CAL DATA 1
MACROBUS CHANNEL ADAPTER
(P/N C81080300)

TECHNICAL MANUAL C21518013-X0

DOCUMENT C21518013 Revision X0 February 1975

Cal Data, MACROBUS, QUADBOARD and HEXBOARD are trademarks of California Data Processors.

The information herein is the property of California Data Processors. Transmittal, receipt or possession of the information does not express, license or imply any rights to use, sell or manufacture from this information and no reproduction or publication of it, in whole or in part shall be made without written authorization from an officer of the above firm.



california data processors

REVISIONS

| Revision | Date | Approval | Description |
|----------|------|----------|-------------|
| xo | 3-75 | yor | Preliminary |

The revision history of each page in this document is indicated below:

| | Re | visi | on | | Re | visi | On |
|--|---|------|----|-------------------------------|---|------|----|
| Page | хo | | | Page | хо | | |
| Page i ii 1-1 1-2 1-3 2-1 2-2 2-3 2-4 2-5 2-6 3-1 3-2 4-1 4-2 4-3 4-4 4-5 4-6 5-1 5-2 5-3 6-1 A-1 A-2 A-3 A-4 A-5 A-6 A-7 A-8 B-1 C-1 | \approx) $11111111111111111111111111111111111$ | | | Page C-3 C-4 C-5 C-6 C-7 C-8 | xo \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ | | |
| C-2 | ~ | | | · | | - | |

CONTENTS

| SECT | ION 1: INTRODUCTION | . Page |
|------|--|--------|
| 1.1 | SCOPE | . 1-1 |
| 1.2 | DOCUMENTATION | . 1-1 |
| | 1.2.1 Publications | . 1-1 |
| | 1.2.2 Engineering Drawings | . 1-1 |
| | 1.2.3 Abbreviations and Conventions | . 1-1 |
| SECT | ION 2: FUNCTIONAL DESCRIPTION | |
| 2.1 | GENERAL | |
| 2.2 | MAJOR FUNCTIONAL AREAS | |
| | 2.2.1 MACROBUS Address Register | |
| | 2.2.2 MACROBUS Output Data Register | |
| | 2.2.3 MACROBUS Input Data Register | |
| | 2.2.4 MACROBUS Control Logic and Error Flags | |
| 2.3 | SPECIFICATIONS | . 2-6 |
| SECT | TION 3: PHYSICAL DESCRIPTION | |
| 3.1 | GENERAL | . 3-1 |
| 3.2 | CONNECTORS | . 3-1 |
| SECT | CION 4: INTERFACE | |
| 4.1 | GENERAL | . 4-1 |
| 4.2 | MACROBUS INTERFACE | . 4-1 |
| | 4.2.1 Access Control Lines | . 4-2 |
| | 4.2.2 Data-Transfer Lines | . 4-3 |
| | 4.2.3 Other Lines | |
| 4.3 | MACROBUS USE | |
| | 4.3.1 Request Priority | . 4-5 |
| | 4.3.2 Data Transfers | |
| | 4.3.3 CPU Interrupts | |
| SECT | TION 5: PROGRAMMING | |
| 5.1 | GENERAL | . 5-1 |
| 5.2 | PERIPHERAL-DEVICE REGISTERS | . 5-1 |
| 5.3 | INTERRUPTS | . 5-3 |
| SECT | CION 6: MAINTENANCE | |
| 6.1 | GENERAL | . 6-1 |
| 6.2 | PREVENTIVE MAINTENANCE | |
| 6.3 | CORRECTIVE MAINTENANCE | |



APPENDICES

| | | Dage |
|--|--|---|
| APPENDIX | A: CONNECTOR PIN ASSIGNMENTS | Page |
| APPENDIX | B: ASSIGNED ADDRESSES | |
| APPENDIX | C: MACROBUS OPERATIONS | |
| C. C. C.2 DA C. | CROBUS ACCESS | C-1 C-1 C-5 C-5 C-7 |
| | | |
| | TABLES | |
| <u>Table</u> | <u>Title</u> | Page |
| 1-1 2-1 4-1 4-2 A-1 A-2 A-3 A-4 A-5 A-6 A-7 A-8 | Abbreviations | 1-3 2-6 4-2 4-4 A-1 A-2 A-3 A-4 A-5 A-6 A-7 |
| | ILLUSTRATIONS | |
| Figure | <u>Title</u> | Page |
| 1-1 2-1 2-2 2-3 3-1 5-1 C-1 C-2 | Relationship of Publications to Cal Data 1 System Elements. Cal Data 1 Computer System Organization | 1-2 2-2 2-3 2-4 3-2 5-2 C-2 C-4 C-6 |
| C-4 | DATO and DATOB Timing | C-8 |



SECTION 1 INTRODUCTION

1.1 SCOPE

This manual provides the information needed to understand and maintain the Cal Data 1 MACROBUS Channel Adapter (MCA, part number C81080300) when used with the drawing package provided. The information in this manual is for the use of a skilled technician familiar with standard test equipment, solid-state logic theory, common maintenance practices and standard troubleshooting techniques. A basic knowledge of design principles and circuits used in small computers is assumed, hence no tutorial material of this kind is included.

As a stand-alone publication, this manual has a good functional and physical description of the MCA, providing the information needed to understand the capabilities and features of the MCA and to plan a system using it. The maintenance coverage of this manual is commensurate with the prerequisite skills and knowledge of the defined user, characteristics of the product and maintainability requirements established by Cal Data.

1.2 DOCUMENTATION

This manual describes the MCA of a Cal Data 1 Computer system.

The following paragraphs define publications and conventions that support this manual.

1.2.1 Publications

Figure 1-1 shows the relation of hardware elements and technical documentation. Controlled copies of documents, given in accordance with the terms of a purchase contract, are kept current for the life of the product.

1.2.2 Engineering Drawings

For maintenance purposes, this manual is supported by a theory of operation and a drawing package that contains schematic diagrams, assembly drawings and other required engineering drawings. The drawing package is updated with the latest revision of each drawing.

1.2.3 Abbreviations and Conventions

Table 1-1 lists the abbreviations found in this manual.

Conventions used in the test of this manual include:

- a. Equipment panel nomenclature is reproduced in all upper-case characters.
- b. The proper names of instructions, microcommands and signals are capitalized.

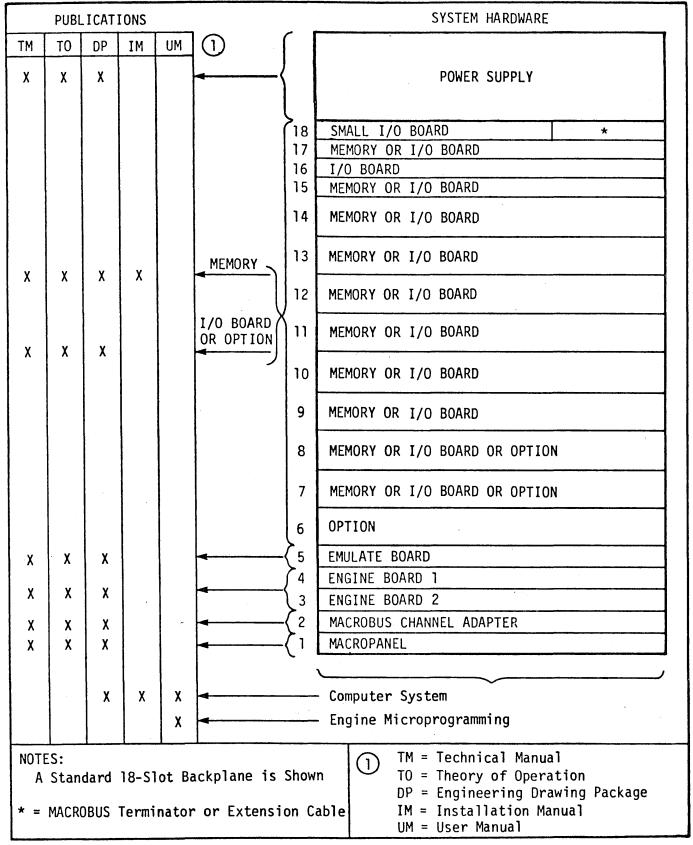


Figure 1-1. Relationship of Publications to Cal Data 1 System Elements



- c. ZERO and ONE are used to indicate binary logic '0" and "1" states, respectively.
- d. Hexadecimal numbers are preceded by a dollar sign for easy identification.
- e. A colon (:) is used to indicate a range of bits. For example, the range of Address bits Al2 to AO3 is written Al2:AO3.

Table 1-1. Abbreviations

| Abbreviation | Meaning |
|---|--|
| Abbreviation Cal Data CPU MCA MMU I/O AR AL MB AB DR DL RR PC PS cm cfm lps ns | California Data Processors central processing unit (Engine) MACROBUS Channel Adapter Memory Management Unit input/output MACROBUS address register AR latch register M bus A-operand bus MACROBUS data register DR latch register data-read register program counter processor status register centimeter cubic feet per minute liters per second nanosecond |
| Vdc ^O C | volt, direct current degrees, Celsius |
| K psi | 1,024 (addresses or memory locations) pounds per square inch |
| kg | kilogram |

SECTION 2 FUNCTIONAL DESCRIPTION

2.1 GENERAL

The Cal Data 1 Computer is specifically designed for broad applications as an efficient emulator of selected computer architectures. It is often desirable to provide an external I/O structure that is compatible with a particular computer as well as to emulate the internal instruction repertoire. Because the Cal Data 1 Computer system consists of a modular set of hardware and software elements, nearly any system configuration can be assembled.

