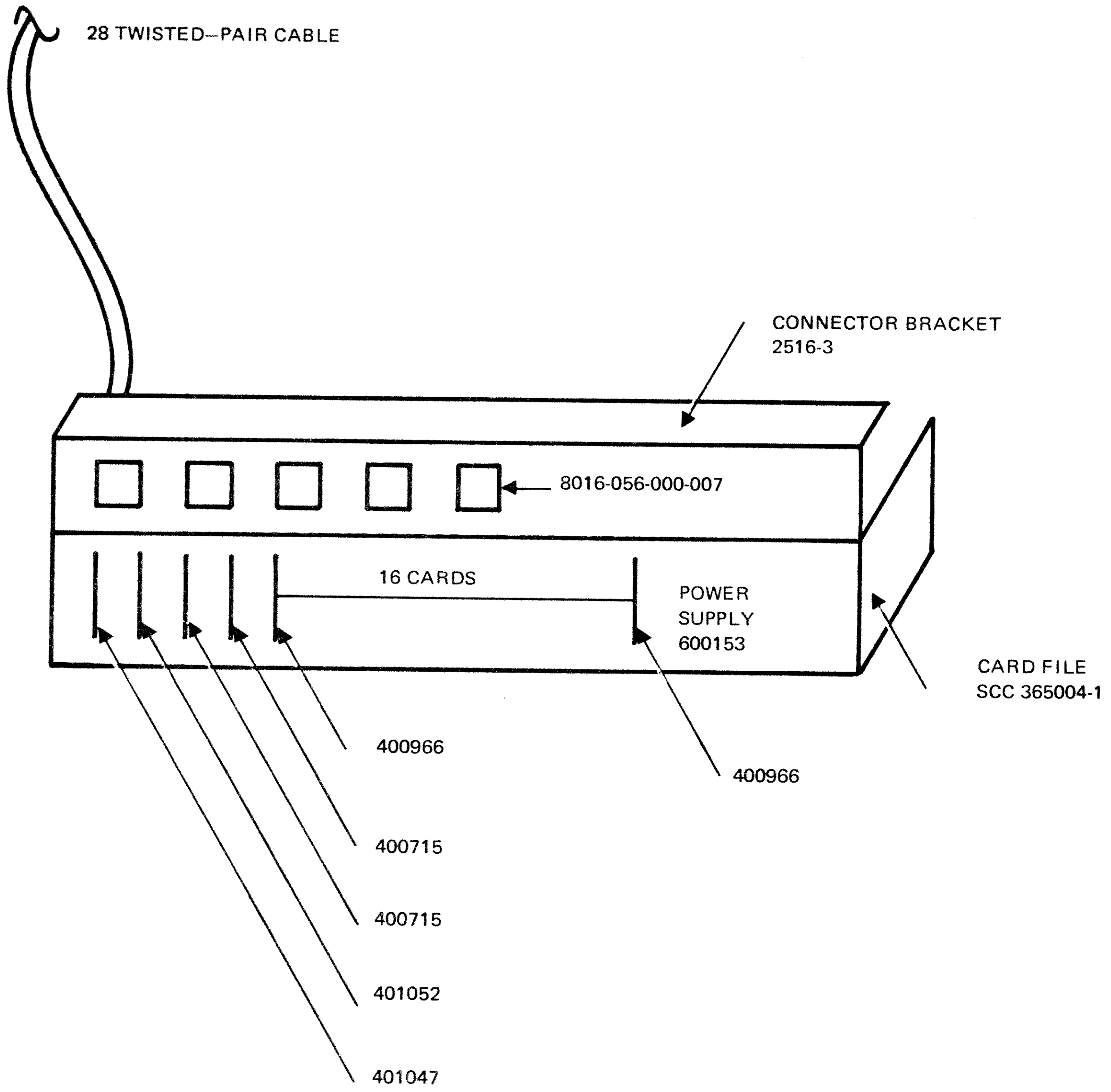


*Figure 7-7. External Subsystem*

# MEMORY GATE INTERFACE

# SECTION VIII

## BASIC DESCRIPTION

The Model 4715-50 Memory Gate Unit provides the SCC 4700 System with a multiple data bus channel for accessing a memory module, or modules, with the CPU, Selector Channel, DMA Channel Controller, or other channel controllers. The basic Memory Gate Element (MGE) provides a 4-data bus channel interface to one memory module as shown in Figure 8-1. Expanded MGE configurations are available with up to an eight channel data bus furnishing access to eight memory modules as shown in Figure 8-2. Configurations of MGE's are available that will provide several CPU's the use of a common memory plus some memory that is privileged for each CPU. With a MGE providing the access to memory, the minimum memory cycle time for a Read or Write operation is 1. 1 microsecond.

