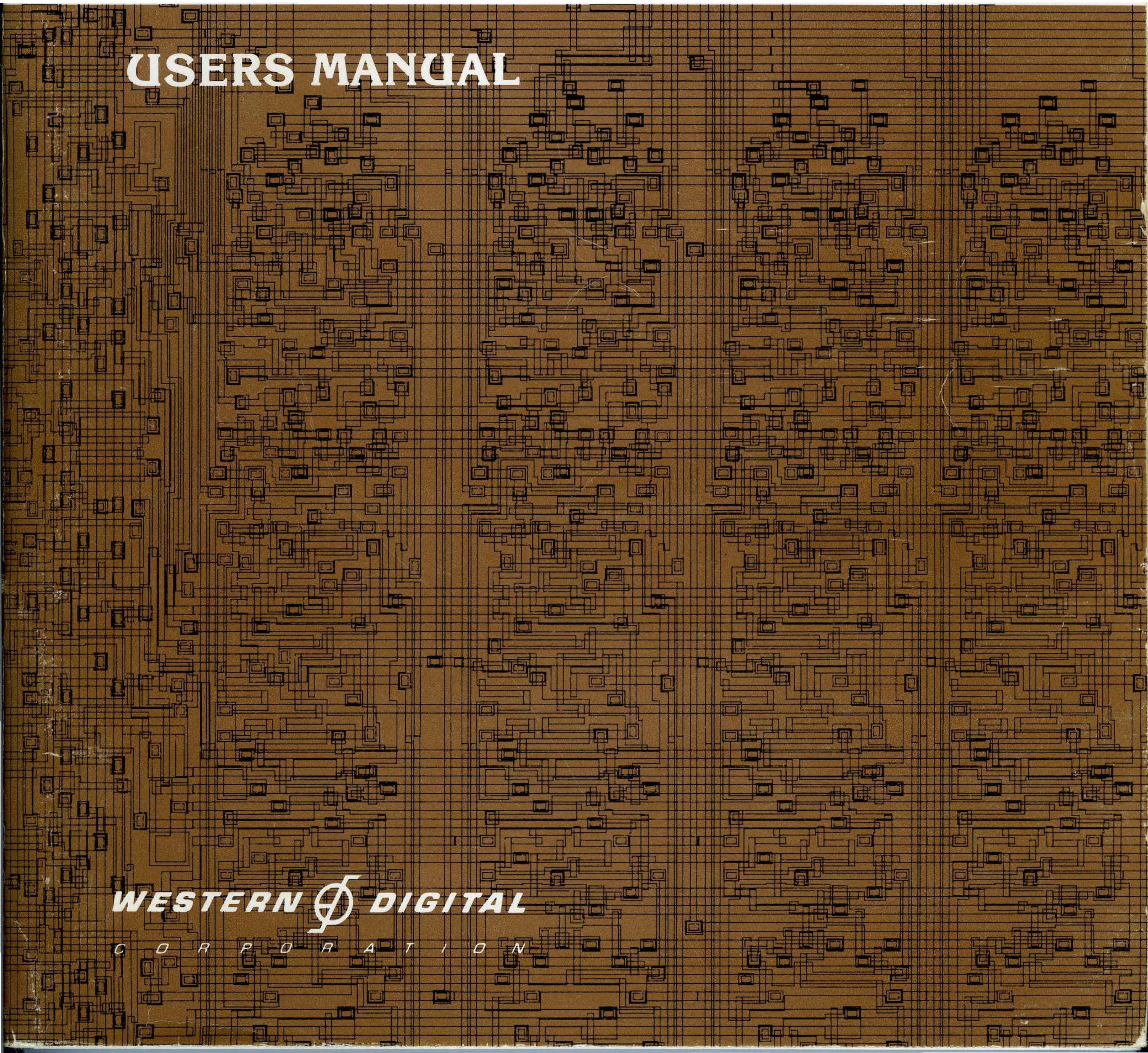
PRICE $10.00

WESTERN DIGITAL MCP-1600 MICROPROCESSOR SYSTEM

USERS MANUAL



WESTERN DIGITAL
CORPORATION

MCP-1600 MICROPROCESSOR USERS MANUAL

WESTERN DIGITAL

CORPORATION

3128 RED HILL AVENUE P.O. BOX 2180
NEWPORT BEACH, CALIFORNIA 92663
TELEPHONE: 714-557-3550
TWX: 910-595-1139

TABLE OF CONTENTS

Paragraph | Page

SECTION I, GENERAL SYSTEM CONFIGURATION

SECTION II, SYSTEM COMPONENTS

SECTION III, CPU OPERATIONS

SECTION IV, PROGRAMMABLE TRANSLATION ARRAY

SECTION V, MICROINSTRUCTIONS

SECTION VI, MICROINSTRUCTION BUS OPERATION

TABLE OF CONTENTS (con't.)

Paragraph Page

SECTION VI, (con't.)

SECTION VII, DATA ACCESS

SECTION VIII, ELECTRICAL SPECIFICATIONS

SECTION IX, PIN ASSIGNMENTS

ILLUSTRATIONS

TABLES