The main hardware data path of the system is a universal, synchronous bus called the Microbus (Figure 2-1). The Microbus is used by all elements of the system including the Engine (CPU). One element that can share the Microbus is the MCA, which conditions the Microbus to interface with a wide range of peripheral devices on an I/O channel called the MACROBUS (Figure 2-2). The MACROBUS is one of the I/O channels that can be provided with the Cal Data 1 Computer. System devices attached to the MACROBUS can be designed to communicate directly with each other independently of the CPU.

the MACROBUS is an asynchronous, bidirectional I/O channel that can accommodate many types of controllers, memory devices, system interfaces and special devices. The only requirement is that such devices obey MACROBUS use rules. All components attached to the MACROBUS are treated as I/O devices by the CPU. Because memory modules are treated as independent I/O devices, semiconductor modules of varying size and speed can be freely mixed with high-speed core memory. All MACROBUS devices with the necessary circuitry may communicate directly with memory, with each other and with the CPU (via the MCA).

Cal Data offers several MACROBUS peripheral subsystems to enhance system application. The subsystems offered to support normal programming and system development operations are:

- a. Paper tape reader. High-speed photoelectric reader, 300 characters per second, fanfold tape.
- b. Paper tape punch. High-speed punch, 75 characters per second, fanfold tape.
- c. Card reader. High-speed photoelectric card reader, 300 cards per minute with code conversion in the controller.
- d. Line printer. 80- or 132-column printer, 125 or 200 lines per minute.
- e. Memory extensions.

The user can readily interface other devices with the MACROBUS using simple design rules.

2.2 MAJOR FUNCTIONAL AREAS

The MCA contains the data, address and control circuitry for all parallel MACROBUS operations in the system. Figure 2-3 is a block diagram of the MCA.



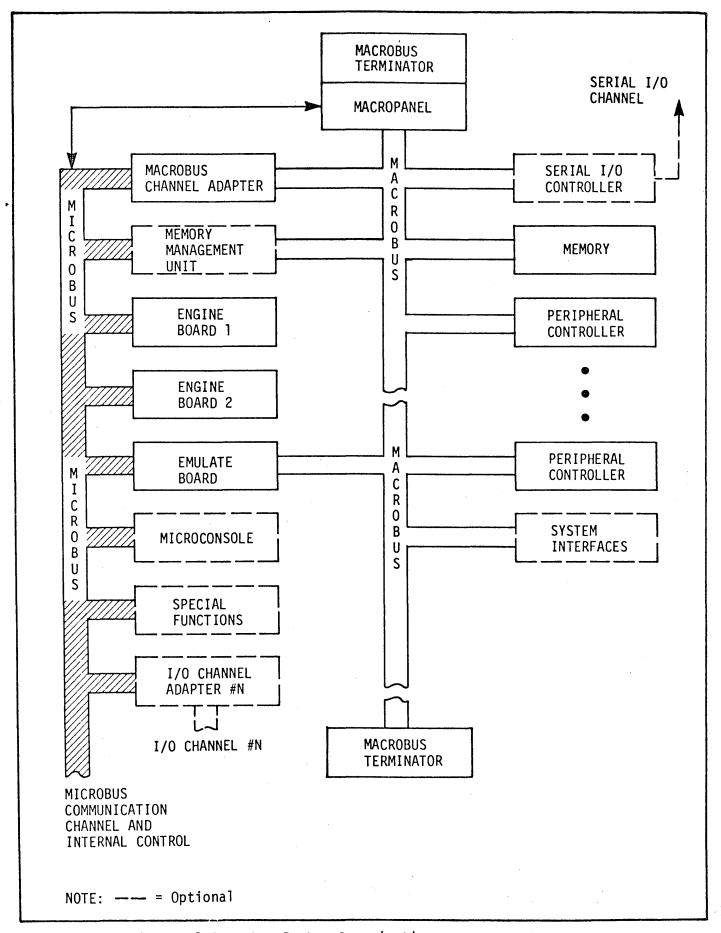


Figure 2-1. Cal Data 1 Computer System Organization

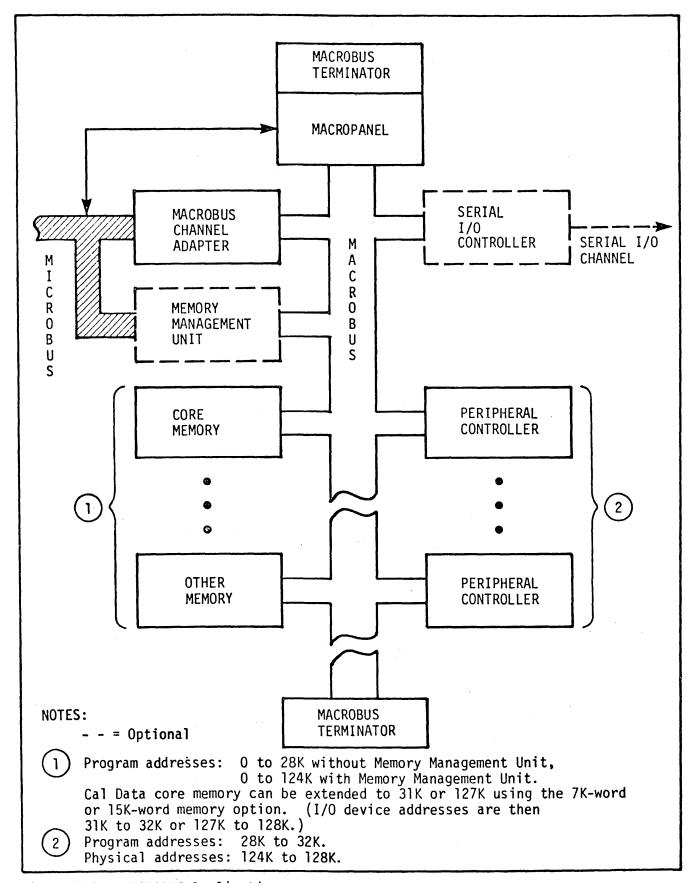
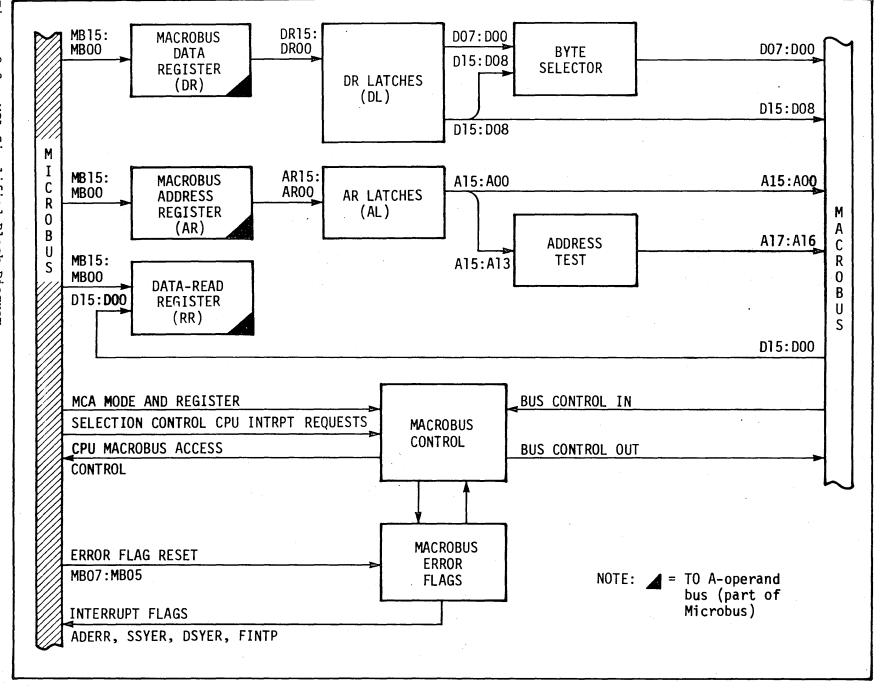


Figure 2-2. MACROBUS Application

C21518013-X0



The MCA consists of internal registers and control circuitry directly accessible to the Engine microprogram. MACROBUS control circuitry handles MACROBUS access requests. The priorities of these requests are decoded and organized by the MACROBUS priority logic located on the Emulate Board before transmission to the MCA.

2.2.1 MACROBUS Address Register

The 16-bit address register (AR) stores address words needed for MACROBUS operations. Associated with AR is a 16-bit latch register (AL) that is loaded with the contents of AR at the time a MACROBUS operation is initiated by a microcommand. The address is held in AL during the entire MACROBUS cycle. AR can be loaded with a new value at any time after the MACROBUS cycle is initiated. AR is loaded by microcommand via the M bus (MB), which is part of the Microbus.

The 16-bit address word in AL is converted to an 18-bit physical address placed on the MACROBUS. To accomplish this, the three most-significant bits of AL are monitored by address test circuits to detect a MACROBUS address in the range 28K to 32K, which is assigned to I/O devices. When any address in this range is detected, the two most-significant bits of the 18-bit MACROBUS address are set to generate a physical address in the range 124K to 128K. Address bits Al7:Al6 are set to ZERO when the AL word is in the range 0 to 28K. For extended MACROBUS addressing beyond 32K (28K memory and 4K I/O addresses), the Cal Data 1 Memory Management Unit is required.

The contents of AR can be read by microcommand via the A-operand bus (AB), which is part of the Microbus.

2.2.2 MACROBUS Output Data Register

The 16-bit data register (DR) stores data words (or bytes) needed for MACROBUS operations. Associated with DR is a 16-bit latch register (DL) that is loaded with the contents of DR at the time a MACROBUS write operation is initiated by a microcommand. The output word is held during the entire MACROBUS cycle. DR can be loaded with a new value at any time after the MACROBUS cycle is initiated. DR is loaded by microcommand via

The output of DL is transferred to the MACROBUS through a set of gates that performs any necessary output byte conversions.

The contents of DR can be read by microcommand via AB.

2.2.3 MACROBUS Input Data Register

Data inputs from the MACROBUS are routed under microprogram control to the data-read register (RR).

RR is used to receive data words and operands. This 16-bit register receives inputs from the MACROBUS via a set of gates that performs any necessary input byte conversions. RR can also be loaded by microcommand via MB.

The contents of RR can be read by microcommand via AB.