## THEORY OF OPERATION

Figure 8-3 shows the input and output signals of a typical data bus. The timing requirements of each of these signals and their interrelationships are discussed in the section covering timing diagrams. Data buses 1, 2, and 3 are available for use by any external channel controller. The interface shown in Figure 8-3 is typical for all three data buses. Data bus 4 is reserved for use by the CPU. Bus 4 may connect directly to the CPU or may connect to a Model 4715-70 Memory Map Unit. When bus 4 connects directly to the CPU a Memory System Interface board is installed with the MGE to provide the proper interface signals and levels for the CPU. When the MGE connects to the Memory Map this board is not required. Sharing of one memory module by more than one data bus is conducted on a priority basis. Data bus 1 usually has the highest priority and data bus 4 the lowest. The establishment of priorities is determined by back panel wiring and can be altered from the previously stated arrangement. If the MGE system is expanded to provide eight data buses accessing several memory modules, the priority of data bus 4 or any other data bus could be wired to any priority level desired. In each MGE there is an MSB address decode for each data bus. By decoding the MSB (2 to 4 bits) of the memory address the only MGE activated will be the one that connects to the desired block of memory. For example, a memory system with 4 MGE's and 4-4K memory modules would be set up so that MGE 1 addresses memory locations 0 through 4K; MGE 2, 4K through 8K; MGE 3, 8K through 12K; MGE 4, 12K through 16K. When a memory request is made to an MGE and the address decode logic indicates that MGE is being addressed, a memory request waiting flip-flop is set. There is a memory request waiting flip-flop for each data bus. If there is no higher priority waiting flip-flop set or the MGE is not actively making a data transfer with another data bus, the MGE goes to an activate state and begins servicing the requesting data bus. A channel controller that is in the process of making a data transfer should not make a second memory request until a FINISH pulse is transmitted to the MGE or until the trailing edge of DRMACK is received by the data bus being serviced.