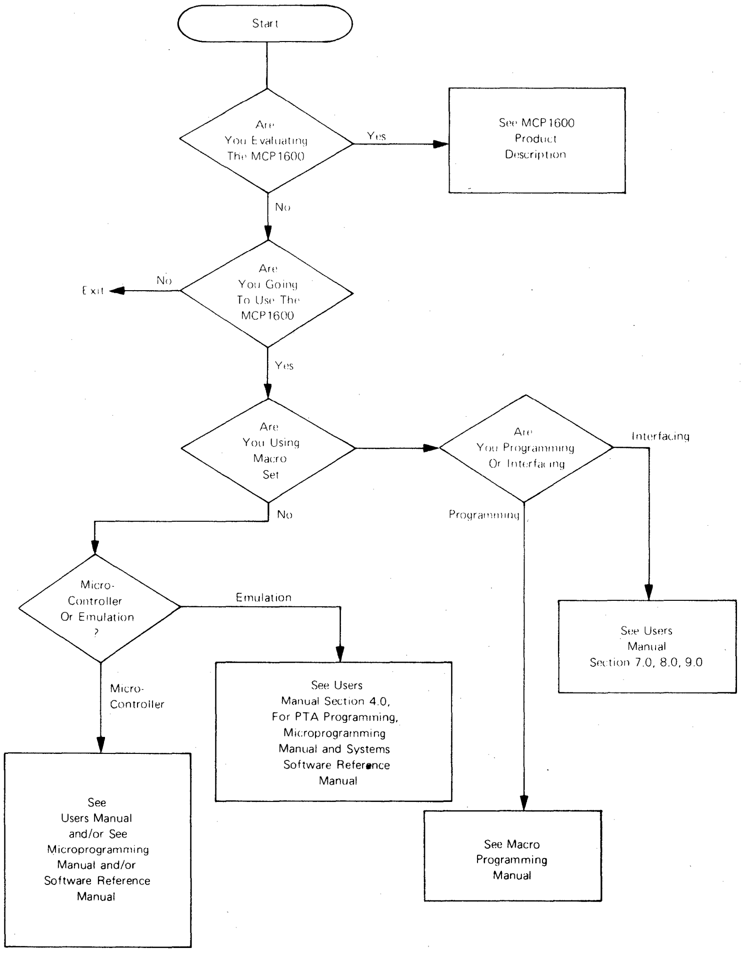
PREFACE

USING THIS MANUAL

This manual is intended to be used by those who need a detailed description of the internal operation of the MCP1600 Microprocessor Set. Users in this category are usually those who are implementing their own microcode structures and thus require a detailed knowledge of the machine.

- Sections 2,3,4 and 5 are of interest to the overall system architect
- Section 6 is of interest to the sophisticated user who will attempt to hookup to the MIB
- Section 7,8, and 9 are of interest to the interface designer

If you are not sure of your need for information, the attached flowchart may be of assistance. Contact a WDC Applications Engineer Representative for any additional information.