2.2.4 MACROBUS Control Logic and Error Flags

The MCA monitors all requests for MACROBUS use from all sources. MACROBUS control logic within the MCA determines whether the request is an ordinary priority Bus Request (BR) or a high-priority nonprocessor request (NPR). If it is a BR and no NPR is pending, MACROBUS priority logic on the Emulate Board is utilized to determine the highest-priority BR and assert a corresponding Bus Grant (BG). The MCA then handles MACROBUS synchronization and data-transfer control.

If the request is an NPR, the Emulate Board is inhibited from honoring BRs and the MCA asserts Nonprocessor Grant (NPG).

If no MACROBUS request is pending, control is given to the CPU by the MCA.

The MCA continuously checks for MACROBUS errors. In case of a detected error, the MCA issues one of three error flags plus a fatal-error interrupt signal (FINTP) to the CPU, where firmware initiates the required error routine. The flags are address error (ADERR), slave-synchronization error (SSYER) and double slave-synchronization error (DSYER).

2.3 SPECIFICATIONS

General specifications for the MCA are given in Table 2-1.

Table 2-1. Cal Data 1 MCA Specifications

| Characteristic | Specification |
|---------------------|---|
| Interface | Two connectors (A and B) provide the MACROBUS. Six connectors (C, D, E, F, Jl and J2) interface with the main Microbus. |
| Configuration | Hex-width printed-circuit board |
| Dimensions | 15.7 by 8.9 inches (39.9 by 22.7 cm) |
| Mounting Centers | 0.75 inch (1.9 cm) recommended minimum |
| Ambient Temperature | 0 to +50° C with 115 cfm (54 lps) airflow |
| Ambient Humidity | 10 to 90%, without condensation |



SECTION 3 PHYSICAL DESCRIPTION

3.1 GENERAL

The MCA board (Figure 3-1) is a hex-width board 15.7 by 8.9 inches (39.9 by 22.7 cm) that normally plugs into slot 2 of the Cal Data computer chassis.* The right-hand edge of the board has a 1.0 by 5.5 inch (2.5 by 14.0 cm) cutout as clearance for the side-mounted cooling fans in the chassis.

There are no controls or adjustable elements on the MCA.

3.2 CONNECTORS

There are six printed-circuit connectors (A to F) on the bottom edge of the board, and two (Jl and J2) on the top edge. Connectors A and B interface with the MACROBUS. Connectors C to F, and Jl and J2 interface with the main computer Microbus. Connectors A to F are standard backplane connectors. Connectors Jl and J2 plug into two small processorinterconnection boards.

^{*}Because of the universal connections in the CPU area of the chassis, the MCA can operate in any slot from 1 to 5.

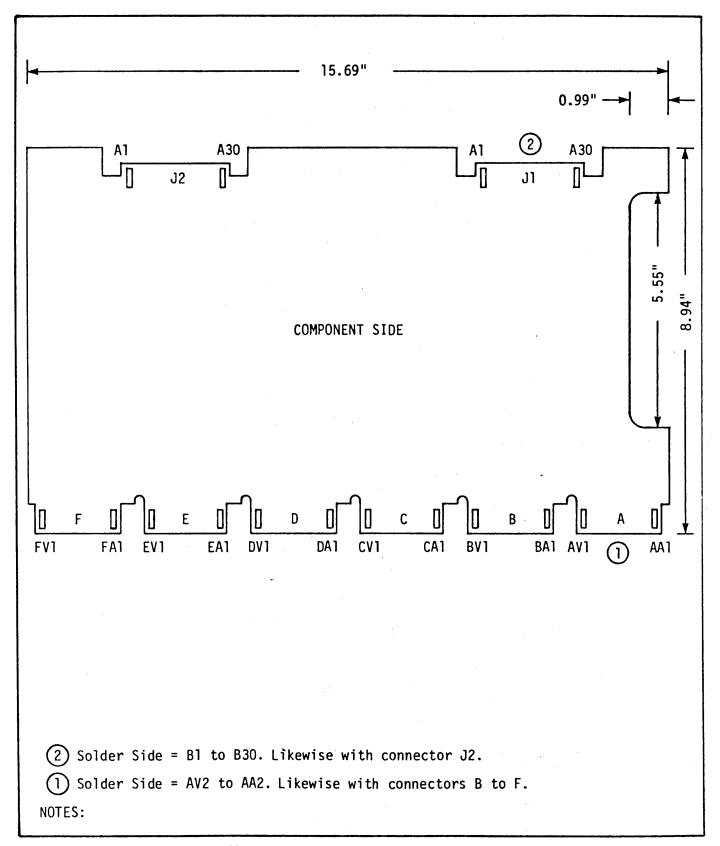


Figure 3-1. MCA Board Configuration

SECTION 4 INTERFACE

4.1 GENERAL

The MCA is the connecting link between the main computer Microbus and the MACROBUS I/O channel. The Microbus is described in the Cal Data 100 Engine Technical Manual, C21518008. This section describes the Cal Data 1 MACROBUS in detail.

Communication between two devices on the MACROBUS is in a master/slave relationship. During a MACROBUS operation, the master device has control of the MACROBUS when communicating with the other device (the slave). Since all MACROBUS elements are addressed in an identical manner, it is possible to implement direct operations between bus elements (device to/from memory, device to/from device) without involving the CPU itself.

A priority structure determines which device obtains control of the MACROBUS, and every device on the MACROBUS capable of becoming bus master has an assigned priority. When two devices with identical priority levels simultaneously request use of the MACROBUS, the device that is electrically closest to the bus receives control. Provision is made for transfers of bus mastership to higher-priority devices requesting MACROBUS

All requests for bus mastership are made to the MCA, which performs the I/O system "housekeeping". A device requesting MACROBUS access is given bus mastership when no higher-priority device is waiting. Once access is obtained, the bus master retains control of the MACROBUS until the necessary operation is completed. Control is then relinquished to the highest-priority device requesting access. Thus, the total available I/O bandwidth is automatically distributed among all devices sharing the MACROBUS. When no other device has control of the MACROBUS, mastership is automatically assigned to the CPU by the MCA.

Communication on the MACROBUS is interlocked, and each control signal issued by the master must be acknowledged by a response from the slave to complete the transfer. The MACROBUS transfer rate is thus automatically balanced among different devices. Communication is independent of the physical bus length and the response time of the master and slave devices.

4.2 MACROBUS INTERFACE

The MACROBUS consists of 54 active and two reserved signal lines. All MACROBUS devices, including the MCA, are connected in parallel to these lines. The signal names and mnemonics are listed in Table 4-1.

Table 4-1. Cal Data 1 MACROBUS Signals

| Туре | Name | Mnemonic | Lines |
|-----------|------------------------|-----------|-------|
| | Nonprocessor Request | NPR | 1 |
| | Bus Request | BR7:BR4 | 4 |
| Access | Nonprocessor Grant | NPG | 1 |
| Control | Bus Grant | BG7:BG4 | 4 |
| | Selection Acknowledge | SACK | 1 |
| | Bus Busy | BBSY | 1 |
| | Interrupt Request | INTR | 1 |
| | Address | A17:A00 | 18 |
| Data | Mode Control | Cl and CO | . 2 |
| Transfer | Data | D15:D00 | 16 |
| ITALISTEL | Master Synchronization | MSYN | 1 |
| | Slave Synchronization | SSYN | 1 |
| | Initialization | INIT | 1 |
| | AC Low | ACLO | 1 |
| Other | DC Low | DCLO | 1 |
| | Reserved | PA | 1 |
| | Reserved | PB | 1 |
| | | | |

Note: Signals on the MACROBUS are asserted when low.

4.2.1 Access Control Lines

The MACROBUS access control lines are:

- a. Bus Requests. BR7:BR4 are used by peripheral devices to request control of the MACROBUS via the MCA. More than one device can share a request line.
- b. Bus Grants. BG7:BG4 are the MCA responses to successful MACROBUS requests. Only one at a time is asserted, and only at the end of instruction execution and in accordance with rules of priority. Each MACROBUS device of the same priority level passes the grant signal to the next device on the line, unless the device has requested MACROBUS, mastership, in which case the requesting device blocks the grant signal from the following devices and assumes MACROBUS control.
- c. Nonprocessor Request. NPR is a MACROBUS request from a peripheral device and is used to gain MACROBUS control for data transfers not involving the CPU.
- d. Nonprocessor Grant. NPG is the MCA response to a successful NPR. It can occur at the end of any MACROBUS cycle (other than data input, pause).
- e. Selection Acknowledge. SACK is asserted by a MACROBUS requesting device that has received a grant. MACROBUS control passes to the device when the current bus master completes operation. (If SACK is not received by the MCA within 10 microseconds after the grant is issued, a time-out occurs and the grant is cleared automatically by the MCA. This does not, however, cause a time-out error trap.)



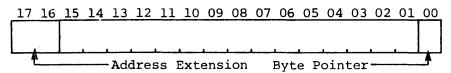
- f. Bus Busy. BBSY is asserted by the bus master to indicate that the MACROBUS is occupied.
- g. Interrupt Request. INTR is asserted by the bus master to start a program interrupt operation in the CPU. INTR can be asserted only by a device gaining control of the MACROBUS via a BR line (not NPR).

4.2.2 Data-Transfer Lines

In a data transfer, the bus master controls the transfer of a word or byte of data to or from a slave device.

Address Lines

Eighteen physical address lines are provided on the MACROBUS. These are used by the master device to specify a memory or device register address. The format of the Address lines is:



Lines A17:A01 specify one of 128K unique addresses, each of which is assumed to be associated with a 16-bit word of data. In byte operations, A00 specifies the byte being referenced: A00 reset to ZERO references the less-significant (even) byte, and A00 set to ONE references the more-significant (odd) byte. Complete words are addressed only on even byte boundaries (i.e., A00 reset to ZERO).