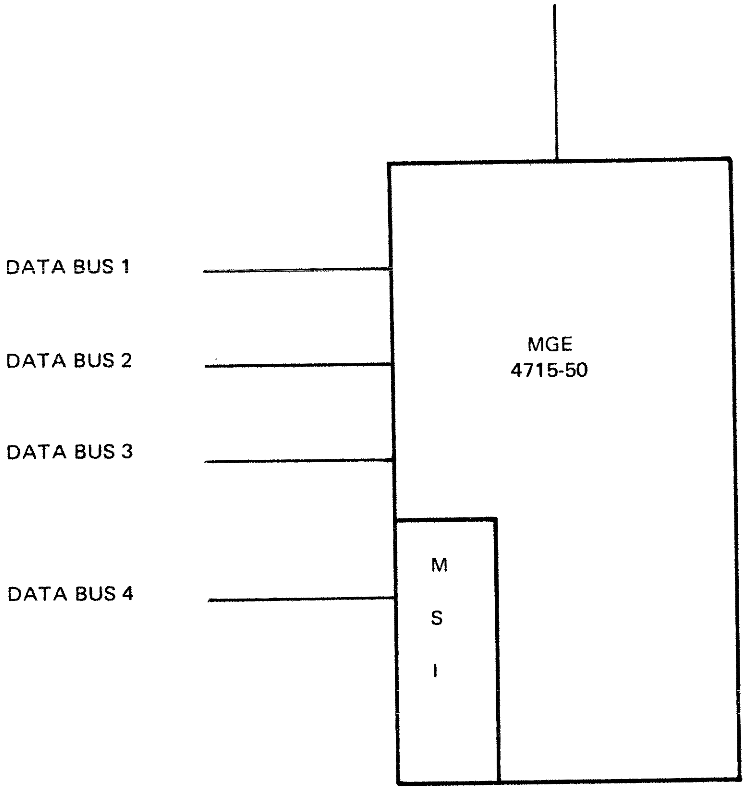


*Figure 8-1. Model 4130 MGE*

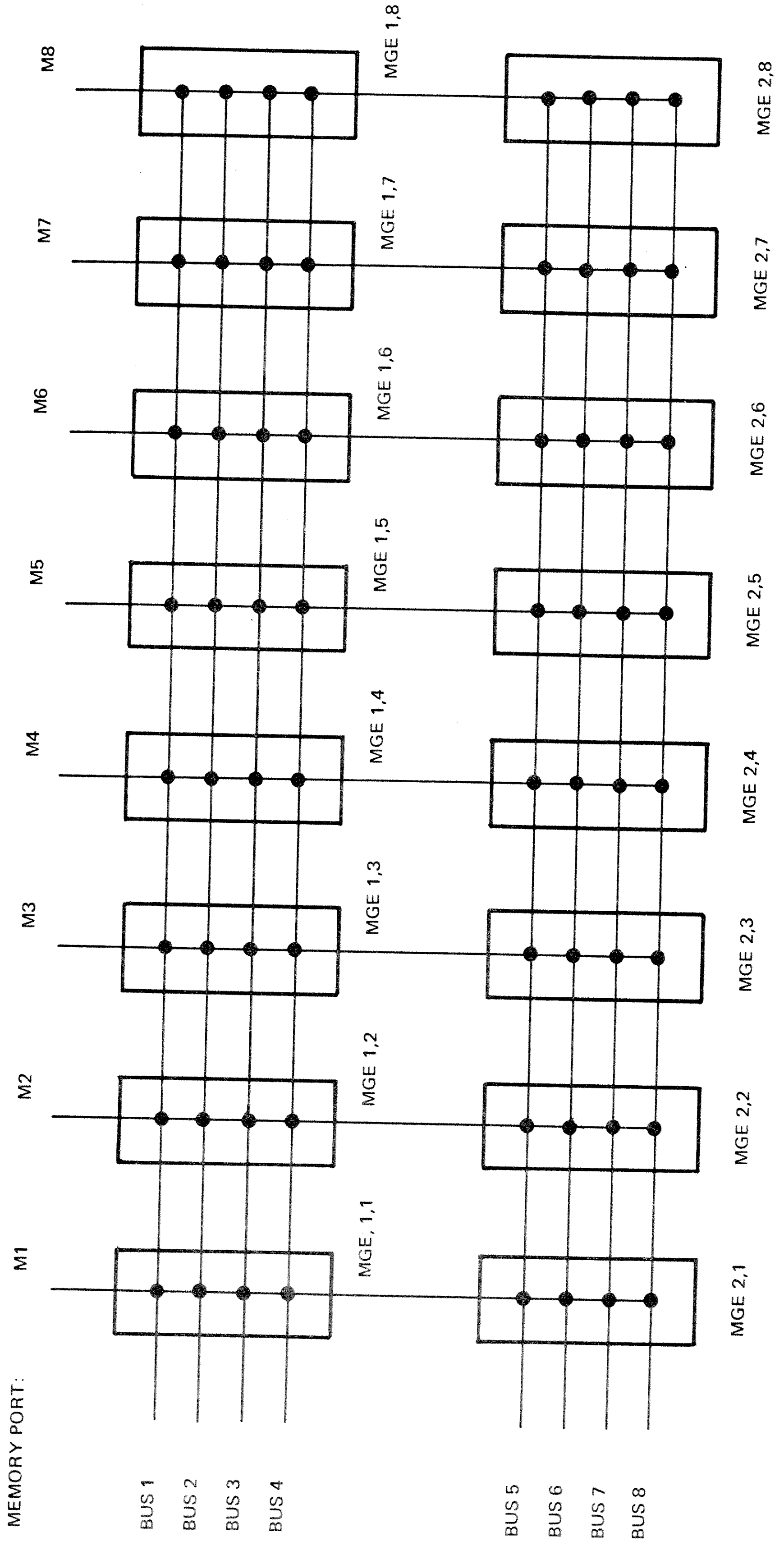


*Figure 8-2. Memory Port System: 16 Model 4715-50 Memory Gate Elements Illustrated*