Start
Are You Evaluating The MCP1600
Yes
See MCP1600 Product Description
No
Are You Going To Use The MCP1600
No
Exit
Yes
Are You Using Macro Set
Are You Programming Or Interfacing
Interfacing
See Users Manual Section 7.0, 8.0, 9.0
Programming
See Macro Programming Manual
No
Micro-Controller Or Emulation ?
Emulation
See Users Manual Section 4.0, For PTA Programming, Microprogramming Manual and Systems Software Reference Manual
Micro-Controller
See Users Manual and/or See Microprogramming Manual and/or Software Reference Manual

This is a preliminary specification with tentative device parameters and may be subject to change after final product characterization is completed.

Information furnished by Western Digital Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Western Digital Corporation for its use; nor any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Western Digital Corporation. Western Digital Corporation reserves the right to change said circuitry at any time without notice.

SECTION I

GENERAL SYSTEM CONFIGURATION

1.1 GENERAL

The Western Digital MCP1600 microprocessor is an 8-bit microprogrammable computer implemented with 3 MOS/LSI chips using WDC's N-channel silicon gate process. The chip set consists of the CP1611B Data chip, the CP1621B Control Chip, and the CP1631B microinstruction ROM (MICROM) chip.

These chips are interconnected by the 22 bit Microinstruction Bus (MIB) which provides bi-directional communication between the chips for addresses and instructions. A Data Access Bus provides a 16 bit port for communicating with other system components such as memory and I/O.

The list below describes the pertinent aspects of the set:

- 8-bit Internal Organization
- 16-bit Data Access Port to Memory and I/O
- 26 Registers
- Extensive Microinstruction Set Including Decimal Operations
- Single and Double Byte Operations
- Micro and Macro Level Condition Flags
- 512 Word x 22-bit Control ROM
- Control ROM Expandable to 2048 words
- Micro Level Subroutine Capability
- Programmable Translation Array for Macroinstruction Interpretation
- Four External Interrupts and Three Internal Interrupts
- External Instruction Set
- Power Supplies +12V, +5V, and -5V
- 3.3 MHZ Four Phase Clock (Also available in 2.75 and 2.2. MHz versions)
- TTL Compatible 3 State Interface

The MCP1600 Microprocessor Set is easily interfaced to a variety of applications through many of the standard product offerings of Western Digital and other semi-conductor companies.

1.2 UNIQUE FEATURES OF THE MCP1600

The MCP1600 has two uncommon and usable features that set it apart from the standard run of MOS microprocessors.

- The MCP1600 is a vertically microprogrammable processor.
- The MCP1600 has a programmable translation array which quickly and automatically generates the jump vectors required for emulation and macroinstruction applications.

The unusual feature of being able to decode macroinstruction quickly and automatically sets the MCP1600 apart from other devices in the microprocessor market.

Unlike other offerings in the MOS microprocessor marketplace, the MCP1600 can be applied to problems that demand speed for their solution. The combination of speed (maximum 3.3 MHz instruction rate) and high density (thus great processing power) brings the economic advantages of MOS to bear upon problems which previously could only be attached by bipolar microprocessors or special logic designs.

Careful review of the contents of this manual will stimulate the imaginative user to think of many applications to which these unique features can apply significant advantage.

1.3 AREAS OF APPLICATION

One of the unique attributes of the MCP1600 is its ability to be sensitive to data. The Programmable Translation Array provides this ability by allowing the address of the next instruction fetched to be determined by data, if desired. This makes the MCP1600 highly suitable to areas of application that require a substantial amount of data decoding. Such application might be:

- The emulation of an existing computer's instruction set.
- The invention of a new, application oriented instruction set.
- The construction of a macro-instruction processor
- The construction of a data driven processor

Another of the attributes of the MCP1600 is its ability to quickly process data in a sophisticated manner. The availability of a rich instruction set coupled with speed and a variety of registers make the MCP1600 especially applicable to problems that require great quickness. Applications such as:

- Communications Multiplexing
- Host computer front ends
- Medium Speed Peripheral Device Control

can readily be attacked by the MCP 1600.

The MCP1600 provides still another advantage to the system designer. It is truly easy to interface. Including the processor set and all other logic a microcontroller can be implemented with as few as 15 parts. Figure 1-2 shows such a system.

Specific application notes detailing ideas relevant the application of the MCP1600 to the above mentioned areas are available from your Western Digital sales representative.