CPU-generated addresses are a maximum of 16 bits in length, which permits direct access to only the first 32K word (64K byte) locations on the MACROBUS. For CPU addresses in the range 0 to 28K, the MCA places ZEROs on lines Al7:Al6. CPU addresses in the range 28K to 32K are converted by the MCA to physical MACROBUS addresses in the range 124K to 128K, which are reserved for nonmemory (I/O) device addresses.

Peripheral controllers assuming bus mastership must directly generate a full 18-bit (if required) physical add;ess.

Control Lines

The data-transfer control lines are:

- a. Mode control. Cl and CO are coded by the master device to control the slave in one of four possible data-transfer modes, as shown in Table 4-2.
- b. Master Synchronization. MSYN is used by the master device to indicate to the slave that address and control information is present on the MACROBUS.
- c. Slave Synchronization. SSYN is the slave device response to an action by the master (usually a response to MSYN).



Table 4-2. Data-Transfer Modes

| Name | Mnemonic | Cont Sign | als | Function |
|----------------------|----------|--------------|-----|--|
| Data Input | DATI | 0 | 0 | Word data transfer from slave to master. |
| Data Input, Pause | DATIP | 0 | 1 | Inhibits restore cycle in destructive readout devices. The pause flip-flop is set to inhibit the clear cycle on the following mandatory DATO or DATOB operation. |
| Data Output | . DATO | 1 | 0 | Word data transfer from master to slave. |
| Data Output, Byte | DATOB | 1 | 1 | Transfers a single byte from master to slave. Data are transmitted on D15:D07 when A00 is ONE, or on D07:D00 when A00 is ZERO. |

Data Lines

The 16 MACROBUS data lines used to transfer information have the format:

| 15 14 13 12 11 10 09 08 | 07 06 05 04 03 02 01 00 |
|-------------------------|-------------------------|
| More-Significant Byte | Less-Significant Byte |
| (bb0) | (Even) |

4.2.3 Other Lines

Three lines that may be used by all MACROBUS devices are:

- a. Initialization. INIT is asserted by the CPU when the START switch on the Macropanel is pressed or when a power-failure sequence occurs. In the latter case, INIT is asserted following the power-failure service routine when power goes down and again when power comes up. INIT is also generated by the Initialize program instruction.
- b. AC Low. ACLO starts the power-failure trap sequence and can also be used in peripheral devices to terminate operations prior to power loss. When ACLO is cleared, the power-up trap sequence in the CPU begins.

It is the programmer's responsibility to ensure that the trap vector memory location is loaded with the pointer to the power-failure routine, otherwise an undefined sequence results.

c. DC Low. DCLO is generated by the power supply or by the CPU power-failure/restart sequences. It remains off as long as all



dc voltages are within specified limits. If an out-of-voltage condition occurs, DCLO is asserted by the power supply. Devices such as core memories use DCLO to inhibit further operations. DCLO goes off before ACLO when power comes up and is asserted after ACLO when power goes down. The CPU power-failure trap sequence is initiated by ACLO, while DCLO is used by the MCA to assert INIT on the MACROBUS.

Two additional signals (PA and PB) are reserved by Cal Data for future use.

4.3 MACROBUS USE

When any device, including the CPU, requires MACROBUS access, it must first gain mastership of the MACROBUS. After that, the device can perform a data transfer or CPU interrupt. The operation to be performed determines the method of gaining bus mastership.

Appendix C is a training aid that will assist in understanding the sequence of events required for MACROBUS operations.

4.3.1 Request Priority

When a device requests MACROBUS access, the handling of the request depends on the location of that device in the priority structure. The following factors determine the priority of the request:

- a. The CPU priority is set under program control to one of eight levels by bits 07:05 of the processor status register (PS). The CPU priority level inhibits recognition of MACROBUS requests on the same or lower levels.
- b. MACROBUS requests from I/O devices can be made on any one of five request lines: BR7:BR4 and NPR. Multiple requests for MACROBUS access are serviced on a priority basis as follows:
 - NPR has the highest priority and is granted by the MCA between any two MACROBUS cycles of instruction execution (except after a DATIP operation), independent of the CPU state.
 - 2. After NPR, BR7 has the next-highest priority, and BR4 has the lowest. BR lines above the current CPU priority level have next-highest priority. They are serviced only at the end of a nonprivilege program instruction execution cycle.
 - 3. The CPU has priority over all requests on BR lines of the same or lower priority and is automatically given mastership of the MACROBUS in this case. CPU priority is monitored in the MCA, which contains the priority control logic (the CPU makes no direct MACROBUS requests, as such).
- c. When multiple devices share the same NPR or BR line, the device electrically closest to the MCA has the highest priority. Any number of devices can be connected to a specific BR or NPR line, using a priority-chaining technique.



4.3.2 Data Transfers

A MACROBUS device can request a data transfer between itself and another MACROBUS device, independently of the CPU. This can be accomplished by making a bus request via either NPR or one of the BR lines, depending on the priority assigned to the device. Once mastership is established and the MACROBUS is clear of any previous operation, the device performs the data transfer by placing the address of a slave device on the Address lines and a control code on Cl and CO (refer to Table 4-2). If the operation is a data output, the device also places data on the Data lines; otherwise, it waits for data from the slave.

Data transfers are defined with reference to the master device (i.e., data input is from slave to master and data output is from master to slave).

All MACROBUS transfers are asynchronous and depend on interlocking of control signals. In every case, a signal from a slave device is generated in response to a signal from a master device, and the master signal is then dropped in response to the slave signal. Signals on the MACROBUS are asserted when low (except for the unidirectional BG lines).

4.3.3 CPU Interrupts

A MACROBUS device can request a CPU interrupt to initiate a programmed service routine. This can be done only via one of the BR lines. (A CPU interrupt sequence initiated via an NPR causes improper system operation.) Once bus mastership is established, the MACROBUS is clear of any previous operation and the CPU completes execution of the current instruction, the new bus master initiates an interrupt sequence by asserting INTR and placing an interrupt vector address on the Data lines.

A vector address is a memory location that contains the starting address of an interrupt service routine (a program counter value, PC) followed by a new PS value. The CPU responds by accessing the specified PC and PS words, and pushing the current PC and PS values into a control stack in memory. When done, the interrupt routine returns control to the interrupted program by popping the stored PC and PS values from the control stack into the CPU.



SECTION 5 PROGRAMMING

5.1 GENERAL

Peripheral devices attached to the MACROBUS are controlled by programmed instructions. This is done by loading and reading registers in the devices, which are addressed by a unified addressing structure. This structure treats all accessible data, status and control registers in each MACROBUS peripheral as memory locations. The direct addressing of these registers eliminates any necessity for a special set of I/O instructions and, at the same time, increases programming flexibility by permitting the use of any memory-reference instruction to access peripheral device registers.

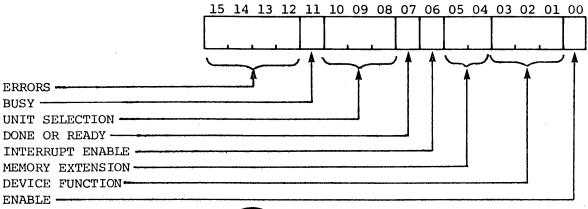
Examples showing the advantages of this concept include direct comparison of device register data with a program-determined value, direct transfer of device register data to or from memory or another peripheral device, performance of arithmetic operations on a peripheral device register, etc.

5.2 PERIPHERAL-DEVICE REGISTERS

Since registers in MACROBUS devices are assigned MACROBUS addresses identical to memory locations (Appendix B), all program instructions that address memory locations are also I/O instructions. Control functions are assigned a register address, and the individual bits within that "register" govern the occurrence of control operations. Status conditions are also handled by the assignment of bits within a device register, and the status is checked with an appropriate I/O instruction. Any number of additional registers can be provided for special functions. Figure 5-1 illustrates the organization of typical peripheral device control logic taking advantage of multiple registers.

Each peripheral has at least one data register for the temporary storage of I/O data. The number and type of such registers depends on the specific peripheral.

Each peripheral also has one or more control/status registers. Although there are some specific and significant differences according to the type of peripheral, the general format of such a register is:





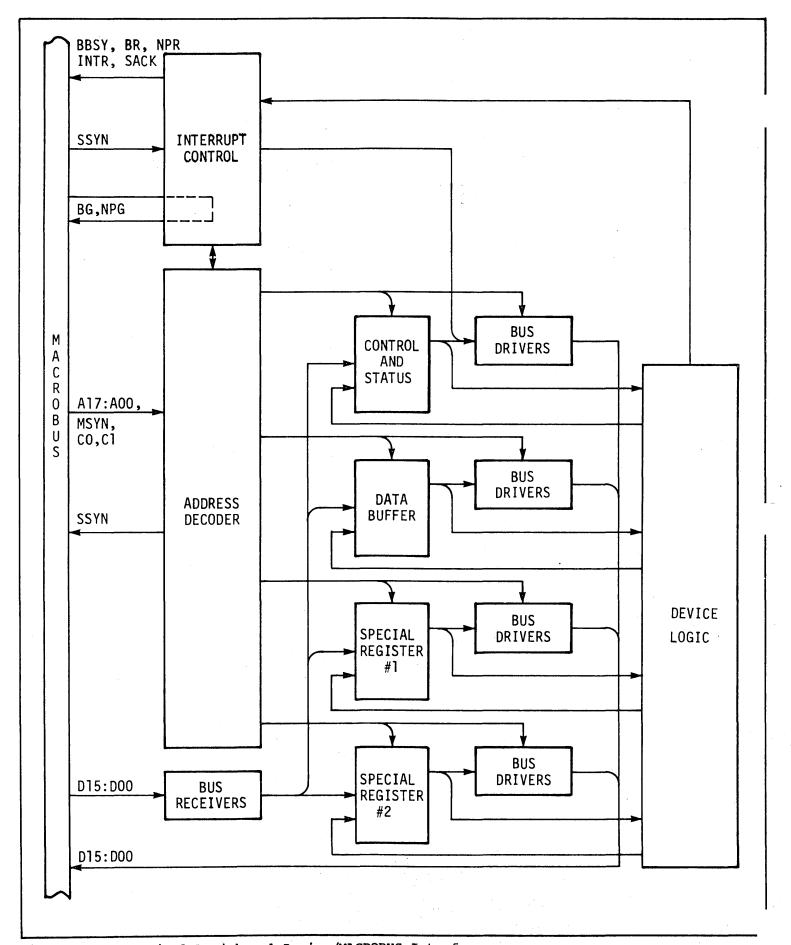


Figure 5-1. Typical Peripheral Device/MACROBUS Interface



When set, error bits 15:12 indicate specific errors defined for the particular peripheral. Bit 15 is often used as the inclusive-OR of all other error bits in use. In the case of other error bits, it is usual to have an individual bit associated with each specific error. An error can either be "fatal", which causes an immediate CPU interrupt if interrupt-enable bit 06 is set, or "warning", which does not immediately cause an interrupt.