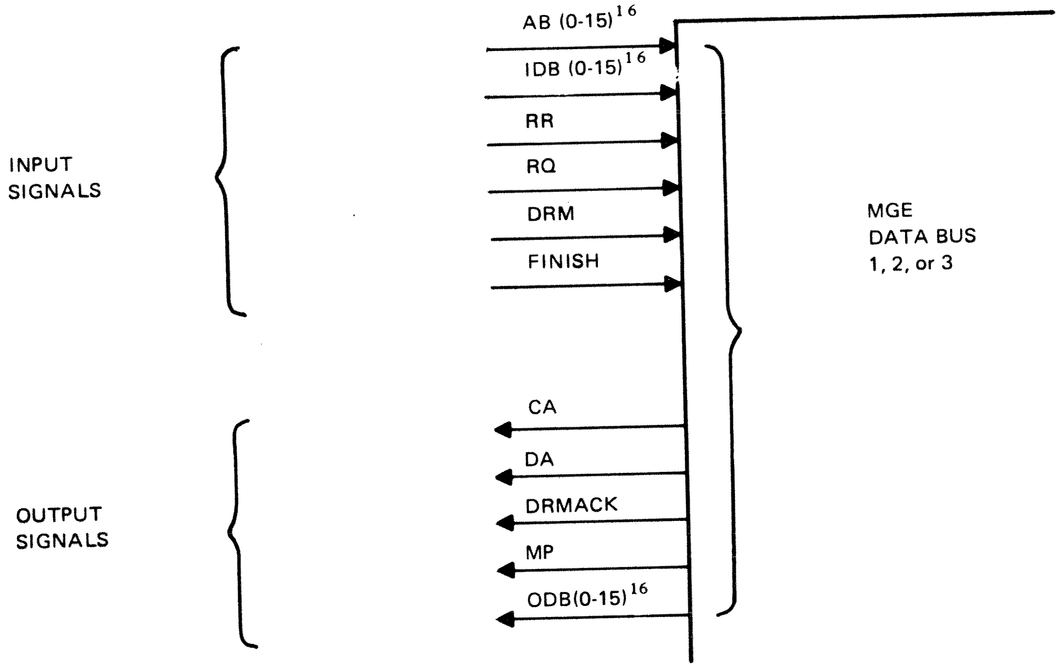


*Figure 8-3. Data and Control Signals*

## SIGNALS

### Input

1. RQ – memory request: The transition from a logical "1" to a logical "0" represents a memory request. RQ should remain at the logical "0" level until a "1" to "0" transition is detected on the CA line from the MGE.

2 RR – Read Restore: This signal should have the same timing as RQ and be activated when a read restore memory request is made.

3. AB00-AB15 – Address data bits

4. DRM – Continue cycle: When DRM makes a "0" to "1" transition, data on the input data lines is loaded into the memory input storage registers. DRM should remain true until DRMACK makes a "1" to "0" transition.

5. IDB00-IDB15 – Input Data Bits: The input data should not change while DRM is at a logic "1".

6. FINISH: Finish is a logical "0" pulse of 50 to 100 nanoseconds in length. It is generated by the channel controller connected to the data bus when it has loaded data from memory into its register and no longer needs service of the memory module. FINISH is generated only when operating in the read restore mode.

### Output

1. CA – Memory Available – CA from the MGE is the logical AND of the memory available signal from the memory module and the active state of the MGE. If a memory module is not busy, CA will make a logical "1" to "0" transition approximately 300 nanoseconds after an RQ has been sent. See the discussion of CA operation in the Timing Diagrams.

2. DA – Data Available – A 60 nanosecond logic "0" pulse indicates data is available from memory. Data should not be loaded into storage registers at the channel controller connected to a data bus until the trailing edge of DA or any time thereafter.

3. ODB00-ODB15 – Output data bits from the MGE.

4. DRMACK – DRM Acknowledge – DRMACK is an acknowledge signal that indicates that data on the Input data lines has been loaded into the memory's input storage register.