1.4 BLOCK DIAGRAM

Figure 1-1 describes the interconnection of the required components of the MCP1600 microprocessor set. The DATA CHIP, (CP1611B) contains the arithmetic logic unit, the microinstructions decode and the register file. Additionally, it contains paths to control the operation of the processor.

The CONTROL CHIP (CP1621B) contains the program translation array, portions of the control circuitry to control operation of the processor set, the microinstruction counter and the I/O control system.

The MICROM CHIP (CP1631B) contains the microinstruction ROM. The MP1600 microprocessor set may be expanded up to four MICROMs giving the user a total of 2,048 22 bit microinstructions. The simple system illustrated in Figure 1-2 shows all of the functional components needed to make a working microprocessor. Note that, in addition to the three parts comprising the Microprocessor Set, twelve other available standard TTL parts are required. These parts serve to:

- Generate the clocks
- Latch and gate input signals
- Latch and gate output signals

This is to be contrasted with some other commercially available microprocessors which may require as many as 50 other parts to implement a system of equivalent capability.

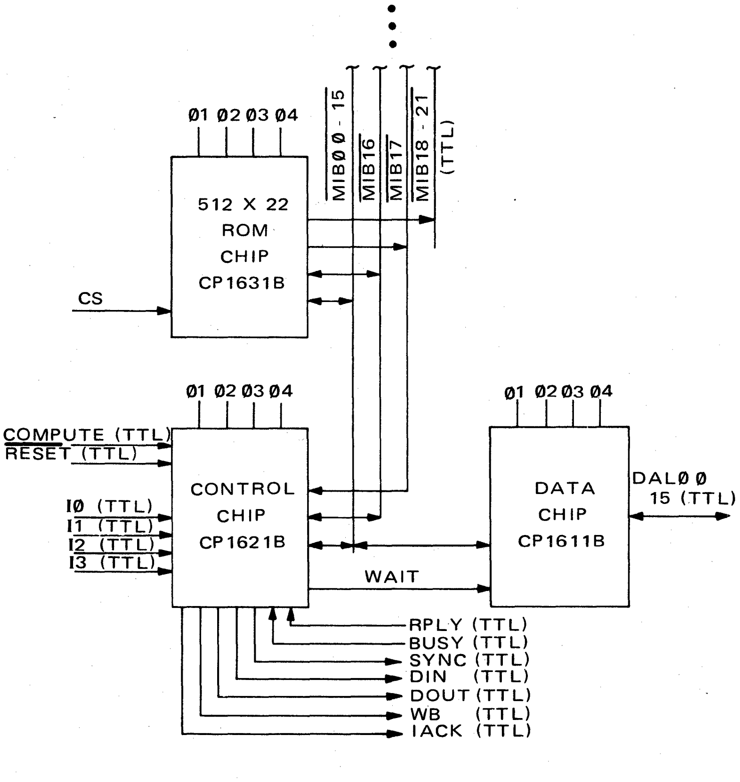
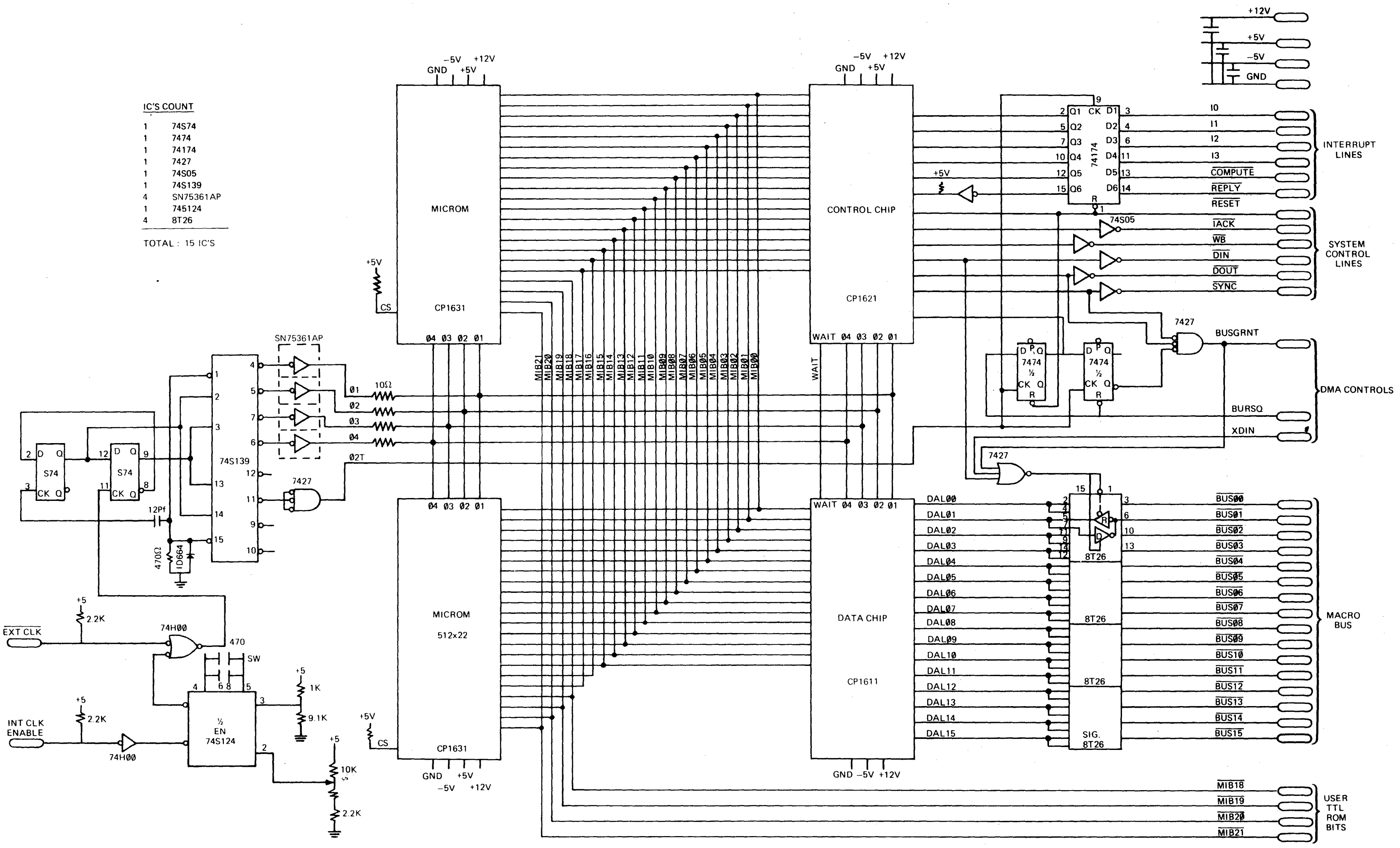