Busy bit 11 indicates that the peripheral device is in use.

Unit-selection bits 10:08 select the particular unit of a multiunit peripheral (e.g., a magnetic tape unit).

Done/ready bit 07, set or cleared by hardware, indicates the completion of a peripheral task or the availability of the device for I/O operations.

The independently programmable interrupt-enable bit 06, when set, causes an interrupt in case of an error or special function.

Memory-extension bits 05 and 04 allow the device to use a total of 18 bits to specify an address on the MACROBUS.

Device-function bits 03:01 specify the particular function to be performed by the peripheral (e.g., instructing a card reader to read one card).

Master enable bit 00, when set, enables the addressed peripheral device to perform normal functions.

5.3 INTERRUPTS

Provided that interrupt-enable bit 06 is set, setting done/ready bit 07 or one of the error bits (15:12) causes issuance of an interrupt to the CPU. This also occurs if bit 07 or one of bits 15:12 is set when bit 06 comes on.

If the requesting device has a priority higher than that of the CPU, the MACROBUS access request is granted, thus initiating the interrupt sequence. This pushes the current PC and PS into the CPU control stack, and loads new PC and PS values from the interrupt vector locations, which constitute a pair of memory locations specific to the particular peripheral.

Unique device interrupt vectors eliminate the need for device polling. The priority for servicing an interrupt is under program control and can be different from the priority of the interrupt itself.

The Return from Interrupt (RTI) program instruction reverses the interrupt sequence, popping two words from the control stack into PC and PS.

SECTION 6 MAINTENANCE

6.1 GENERAL

This section describes preventive and corrective maintenance procedures that apply to the MCA. In general, corrective maintenance is limited to the isolation of a fault to the MCA board followed by replacement of the board. Repair should be conducted at the factory or by an authorized Cal Data representative.

6.2 PREVENTIVE MAINTENANCE

The MCA board is a reliable solid-state device designed to perform continuously for many years. Preventive maintenance consists of performing the following tasks every six months:

- a. Inspect the board for damaged wires or components, or other obvious defects.
- b. Using a low pressure source of air (75 psi one foot from the board or 5 kg/cm^2 30 cm from the board), blow off accumulated dust and foreign matter.

Another aspect of preventive maintenance is proper handling of the board. The following points should be observed:

- a. Always be sure that system power is OFF before installing or removing any board.
- b. Install each board with the component side toward the front of the chassis. Check each board for proper orientation before attempting to install it. Because the connectors are keyed, excessive force applied to a reversed board can result in connector damage. Make sure that the board is completely and evenly seated.
- c. Insert and remove each board slowly and carefully so that it does not make contact with adjacent boards.
- d. Never use components as finger grips; use the grip areas at the corners of the board.
- e. To prevent oxides from forming on the gold plating, do not touch connectors.

6.3 CORRECTIVE MAINTENANCE

Repair or adjustment of the MCA board in the field is not recommended. If a malfunction is detected, replace the board with a spare known to be operating properly and return the malfunctioning one for repair to California Data Processors or an authorized representative.

APPENDIX A CONNECTOR PIN ASSIGNMENTS

| Name | Signal | Pin | Pin | Signal | Name | Table | | |
|--|-------------|-----|-----|------------|---------------------------|--------|--|--|
| Initialize | BUS INIT-L | Al | A2 | +5V | +5 Vdc | 7 7 | | |
| Interrupt | BUS INTR-L | B1 | B2 | GND | Ground | 1 - | | |
| Data 00 | BUS DOO-L | Cl | C2 | GND | Ground | 1. | | |
| Data 02 | BUS DO2-L | Dl | D2 | BUS DO1-L | Data 01 | 0 | | |
| Data 04 | BUS DO4-L | El | E2 | BUS D03-L | Data 03 | Conne | | |
| Data 06 | BUS D06-L. | Fl | F2 | BUS DO5-L | Data 05 | ne | | |
| Data 08 | BUS D08-L | Gl. | H2 | BUS DO7-L | Data 07 | ctor | | |
| Data 10 | BUS D10-L | Jl | J2 | BUS D09-L | Data 09 | H | | |
| Data 12 | BUS D12-L | Кl | K2 | BUS D11-L | Data 11 | A | | |
| Data 14 | BUS D14-L | L1 | L2 | BUS D13-L | ·Data 13 | ъ | | |
| Parity Bit Low | * BUS PA-L | MI | M2 | BUS D15-L | Data 15 | Pin | | |
| Ground | GND | Nl | N2 | * BUS PB-L | Parity Bit High | As | | |
| Ground | GND | Pl | P2 | BUS BBSY-L | Bus Busy | 1 70 | | |
| Ground | GND | Rl | R2 | BUS SACK-L | Selection Acknowledgement | ig | | |
| Ground | GND | Sl | S2 | BUS NPR-L | Nonprocessor Request | 1 3 | | |
| Ground | GND | Tl | T2 | BUS BR7-L | Bus Request 7 | ignmen | | |
| Nonprocessor Grant | * BUS NPG-H | Ul | U2 | BUS BR6-L | Bus Request 6 | ts | | |
| Bus Grant 7 | * BUS BG7-H | V1 | V2 | GND | Ground | | | |
| | _ | | 11 | | | 13 | | |
| * These signals are assigned on the backplane but are not used on this assembly. | | | | | | | | |



^{*} These signals are assigned on the backplane but are not used on this assembly.

| 「ツ | |
|----|--|

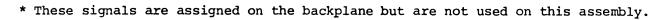
| Name | Signal | Pin | Pin | Signal | Name |
|------------------------|-------------|-----|-----|-------------|---------------|
| Bus Grant 6 | * BUS BG6-H | Al | A2 | +5V | +5 Vdc |
| Bus Grant 5 | * BUS BG5-H | Bl | B2 | GND | Ground |
| Bus Request 5 | * BUS BR5-L | Cl | C2 | GND | Ground |
| Ground | GND | D1 | D2 | * BUS BR4-L | Bus Request 4 |
| Ground | GND | E1 | E2 | * BUS BG4-H | Bus Grant 4 |
| AC Low | BUS ACLO-L | Fl | F2 | BUS DCLO-L | DC Low |
| Address 01 | BUS A01-L | н1 | Н2 | BUS A00-L | Address 00 |
| Address 03 | BUS A03-L | Jl | J2 | BUS A02-L | Address 02 |
| Address 05 | BUS A05-L | Кl | К2 | BUS A04-L | Address 04 |
| Address 07 | BUS A07-L | Ll | L2 | BUS A06-L | Address 06 |
| Address 09 | BUS A09-L | Ml | M2 | BUS A08-L | Address 08 |
| Address 11 | BUS All-L | Nl | N2 | BUS AlO-L | Address 10 |
| Address 13 | BUS A13-L | P1 | P2 | BUS Al2-L | Address 12 |
| Address 15 | BUS A15-L | Rl | R2 | BUS Al4-L | Address 14 |
| Address 17 | BUS A17-L | S1 | S2 | BUS Al6-L | Address 16 |
| Ground | GND | Tl | Т2 | BUS Cl-L | Control 1 |
| Slave Synchronization | BUS SSYN-L | Ul | U2 | BUS CO-L | Control 0 |
| Master Synchronization | BUS MSYN-L | V1 | V2 | GND | Ground |
| | | | | | |

^{*} These signals are assigned on the backplane but are not used on this assembly.



| Name | Signal | Pin | Pin | Signal | Name |
|----------|---------|-----|-----|---------|----------|
| M Bus 00 | MB000-L | Al | A2 | +5V | +5 Vdc |
| M Bus 01 | MB001-L | B2 | B2 | -15V | -15 Vdc |
| M Bus 02 | MB002-L | C1 | C2 | GND | Ground |
| M Bus 03 | MB003-L | Dl | D2 | MB004-L | M Bus 04 |
| M Bus 05 | MB005-L | El | E2 | MB006-L | M Bus 06 |
| M Bus 07 | MB007-L | Fl | F2 | MB008-L | M Bus 08 |
| M Bus 09 | MB009-L | H1 | Н2 | MB010-L | M Bus 10 |
| M Bus 11 | MBOll-L | J1 | J2 | MB012-L | M Bus 12 |
| M Bus 13 | MB013-L | K1 | К2 | MB014-L | M Bus 14 |
| M Bus 15 | MB015-L | Ll | L2 | AB000-H | A Bus 00 |
| A Bus 01 | AB001-H | Ml | M2 | AB002-H | A Bus 02 |
| A Bus 03 | AB003-H | Nl | N2 | AB004-H | A Bus 04 |
| A Bus 05 | AB005-H | P1 | P2 | AB006-H | A Bus 06 |
| A Bus 07 | AB007-H | Rl | R2 | AB008-H | A Bus 08 |
| A Bus 09 | AB009-H | Sl | S2 | AB010-H | A Bus 10 |
| Ground | GND | Tl | т2 | AB011-H | A Bus 11 |
| A Bus 13 | AB013-H | וט | U2 | AB012-H | A Bus 12 |
| A Bus 15 | AB015-H | Vl | V2 | AB014-H | A Bus 14 |