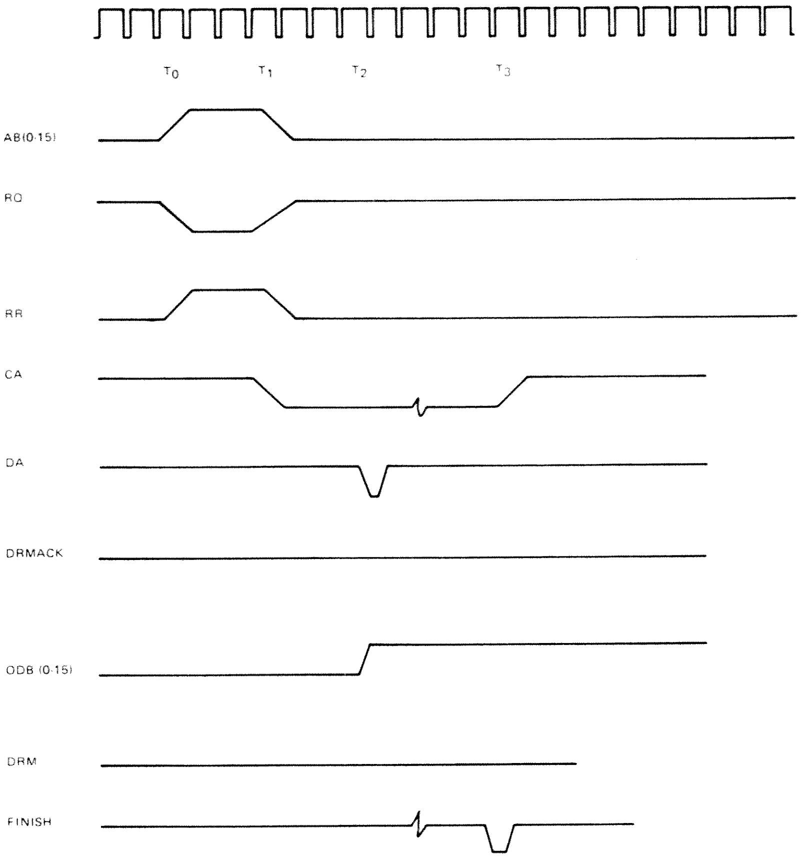
5. MP – Memory Parity – When this option is present in the memory, MP will go from a logic "1" to a "0" 100 nanoseconds after DA if a parity error is detected when reading data from memory. MP will stay at a logic "0" level until a FINISH signal or the trailing edge of DRMACK is detected by the MGE.

## TIMING DIAGRAMS

There are three modes of accessing memory: Read Restore, Clear Write, and Read Modify Write. The timing diagram for each is shown in Figure 8-4 through 8-6. The Read Restore mode is used when it is desired to read data from memory and not destroy the data at that particular address. To perform a Read Restore operation refer to the timing diagram given in Figure 8-4. At T0 the address bits (AB00-15) should be set for the desired memory address. RQ and RR should also go to the logic "1" level at T0. If the MGE which is being addressed is not busy and no higher priority waiting flip flop is set, CA will go from a logic "1" to "0" 300 nanoseconds after T0. At this time (T1) the address bits may be changed and RQ and RR set to a logic "0". If the MGE being addressed is busy T1 may come from 300 nanoseconds to several microseconds after T0. The address bits, RQ, and RR must remain stable from T0 to T1. 330 nanoseconds after T1 a DA pulse will be sent to the channel controller from the MGE. This indicates that data is available on the output data lines (ODB00-15) and may be read by the channel controller any time after the trailing edge of DA. The FINISH signal is to be generated by the channel controller when it has loaded data into its input storage register. If FINISH is generated within 1 microsecond after T0, CA will go to a logic "1" upon receipt of FINISH by the MGE. If FINISH is generated more than 1 micrósecond after T0 then CA will go to a logic "1" 1 microsecond after T0.

The Clear Write mode is used whenever it is desired to load new data into memory. The data that may have been at any given address when a clear write operation is done is destroyed. To perform a Clear Write operation see timing diagram Figure 8-5. At T0 the address bit (AB00-15) and the input data (IDB00-15) should be set for the desired memory address and the data that is to be loaded into that address. RQ and DRM should be set to a logic "1" at T0. If the MGE which is being addressed is not busy and no higher priority waiting, the flip flop is set. CA will go from a logic "1" to a "0" 300 nanoseconds after T0. At this time (T1) the address bits may be changed and RQ may be set to a logic "0." If the MGE being addressed is busy T1 may come from 300 nanoseconds to several microseconds after T0. The address bits and RQ must remain stable until T1. 270 nanoseconds after T1 DRMACK will be sent from the MGE to the channel controller. On the leading edge of DRMACK (T2) DRM may be reset to a logic "0" and the input data (IDB00-15) may be changed. The input data and DRM must remain stable from T0 to T2. On the trailing edge of DRMACK (T3) CA will be set to a logic "1".