FIGURE 1-1 MCP1600 MICROPROCESSOR BLOCK DIAGRAM



IC'S COUNT

1	74S74
1	7474
1	74174
1	7427
1	74S05
1	74S139
4	SN75361AP
1	745124
4	8T26

TOTAL : 15 IC'S

FIGURE 1-2
A SIMPLE MICROPROCESSOR SYSTEM

SECTION II

SYSTEM COMPONENTS

2.1 GENERAL

This section describes the main functional components of the MCP1600 system. The reader should be aware that the system is physically partitioned into three kinds of devices. He should be more interested and aware of the logical partitioning of the system, which encompasses five areas:

- Processing and data handling. This is generally handled on the CP1611B Data Chip.
- Next Instruction Address Generation. This is generally handled on the CP1621B Control Chip.
- Microinstruction Storage. These are contained on one or more CP1631B Microinstruction ROMs.
- Microinstruction Bus. This bus connects the three types of devices together and provides a path for the microinstructions to flow from the microinstruction ROM to both the Control and the Data Chip.
- Data Access Bus. This bus provides access from the MCP1600 microprocessor set to the outside world. Lines comprising this data access bus come from all three of the chip types. Lines containing the address and data come from the Data Chip, control lines defining the state of the data access come from the Control Chip and user programmable control lines come from the Microinstruction ROM.

While not strictly a logic component, the clocking scheme should be noted here.

The MCP1600 operates on a four phase clock system. While there are some variations, the general use of each phase period is:

–	Ø1	Instruction Access
–	Ø2	Data Access
–	Ø3	Execution
–	Ø4	Data Update

The clocking scheme can be more easily understood by reference to section III and section VIII.