| Name Signal Pin Pin Signal | Name |
|---|-------------------------|
| Power Failure Interrupt PFINT-H Al A2 +5V | +5 Vđc |
| Halt Interrupt * HLINT-H B1 B2 *-15V | -15 Vdc |
| Data Switch 16 | Ground |
| Data Switch 17 * DS17-H D1 D2 * LTCL-L 1 | Line-Frequency Clock |
| Virtual Address * VIRTAD-H E1 E2 PBBSY-L | Processor Bus Busy |
| Control Count 00 | Panel Halt |
| Control Count 01 | Microstatus Register 15 |
| Control Count 02 | Reset |
| Control Count 03 | Bus Grant 7 In |
| Control Count 04 | Bus Grant 7 Out |
| Control Count 05 | Bus Grant 6 In |
| Control Count 06 | Bus Grant 6 Out |
| Control Count 07 | Bus Grant 5 In |
| Control Count 08 | Bus Grant 5 Out |
| Control Count 09 * CC009-L S1 S2 * BUS BG4-IN 1 | Bus Grant 4 In |
| Ground GND T1 T2 * BUS BG4-OUT T | Bus Grant 4 Out |
| Control Count 10 | Nonprocessor Grant In |
| Control Count 11 | Nonprocessor Grant Out |





| Name | Signal | Pin | Pin | Signal | Name |
|-------------------|-----------|-----|-----|-----------|-------------------|
| Control Memory 00 | СМ000-Н | Al | A2 | +5V | +5 Vdc |
| Control Memory 01 | CM001-H | B1 | В2 | * -15V | -15 Vdc |
| Control Memory 02 | CM002-H | C1 | C2 | GND | Ground |
| Control Memory 03 | CM003-H | D1 | D2 | CM004-H | Control Memory 04 |
| Control Memory 05 | CM005-H | El | E2 | CM006-H | Control Memory 06 |
| Control Memory 07 | CM007-H | F1 | F2 | * EMINH-L | Emulate Inhibit |
| Control Memory 09 | CM009-H | Hl | Н2 | CM008-H | Control Memory 08 |
| Control Memory 11 | CM011-H | Jl | J2 | CM010-H | Control Memory 10 |
| Decode Address 00 | DADOO-H | Kl | K2 | CM012-H | Control Memory 12 |
| Control Memory 13 | CM013-H | Ll | L2 | CM014-H | Control Memory 14 |
| Control Memory 15 | CM015-H | Ml | M2 | DAD01-H | Decode Address 01 |
| Control Memory 17 | * CM017-H | Nl | N2 | * CM016-H | Control Memory 16 |
| Control Memory 19 | * CM019-H | P1 | P2 | * CM018-H | Control Memory 18 |
| Switch Register 0 | * SRO-L | Rl | R2 | * CM020-H | Control Memory 20 |
| Control Memory 21 | * CM021-H | S1 | S2 | * CM022-H | Control Memory 22 |
| Ground | GND | Tl | Т2 | CM024-H | Control Memory 24 |
| Control Memory 23 | * CM023-H | Ul | U2 | CM026-H | Control Memory 26 |
| Control Memory 25 | CM025-H | V1 | V2 | CM027-H | Control Memory 27 |

^{*} These signals are assigned on the backplane but are not used on this assembly.

| Table . |
|-----------------|
| A-6. |
| Connector I |
| |
| Pin |
| Pin Assignments |

| Name | Signal | Pin | Pin | Signal | Name | |
|--------------------|-----------|-----|-----|-----------|---------------------------------|-----|
| Control Memory 28 | СМ028-Н | Al | A2 | +5V | +5 Vdc |]: |
| Control Memory 29 | * CM029-H | Bl | B2 | * -15V | -15 Vdc | |
| Control Memory 31 | * CM031-H | Cl | C2 | GND | Ground | ' |
| Control Memory 31 | * CM030-H | Dl | D2 | * CM032-H | Control Memory 32 | |
| Control Memory 33 | * CM033-H | El | E2 | * CM034-H | Control Memory 34 | |
| Control Memory 35 | * CM035-H | F1 | F2 | DAD02-H | Decode Address 02 | |
| Control Memory 37 | * CM037-H | Hl | Н2 | * CM036-H | Control Memory 36 | |
| Control Memory 39 | * CM039-H | Jl | J2 | * CM038-H | Control Memory 38 | 1 |
| Instruction Repeat | IRPTE-L | Kl | K2 | * CM040-H | Control Memory 40 | 1 |
| Control Memory 41 | * CM041-H | Ll | L2 | * CMO42-H | Control Memory 42 | 1: |
| Control Memory 43 | * CMO43-H | Ml | М2 | * CPEN-L | Control Panel Enable | |
| Control Memory 45 | * CM045-H | Nl | N2 | * CMO44-H | Control Memory 44 | |
| Control Memory 47 | * CM047-H | Pl | P2 | * CM046-H | Control Memory 46 | |
| Decode Address 03 | DAD03-H | Rl | R2 | * ACMSL-L | Alterable Control Memory Select | ا ا |
| | Reserved | sı | S2 | * AUXRM-L | Auxiliary ROM Select | |
| Ground | GND | Tl | Т2 | IRINH-L | Instruction Inhibit | |
| | Reserved | Ul | U2 | IWAIT-L | Instruction Wait | 18 |
| System Clock | SYSCK-L | Vl | V2 | GND | Ground | |
| | i . | | 1 | 1 | 1 | 1 |

^{*}These signals are assigned on the backplane but are not used on this assembly

| Name | Signal | Pin | Pin | Signal | Name |
|------------------------------------|------------|-----|-----|-----------|----------------------------------|
| Skip | * SKIPP-L | 1A | 1B | * EMA00-H | Emulate Address 00 |
| AR Write Enable | ARWEN-L | 2A | 2B | * EMAO1-H | Emulate Address 01 |
| Stack Limit Write Enable | SLWEN-L | 3A | 3B | * EMA02-H | Emulate Address 02 |
| Slave Synchronization Error | SSYER-H | 4A | 4B | * EMA03-H | Emulate Address 03 |
| Double Slave Synchronization Error | DSYER-H | 5A | 5B | * EMAO4-H | Emulate Address 04 |
| Load Special Function | LDSPF-H | 6A | 6B | * EMA05-H | Emulate Address 05 |
| Fatal Interrupt | FINTP-L | 7A | 7B | * EMA06-H | Emulate Address 06 |
| Special Function | * SPFNC-H | 8A | 8B | * EMA07-H | Emulate Address 07 |
| Panel Halt | * HALTP-L | 9A | 9в | Reserved | |
| | Reserved | 10A | 10B | PSSEL-L | Program Status Select |
| Carry | * CARRY-H | 11A | 11B | Reserved | |
| | Reserved | 12A | 12B | Reserved | · |
| Address Error | ADERR-H | 13A | 13B | Reserved | |
| Program Status 03 | PS003-L | 14A | 14B | Reserved | |
| | Reserved | 15A | 15B | * XD007-L | Inhibit Destination File 0 to 7 |
| | Reserved | 16A | 16B | * XD815-L | Inhibit Destination File 8 to 15 |
| | Reserved | 17A | 17B | * XB815-L | Inhibit B-Field File 8 to 15 |
| Control Count Write Enable | * CCWEN-H | 18A | 18B | * XB007-L | Inhibit B-Field File 0 to 7 |
| Static Condition | * STATIC-L | 19A | 19B | * LITRL-L | Literal |
| Master Synchronization | MSYN-H | 20A | 20B | * PLUS1-L | Plus 1 |
| Special Function 04 | * SPFO4-L | 21A | 21B | * PSWEN-L | Processor Status Write Enable |
| B Bus Inhibit | * BBINH-L | 22A | 22B | * IRWEN-L | IR Write Enable |
| B Bus 01 | * BB001-H | 23A | 23B | * BB000-H | B Bus 00 |
| B Bus 03 | * BB003-H | 24A | 24B | * BB002-H | B Bus 02 |
| B Bus 05 | * BB005-H | 25A | 25B | * BB004-H | B Bus 04 |
| B Bus 07 | * BB007-H | 26A | 26B | * BB006-H | B Bus 06 |
| B Bus 09 | * BB009-H | 27A | 27B | * BB008-H | B Bus 08 |
| B Bus 11 | * BB011-H | 28A | 28B | * BB010-H | B Bus 10 |
| B Bus 13 | * BB013-H | 29A | 29B | * BB012-H | B Bus 12 |
| B Bus 15 | * BB015-H | 30A | 30B | * BB014-H | B Bus 14 |

^{*}These signals are assigned on the small processor interconnection board but are not used on this assembly.