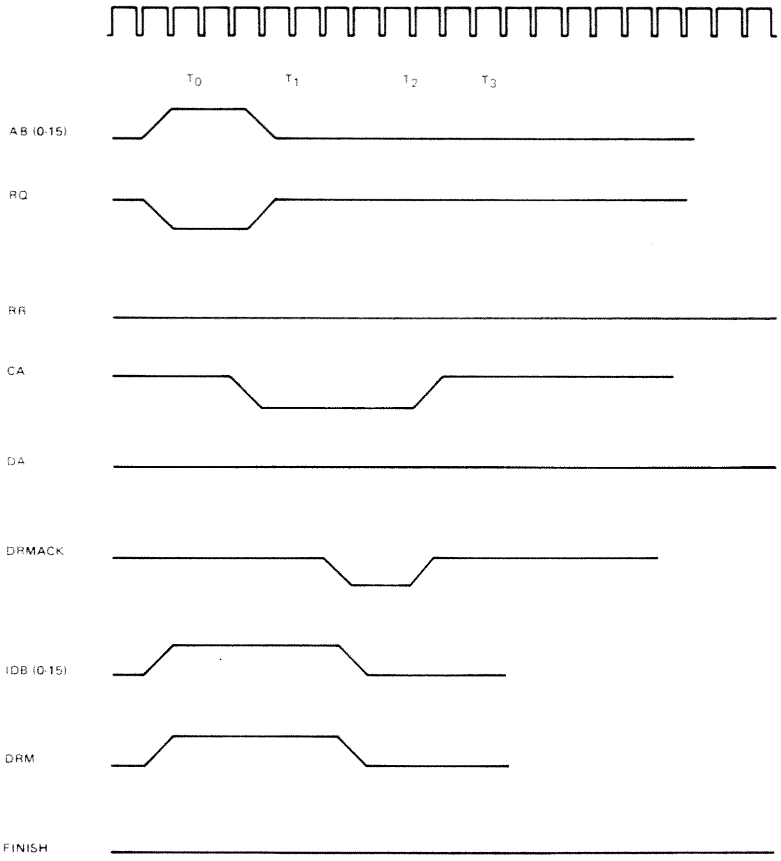
The Read Modify Write mode enables the user to read data from memory, modify it, and write it back in the same memory address location with one memory cycle rather than two if the read and write operations were performed separately. At T0 the address data bits (AB00-15) should be set to the desired memory address and RQ should be set to a logic "1". If the MGE which is being addressed is not busy and no other higher priority writing flip flop is set CA will go from a logic "1" to a "0" 300 nanoseconds after T0. At this time (T1) the address bits may be changed and RQ may be set to a logic "0". If the MGE being addressed is busy T1 may come from 300 nanoseconds to several microseconds after T0. The address bits and RQ must remain stable from T0 to T1. 330 nanodata is available on the output data lines (ODB00-15) and may be read by the channel controller any time after the trailing edge of DA. When the channel controller has read the data from memory and modified it as desired the new data should be placed on the input data lines (IDB00-15). The changing of DRM to a logic "1" will cause the memory to read the data on the input data lines and write it in memory. When DRMACK is received by the channel controller (T4) DRM should be set to a



Read Restore Timing

| Time Span | Min | Typ | Max |
|---|---|---|---|
| $T_0$-$T_1$ | 250ns | 300ns | None |
| $T_1$-$T_2$ | 280ns | 330ns | 380ns |
| $T_2$-$T_3$ | governed by channel controller | | |