This section will describe the functional components that make up these partition areas.

2.2 REGISTERS

This section will describe data handling registers in the MCP1600 set of interest to the system designer.

- Register File. The Register File consists of 26, 8 bit registers which provide RAM data storage for the MCP1600 processor set. The register file has 2 output ports and 1 input port. Fourteen of the registers of the file can be directly addressed by the A and/or B register designators of a microinstruction. Additionally, the 16 top-most registers of the register file may be considered as register pairs and can be addressed by the G register (see below) to permit operation on full words of data. The Register File is on the Data Chip.

  The A and the B output ports of the register file feed into the ALU.

- G register. The G register is a pointer register on the Data Chip that describes the currently accessed linked consecutive pair of registers in the register file. Figure 2-1 describes the interaction between the G register and the register file. Note that, when the G register is being used, access to the register file is from the top down. This is opposed to the access to the register file when only the A and B fields are being used as designators, in which case it is from the bottom up. This register is loaded by IW and LGL instructions. "Input Word" instruction loads the G Register from the DAL bus as specified by the "b"field of the instruction. "Load G Low" instruction loads the G Register Ra.

  The first (or lowest) 14 registers of the Register File are addressable only from the MIR register. The top 12 registers are addressable only from the G-Register. The middle 4 registers are addressable from either the G or the MIR registers. Figure 2.1 describes the addressing conventions of this file. It is helpful to note that if the a or b fields of the MIR are 1 or 0, then this enables G-Register addressing.

Some Examples

– Assume G = 4, a = 0, b = B
Then one operand (A Port) will be from G'8'
The other operand (B Port) will be from R'B'

– Assume G = 0, a = 7, b = 1
Then one operand (A Port) will be from R'7'
The other operand (B Port) will be from G'1'

– Assume G = 6, a = 1, b = 0
Then one operand (A Port) will be from G'D'
The other operand (B Port) will be from G'C'

– Assume G = 3, a = 6, b = A
Then one operand (A Port) will be from R'6'
The other operand (B Port) will be from R'A'

– Assume G = 5, a = 0, b = 1
Then one operand (A Port) will be from G'A'
The other operand (B Port) will be from G'B'