| Name | Signal | Pin | Pin | Signal | Name |
|--------------------------------|------------|-----|-----|------------|--------------------------------|
| Load CC Register | * LOADC-L | 1A | 18 | Reserved | |
| Bus Request | BREQ-H | 2A | 2B | * MINTP-L | Microinterrupt |
| Bus Grant | BGRNT-L | 3A | 3B | * BYTDA-L | Byte Data |
| Bus Grant Enable | BGEN-H | 4A | 4B | Reserved | |
| Memory Management Inhibit | MMINH-L | 5A | 5B | * MARLD-H | Management Address Load |
| Data Inhibit | DAINH-L | .6A | 6B | * CCCEN-H | CC Count Enable |
| Special Function 7 | SPF07-L | 7A | 7B | * SPRlA-L | Special Register 1A |
| Special Function 5 | SPF05-L | 8A | 8B | * SPR19-L | Special Register 19 |
| Special Function 6 | SPF06-L | 9A | 9B | * SPRlB-L | Special Register 1B |
| Special Function Decode | * SPFNC-H | 10A | 10B | * MLTPY-L | Multiply |
| Inhibit B Field | INHBF-L | 11A | 11B | ENSPF-H | Enable Special Function |
| Emulate | EMLAT-H | 12A | 12B | * CR008-H | Microcommand Register 08 |
| Power Failure | PFAIL-L | 13A | 13B | Reserved | |
| AU Carry In | * AUCIN-L | 14A | 14B | Reserved | |
| Write | WRITE-L | 15A | 15B | * FILE6-H | File 6 |
| IR Read | IRERD-H | 16A | 16B | * XA815-L | Inhibit A-Field File 8 to 15 |
| Interrupt | INTR-H | 17A | 17B | * XA007-L | Inhibit A-Field File 0 to 7 |
| Memory Management CO | MMCO-L | 18A | 18B | * RSTRA-L | Restore A |
| Memory Management Cl | MMCl-L | 19A | 19B | * YELLW-L | Yellow |
| Microcommand Register 07 | * CR007-H | 20A | 20B | * BYTMD-L | Byte Mode |
| Stack Limit Interrupt | * SLINT-H | 21A | 21B | * MS006-H | Microstatus Register 06 |
| DR Write Enable | DRWEN-L | 22A | 22B | RRWEN-L | RR Write Enable |
| Emulate Instruction Address 01 | * EIA001-H | 23A | 23B | * EIA000-H | Emulate Instruction Address 00 |
| Emulate Instruction Address 03 | * EIA003-H | 24A | 24B | * EIA002-H | Emulate Instruction Address 02 |
| Emulate Instruction Address 05 | * EIA005-H | 25A | 25B | * EIA004-H | Emulate Instruction Address 04 |
| Emulate Instruction Address 07 | * EIA007-H | 26A | 26B | * EIA006-H | Emulate Instruction Address 06 |
| Emulate Instruction Address 09 | * EIA009-H | 27A | 27B | * EIA008-H | Emulate Instruction Address 08 |
| Emulate Instruction Address 11 | EIAO11-H | 28A | 28B | EIAO10-H | Emulate Instruction Address 10 |
| Emulate Instruction Address 13 | * EIA013-H | 29A | 29B | * EIA012-H | Emulate Instruction Address 12 |
| Emulate Instruction Address 15 | * EIA015-H | 30A | 30B | * EIA014-H | Emulate Instruction Address 14 |
| | | | | | |

^{*}These signals are assigned on the small processor interconnection board but are not used on this assembly.

APPENDIX B ASSIGNED ADDRESSES

| | T | r | |
|---------|----------------------------|--------------|------------------------------|
| Octal | | Octal | |
| Address | Use | Address | Use |
| | | | |
| 777776 | Processor status, PS | 777560 | Teleprinter IN CSR |
| 777774 | Stack-limit register | 777556 | Punch DBR |
| 777772 | | 777554 | Punch CSR |
| 777717 | | 777552 | Reader DBR |
| 777716 | | 777550 | Reader CSR |
| 777715 | 1 | 777546 | Line-Frequency Clock |
| 777714 | Other CPU registers | 777540 | Macropanel control switches |
| 777713 | _ | 777516 | Line printer DBR |
| 777712 | | 777514 | Line printer CSR |
| 777711 |] | 777512 | Line printer XX |
| 777710 | , | 777510 | Line printer XX |
| 777707 | FR7 register | 777166 | |
| 777706 | FR6 register | 777164 | Card reader |
| 777705 | FR5 register | 777162 | July 100001 |
| 777704 | FR4 register | 777160 | |
| 777703 | FR3 register | 776676 | Multiple teleprinter (first |
| 777702 | FR2 register | to | unit is at 776500) |
| 777701 | FR1 register | 776500 | Wife 13 de 770300) |
| 777700 | FRO register | 772356 | EPAR #7 |
| 777656 | UPAR #7 | 772354 | EPAR #6 |
| 777654 | UPAR #6 | 772354 | EPAR #5 |
| 777652 | UPAR #5 | 772352 | EPAR #3 MMU* |
| 777650 | UPAR #4 MMU* | 772336 | 1 |
| 777646 | UPAR #3 (USER) | 772344 | EPAR #3 (EXECUTIVE) EPAR #2 |
| 777644 | UPAR #2 | 772344 | EPAR #2 EPAR #1 |
| 777642 | UPAR #1 | 772342 | EPAR #1 EPAR #0 |
| 777640 | UPAR #0 | 772340 | EPDR #7 |
| 777616 | UPDR #7 | 772316 | 1 |
| 777614 | I I | 772314 | EPDR #6 |
| 777612 | UPDR #6 | 772312 | EPDR #5 |
| 1 | UPDR #5 | | EPDR #4 MMU* |
| 777610 | UPDR #4 MMU* | 772306 | EPDR #3 (EXECUTIVE) |
| 777606 | UPDR #3 (USER) | 772304 | EPDR #2 |
| 777604 | UPDR #2 | 772302 | EPDR #1 |
| 777602 | UPDR #1 | 772300 | EPDR #0 / |
| 777600 | UPDR #0 / | 760000 | |
| 777576 | MMU status register #2 | 754005 | |
| 777572 | MMU status register #0 | 764000 | User addresses start here |
| 777570 | Macropanel switch register | | and run upward to 767750 |
| 777566 | Teleprinter OUT DBR | | <u> </u> |
| 777564 | Teleprinter OUT CSR | | |
| 777562 | Teleprinter IN DBR | *Cannot be a | ccessed by a MACROBUS device |

APPENDIX C MACROBUS OPERATIONS

C.1 MACROBUS ACCESS

The method of gaining mastership of the MACROBUS differs depending on whether a data transfer or CPU interrupt is to take place. Both methods are described and illustrated below.

C.1.1 Data Transfers

Data transfers are initiated by a request on either NPR or a BR line. A signal sequence selects the device as next bus master when the device has the highest priority. The appropriate data-transfer sequence then takes place.

Although BR lines can be used to initiate data transfers, the NPR is usually used. The MACROBUS access sequence described in the following paragraphs is for NPR only. The same operations occur using a BR line, the difference being recognition of the NPR between any two MACROBUS cycles versus recognition of a BR only after execution of a nonprivileged program instruction. The sequence of operations performed for selection of the next NPR bus master is given below and in Figure C-1:

- 1. The requesting MACROBUS device asynchronously asserts NPR.
- Provided the SACK line on the MACROBUS is clear and no BR operation is in progress, the MACROBUS priority control logic immediately asserts NPG.
- 3. The first requesting device on the NPG line asserts SACK, blocks NPG from all following devices and clears NPR.
- 4. When SACK is received, priority control logic clears NPG. (If SACK is not received within 10 microseconds, a time-out occurs and NPG is cleared automatically by the MCA. This does not result in a time-out error trap.) The requesting device, now selected as next bus master, continues to assert SACK until BBSY and MSYN are cleared by the current bus master on completion of its operation, and SSYN is cleared by the current slave. Assertion of SACK by the selected device inhibits recognition of any other NPR by the priority control logic.
- 5. When BBSY, MSYN and SSYN are cleared after the previous MACROBUS operation, the selected device asserts BBSY, clears SACK and initiates a data transfer sequence.

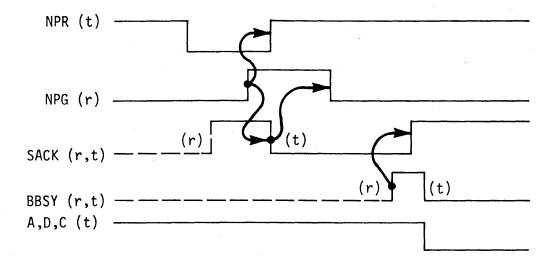
An NPR device can make multiple data transfers after gaining control of the MACROBUS by leaving BBSY asserted and repeating the appropriate data transfer sequence. This occupies the MACROBUS during the entire interval.

C.1.2 CPU Interrupts

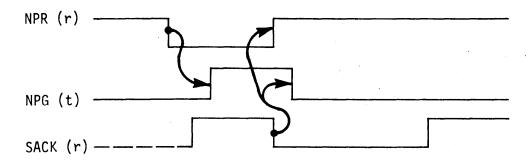
MACROBUS devices interrupt the CPU through the MCA via one of the BR lines. NPR cannot be used for a CPU interrupt, since indeterminate results occur. The program sequence being performed by the CPU is interrupted and an interrupt service routine is initiated. After the device



SIGNALS AT REQUESTING DEVICE



SIGNALS AT MCA PRIORITY CONTROL



(t)=Transmitted
(r)=Received

Figure C-1. NPR MACROBUS Access

request has been serviced, the program returns the CPU to its former sequence.

The sequence by which the device gains MACROBUS mastership is similar to that previously described for an NPR. Figure C-2 illustrates the sequence:

- 1. The requesting device asynchronously asserts the assigned BR.
- 2. BRs received by priority control logic are compared with the CPU priority and with each other. At the end of an instruction-execution cycle, and provided the SACK line is clear and no NPR is present, the priority control logic asserts a BG signal on the highest grant line above the current CPU priority level.
- 3. The first requesting device on the asserted BG line asserts SACK, blocks BG from all following devices and clears BR. (If SACK is not received within 10 microseconds, a time-out occurs and BG is cleared automatically by the MCA. This does not result in a time-out error trap.)
- 4. When SACK is received, the priority control logic clears BG. The requesting device, now selected as the next bus master, continues to assert SACK until BBSY and MSYN are cleared by the current bus master on completion of its operation and SSYN is cleared by the slave.
- 5. When BBSY, MSYN and SSYN are cleared, the selected device asserts BBSY, then asserts INTR and places an interrupt vector address on the Data lines at the same time that it clears SACK.
 - A vector address is a memory location that contains the starting address of an interrupt service routine. At the vector address plus two is a word that becomes the new processor status (PS contents).
- 6. The CPU receives INTR and asserts SSYN when it receives the vector address. A 75-nanosecond delay is provided at the CPU to ensure deskewing of the data lines.
- 7. The interrupting device clears INTR, BBSY and the Data lines when SSYN is received from the CPU, thus releasing the MACROBUS.
- 8. Assuming that an NPR device has not gained bus mastership, the CPU asserts BBSY and proceeds to execute access to the interrupt vector in memory, initiating the internal interrupt sequence.