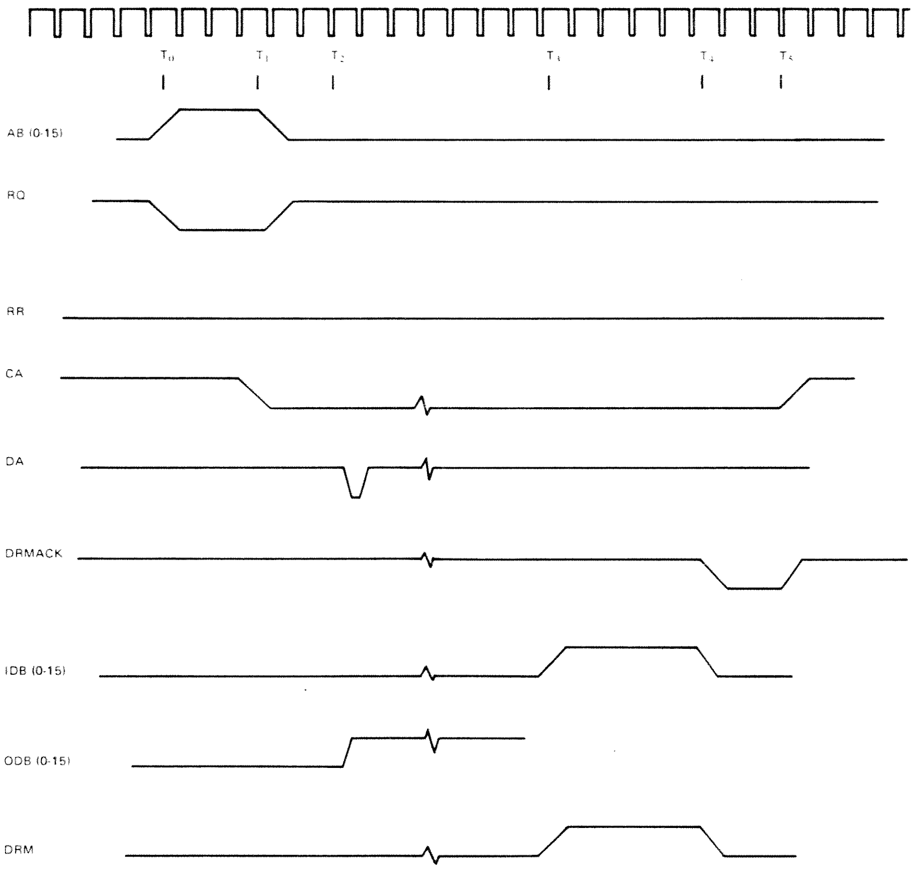
*Figure 8-4. Read Restore Timing*



Clear Write Timing

| Time Span | Min | Typ | Max |
|---|---|---|---|
| $T_0$-$T_1$ | 250ns | 300ns | None |
| $T_1$-$T_2$ | 220ns | 270ns | 320ns |
| $T_2$-$T_3$ | 150ns | 200ns | 250ns |

*Figure 8-5. Clear Write Timing*



Read Modify Write Timing

| Time Span | Min | Typ | Max |
|---|---|---|---|
| $T_0$-$T_1$ | 250ns | 300ns | None |
| $T_1$-$T_2$ | 280ns | 330ns | 380ns |
| $T_2$-$T_3$ | governed by channel controller | | |
| $T_3$-$T_4$ | 420ns | 470ns | 520ns |
| $T_4$-$T_5$ | 150ns | 200ns | 250ns |

*Figure 8-6. Read Modify Write Timing*

logic "0" and the data on the input data lines may be changed. DRM and the data on the input data lines should remain stable from T3 to T4. On the trailing edge of DRMACK (T5) CA will go to the logic "1" state.

All signals are listed as they appear on the connectors in Figure 8-7.

## BUS OUTPUT DATA

| PIN | SIGNAL |
|---|---|
| A | ODB00 |
| B | |
| C | ODB01 |
| D | |
| E | ODB02 |
| F | |
| H | ODBC3 |
| J | |
| K | ODB04 |
| V | |
| L | ODB05 |
| M | |
| N | ODB06 |
| P | |
| R | ODB07 |
| S | |
| T | ODB08 |
| U | |
| W | ODB09 |
| X | |
| Y | ODB10 |
| Z | |
| a | ODB11 |
| b | |
| c | ODB12 |
| d | |
| e | ODB13 |
| h | |

| PIN | SIGNAL |
|---|---|
| f | ODB14 |
| j | |
| k | ODB15 |
| l | |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 8-7. Bus Data (Sheet 1 of 4)*

BUS INPUT DATA

| PIN | SIGNAL |
|---|---|
| A | IDB00 |
| B | |
| C | IDB01 |
| D | |
| E | IDB02 |
| F | |
| H | IDB03 |
| J | |
| K | IDB04 |
| V | |
| L | IDB05 |
| M | |
| N | IDB06 |
| P | |
| R | IDB07 |
| S | |
| T | IDB08 |
| U | |
| W | IDB09 |
| X | |
| Y | IDB10 |
| Z | |
| a | IDB11 |
| b | |
| c | IDB12 |
| d | |
| e | IDB13 |
| h | |

| PIN | SIGNAL |
|---|---|
| f | IDB14 |
| j | |
| k | IDB15 |
| l | |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 8-7. Bus Data (Sheet 2 of 4)*