REGISTER FILE ADDRESSING MODES

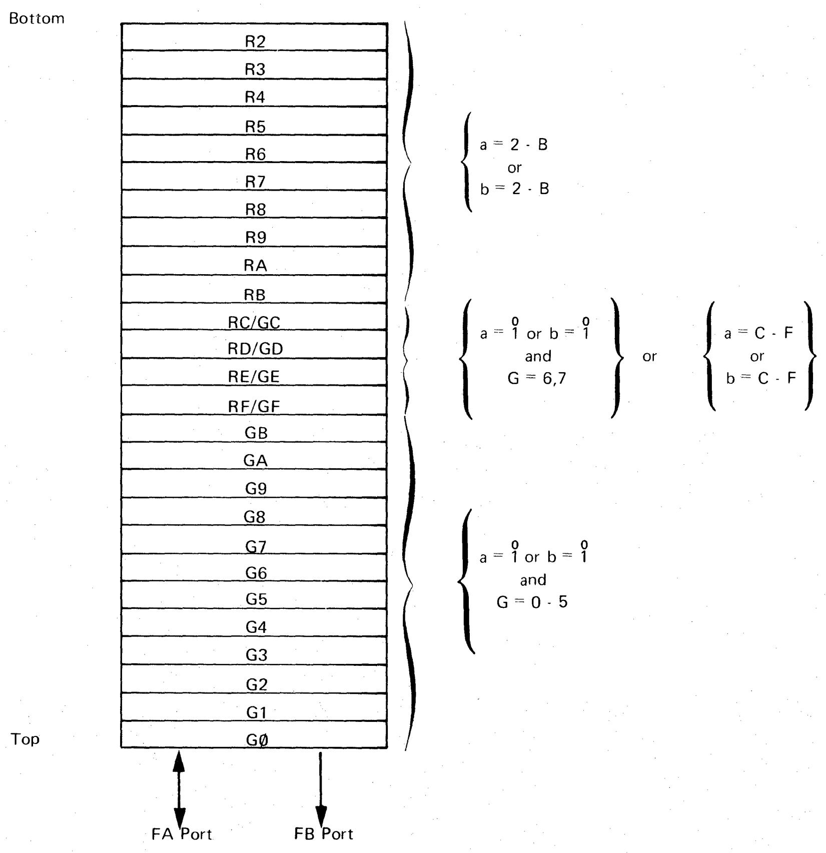


FIGURE 2-1
REGISTER FILE

— ALU Status Bits. The ALU provides 4 status bit outputs which reflect the result of each 8 bit operation. The ZB and NB bits are updated after every ALU operation. C4 and C8 are updated for Arithmetic or Shift operations. These bits may be tested by Jump but their primary function is to pass result data from the first cycle of a word instruction to the second cycle.

The available status bits are:

ZB: Set if the result of a Byte or Word operation is zero; cleared otherwise.

NB: Set if the high-order bit of a Byte or Word operation is a one; cleared otherwise (except for SRW and SRWC).

C4: Set if Carry Out of Position 3 is a one; cleared otherwise. Updated only on Arithmetic operations. This status bit is used mainly for decimal arithmetic corrections.

C8: Set if Carry Out of Position 7 is a one; cleared otherwise. (Note that this status bit is not set to borrow for subtract as is the case with the C Flag.) Also set if the shifted off bit of a Shift operation is a one; cleared otherwise.

— Condition Flags. The Condition Flags consists of 4 latches which can reflect the status of the previous ALU results. The updating of these flags can be selectively enabled or disabled at the discretion of the microprogrammer. The condition flags are updated with odd-numbered instruction opcodes in the range of 8Ø-EF.

Z Flag: Set if the result of a Byte or Word operation is zero, cleared otherwise.

N Flag: Set if the high-order bit of the result of Byte or Word operation is (except for SRW and SRWC) is a one; cleared othewise. (Note that this is the complement of the sign of the result if overflow occurs.)

C Flag: Monitors the carry, borrow and shifted off bits as follows:

Add and Increment: Set if there is a carry from the most significant bit of the Byte or Word result; cleared otherwise.

Subtract and Decrement: Set if there is a borrow (complement of carry) from the most significant bit of the Byte or Word result; cleared otherwise.

Shift: Set if the bit shifted off in a left or right shift is a one; cleared otherwise.

The C Flag is not affected for operations (other than those listed above) even if the other flags are updated.

V Flag: Set if there is an arithmetic overflow on Arithmetic operations. Cleared if there is no overflow and on Non-arithmetic operations. On Add operations overflow occurs when the sign of the two operands are the same and the sign of the result is different. On Subtract operations overflow occurs when the signs of the two operands are different and the sign of the result is different from the operand in Ra (V=C7 ⊕ C8).

The format of the ALU status and flag register word is shown below:

NB	ZB	C4	C8	N	Z	V	C
7	6	5	4	3	2	1	0

— MIR Register. The 16 bit MIR register holds the least significant 16 bits of the 22 bit microinstruction currently being executed. The MIR register is loaded every Ø 1 unless the currently executing microinstruction is of the 2 cycle variety. In this case the register is not clocked on the second occurrence of Ø 1.

2.3 INSTRUCTION ADDRESS GENERATION

This section describes the register and arrays used in generating the next microinstruction address.

- Location Counter Register. The Location Counter (LC) is an 11 bit register which holds the address of the next microinstruction to be accessed. Under normal conditions, the LC is incremented by one after each access of a microinstruction. The LC can be altered by other than one by execution of a Jump instruction or a Return from Sub-routine (RFS) instruction or by invoking a PTA translation.

- Return Register. The Return Register (RR) holds an 11 bit address and permits a subroutine depth of one in the microprogram. When a subroutine jump is indicated (MIB 16 = 1) the return register stores the incremented contents of the LC. Contents of the Return Register can be transferred to the LC under the control of a RFS - (Return From Subroutine).

- Translation Register. The Translation Register (TR) is a 16 bit register which holds the data presently being input to the PTA. Data on the Microinstruction Bus (MIB) is used to load the Translation Register. Note that only the upper or lower half of the Translation Register can be fed into Array 3 at one time.