The CPU pushes the current PC and PS values into a control stack in memory. New PC and PS values are taken from the vector locations specified by the interrupting device.

An interrupt routine can be interrupted in the same manner by a higherpriority interrupt any time after the current PC and PS values have been pushed into the stack. Nesting of priority interrupts can continue to any level within the limits of memory space available for the stack.

When done, the interrupt routine returns control to the interrupted program by popping stored PC and PS values from the control stack and using these as new current values in the CPU.



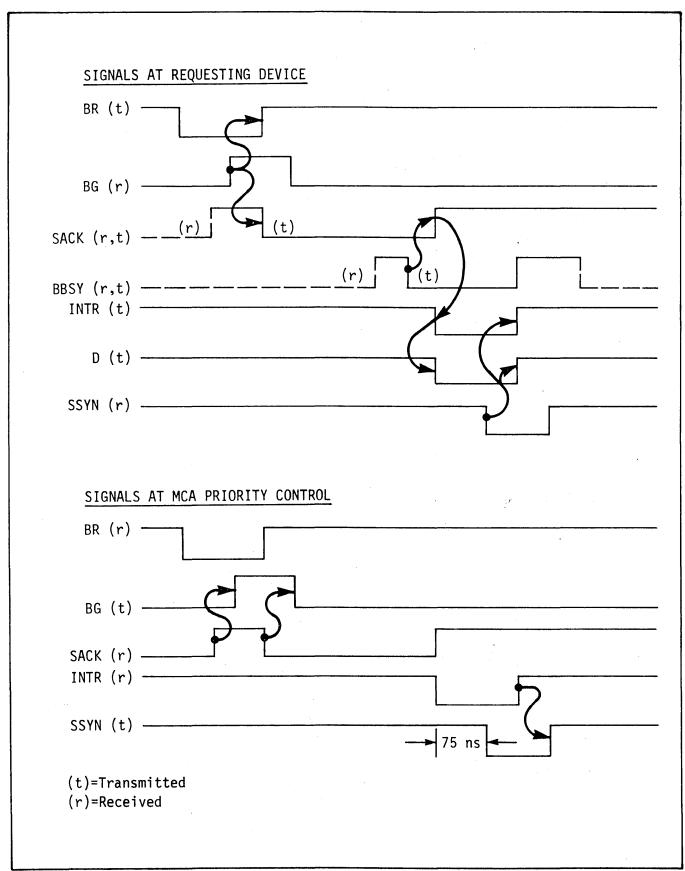


Figure C-2. CPU Interrupt MACROBUS Access



The device can precede the interrupt with one or more data transfers, if required. To do this, it must leave SACK asserted until INTR is asserted, otherwise the request may not be recognized by the MCA. The CPU program is delayed during the period that SACK remains asserted by the device, and NPRs are not serviced during this period.

C.2 DATA-TRANSFER SEQUENCES

Once a device has mastership of the MACROBUS, it can transfer data to (output) or from (input) a slave.

All MACROBUS transfers are asynchronous and depend on interlocking of control signals. To ensure reliable operation, the master must provide certain timing considerations:

- a. The master always delays MSYN 75 nanoseconds to ensure that it does not reach the slave device prior to valid data or addresses. MSYN is further delayed 75 nanoseconds to allow for address decoding by the slave device. These delays are always made by the master even when a slave transmits to a master (DATI or DATIP).
- b. A slave can place data or an address on the MACROBUS at the same time that it responds with SSYN or INTR, respectively. Because failure to receive SSYN from a slave locks the MACROBUS indefinitely, every master should incorporate a time-out circuit. Delay time is dependent on system requirements. Failure to receive SSYN within the time-out period should cause the master to clear all of its asserted signals and set an error flag in a status register. The CPU provides a 25 microsecond delay for this purpose.
- c. To ensure that there are no spurious selections, the master does not clear the Address or Mode Control lines until 75 nanoseconds after MSYN has been cleared.

C.2.1 Data Input

Timing of the DATI and DATIP transfers after a device has gained bus mastership is determined as follows and as shown in Figure C-3:

- 1. If SSYN from a previous MACROBUS transfer is still asserted, the master waits until SSYN is clear before asserting MSYN, although Address and Data are placed on the MACROBUS. The master then places the address of the slave device on the Address lines and the DATI or DATIP control code on Cl and CO. The master waits a minimum of 150 nanoseconds before asserting MSYN (75 nanoseconds for signal skew and 75 nanoseconds for slave address decoding).
- 2. The slave decodes the Address and Mode Control lines. When MSYN is received after address recognition, the slave accesses the specified data location for transfer of its contents to the master.
- 3. The slave places data, when available, on the Data lines and asserts SSYN. In a DATI operation where the slave is a destructive read-out device (such as core memory), it immediately enters a restore cycle. For DATIP, the slave sets a pause flip-flop and waits for modified data before performing the restore cycle.

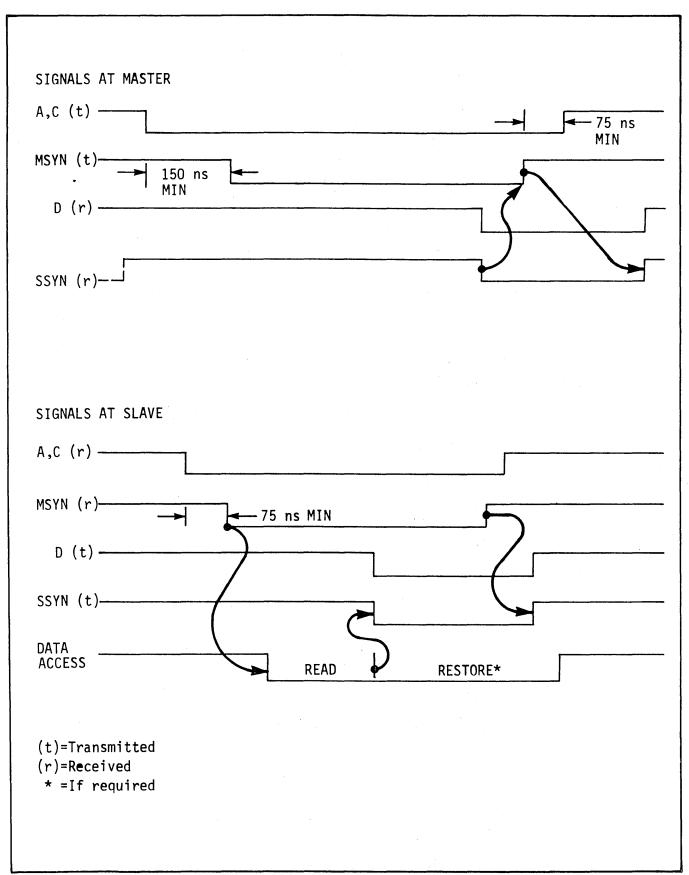


Figure C-3. DATI and DATIP Timing



C.2.2 Data Output

Timing of the DATO and DATOB transfers after a device has gained bus mastership is determined as follows and as shown in Figure C-4:

- 1. If SSYN from a previous MACROBUS transfer is still asserted, the master waits until SSYN is clear. The master then places the address of the slave device on the Address lines, data on the Data lines, and the DATO or DATOB control code on Cl and CO. The master waits a minimum of 150 nanoseconds before asserting MSYN.
- 2. The slave decodes the Address and Mode Control lines. When MSYN is received after address recognition, the slave asserts SSYN when the transmitted data have been received. The slave accepts a full word for DATO and a byte (specified by AOO) for DATOB.
- 3. The master receives SSYN and clears MSYN. (If the CPU is master and SSYN is not received with 25 microseconds, a time-out error trap occurs and MSYN is automatically cleared by the CPU.)
- 4. The master can modify the Address, Data and Mode control lines after a minimum delay of 75 nanoseconds from MSYN clear.
- 5. When MSYN is cleared, the slave clears SSYN.



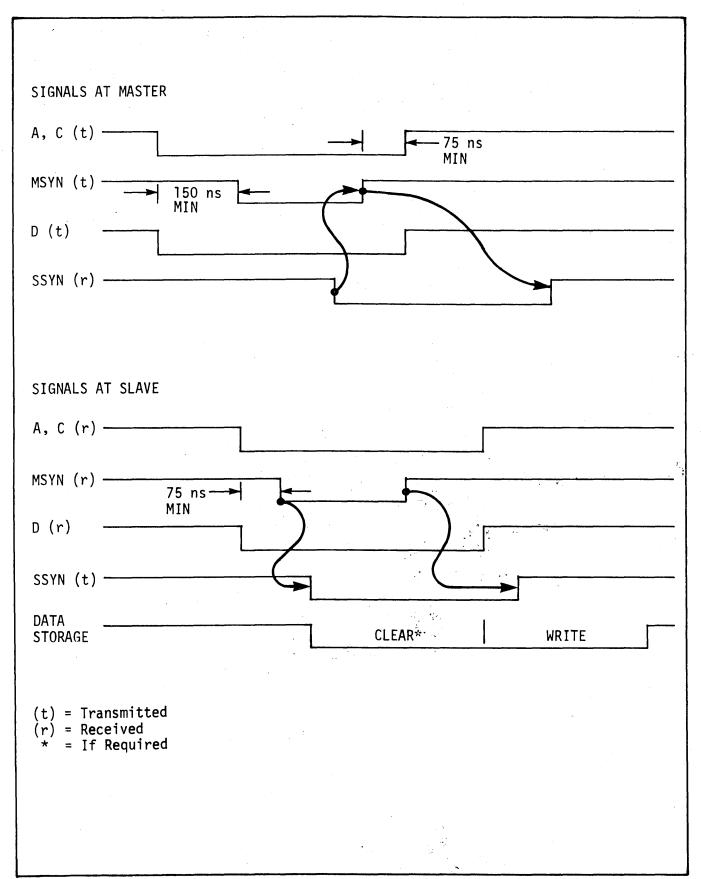


Figure C-4. DATO and DATOB Timing