BUS CONTROL

| PIN | SIGNAL |
|---|---|
| A | CA |
| B | |
| C | DA |
| D | |
| E | DRMACK |
| F | |
| H | RQ |
| J | |
| K | RR |
| V | |
| L | DRM |
| M | |
| N | MP |
| P | |
| R | |
| S | |
| T | |
| U | |
| W | |
| X | |
| Y | FINISH |
| Z | |
| a | |
| b | |
| c | |
| d | |
| e | |
| h | |

| PIN | SIGNAL |
|---|---|
| f | |
| j | |
| k | |
| l | |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 8-7. Bus Data (Sheet 3 of 4)*

BUS ADDRESS

| PIN | SIGNAL |
|---|---|
| A | AB00 |
| B | |
| C | AB01 |
| D | |
| E | AB02 |
| F | |
| H | AB03 |
| J | |
| K | AB04 |
| V | |
| L | AB05 |
| M | |
| N | AB06 |
| P | |
| R | AB07 |
| S | |
| T | AB08 |
| U | |
| W | AB09 |
| X | |
| Y | AB10 |
| Z | |
| a | AB11 |
| b | |
| c | AB12 |
| d | |
| e | AB13 |
| h | |

| PIN | SIGNAL |
|---|---|
| f | AB14 |
| j | |
| k | AB15 |
| l | |
| m | |
| n | |
| p | |
| r | |
| s | |
| t | |
| u | |
| v | |
| w | |
| x | |
| y | |
| JJ | |
| z | |
| AA | |
| BB | |
| CC | |
| DD | |
| EE | |
| FF | |
| HH | |
| KK | |
| LL | |
| MM | |
| NN | |

*Figure 8-7. Bus Data (Sheet 4 of 4)*

All input lines are terminated as shown in Figure 8-8.

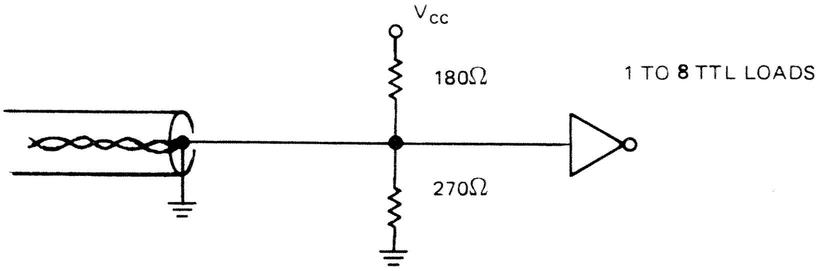


*Figure 8-8.*

All output lines are driven by the circuit shown in Figure 8-9.

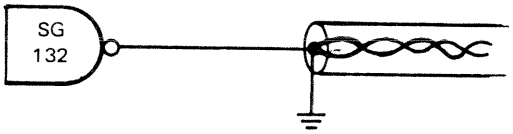


*Figure 8-9.*

All signals that are transmitted over a cable should use a tw isted pair of wires for each signal. The common wire of each pair should be connected to ground at each end of the cable as close as is physically possible to its respective driver or terminator. A logic "1" is defined as a 3.0 to 5.0 volt level. A logic "0" is defined as 0 to 0.4 volt signal level.

## Scientific Control Corporation

Dallas, Texas 214–242-6555 TWX–910-860-5956

**Order by part number from above address or call the nearest sales office for additional information.**

EASTERN

REGIONAL OFFICE
Washington, D.C.
301–779-3330

DISTRICT OFFICES
College Park, Md.
301–779-2510
Orlando, Fla.
305–855-5833
New York, N.Y.
201–335-3001
Boston, Mass.
617–449-0880

CENTRAL

REGIONAL OFFICE
Dallas, Texas
214–358-1331

DISTRICT OFFICES
Dallas, Texas
214–358-1332
Houston, Texas
713–526-5721
Chicago, Ill.
312–297-2470
Denver, Colorado
303–322-0516
Detroit, Mich.
Branch Office

WESTERN

REGIONAL OFFICE
San Francisco, Calif.
415–328-8980

DISTRICT OFFICES
Palo Alto, Calif.
415–328-8981
Los Angeles, Calif.
213–443-0143