- Translation Arrays. There are four arrays on the control chip that control the generation of the next microinstruction address. They generate new inputs into the LC register, as a function of the LC register contents, the contents of the translation register, interrupts, and other miscellaneous inputs. The detailed operation of these arrays is discussed in Section 4, PROGRAMMED TRANSLATION ARRAY OPERATIONS.

2.4 MICROINSTRUCTION BUS

The Microinstruction Bus serves to interconnect the three required parts of the MCP1600 Microprocessor set. The bus is organized into 4 distinct sections:

- MIB00 - MIB15 carry proper microinstruction data from the microinstruction ROM chip to both the control chip and the data chip. These lines may also carry data between the data chip and the control chip under certain conditons.

- MIB16 controls the subroutine jump operation. When MIB16 is set on a jump instruction, it causes the incremented contents of the LC register to be placed into the Return Register.

- MIB17, if set, causes the PTA on the Control Chip to perform a Read Next Instruction translation.

- MIB18 - MIB21 are TTL level outputs that can be programmed by the user as his needs dictate. They are not used by the MCP1600 Microprocessor set to control its operations. They are provided for the convenience of the user in order that he may interface to the MIB bus or control devices directly from the microprogram level.

2.5 DATA ACCESS BUS

The Data Access consists of three sections:

- DAL00 - DAL15 carry address and data between the data chip and the outside world.

- Outside Control Lines. There are five TTL level Outgoing Control Lines that inform the outside world of the present state of the MCP1600 set. These signals include Sync, Input Instruction, Output Instruction, Interrupt Acknowledge, and Write-Byte.

- Incoming Control Lines. There are 8 TTL level Incoming Control Lines. These incoming control lines inform the MCP1600 Microprocessor set of the state of affairs in the outside world. There are four interrupt lines, a Compute line, a Reset line, a Reply line and a Busy line.

With these lines, the MCP1600 can control a wide variety of peripheral devices.

2.6 MICROINSTRUCTION STORAGE

MICROM

The microinstruction ROM is a 512 X 22 -bit word, high speed ROM which stores the instructions of the microprogram. The transfer of addresses into the chip and the microinstruction out of the chip are performed over the MIB. Address is received from push-pull drivers in the Control Chip on Ø2. The decoding takes place on Ø3. On Ø4 the selected microinstruction is internally accessed and the MIB is precharged. The accessed microinstruction is placed on MIB15-MIBØ for transfer to the Data Chip and Control Chip during Ø1.

SECTION III

CPU OPERATIONS

3.1 GENERAL

This section describes the internal workings of the Data Chip, the CPU of the MCP1600 system. Its purpose is to provide the designer with a background understanding to enable him to more effectively interface with the Micro Instruction Bus and the Data Access.

The description of the MCP1600 Processor must proceed from the knowledge that there are two important and distinct partitions to the processor. The first partition, called the Data Chip, provides classical stored program processing. The Data Chip consists of:

- A register file
- An arithmetic logic unit
- A Microinstruction register
- A register file address decoder
- A control signal generation function
- Condition flags
- Jump decoding
- Input/Output gating

The data chip is responsible for data manipulation as a result of instruction execution.

The second partition, the Control Chip, generates address data that directs the accessing of the next microinstruction to be executed. This address generation mechanism provides the MCP1600 with its unique emulation capabilities. While the Data Chip is executing the presently fetched microinstruction, the Control Chip is performing a transformation upon the presently executing macroinstruction to determine the address from which to fetch the next microinstruction to be executed.

The description of the processor proceeds by first outlining the operation of the Data Chip (CPU) and then in the following section outlining the operation of the Control Chip. It is important to note that the two chips are not synchronized except by a common clock and the fact that the contents of the MI register on the Control Chip and the contents of the MIR register on the Data Chip are the same. One control line is passed between the Data Chip and the Control Chip.

3.2 CPU OPERATIONS

We start the CPU operation description by first assuming that the machine has been properly reset and has just entered the compute mode. The Control Chip will cause microinstruction to be fetched from the MICROM to be placed on the MIB bus. (See Figure 3-1) Assuming that the clock is now on the leading edge of Ø 1, the contents of the MIB bus are gated into the MIR register on the Data Chip and the MI register on the Control Chip.

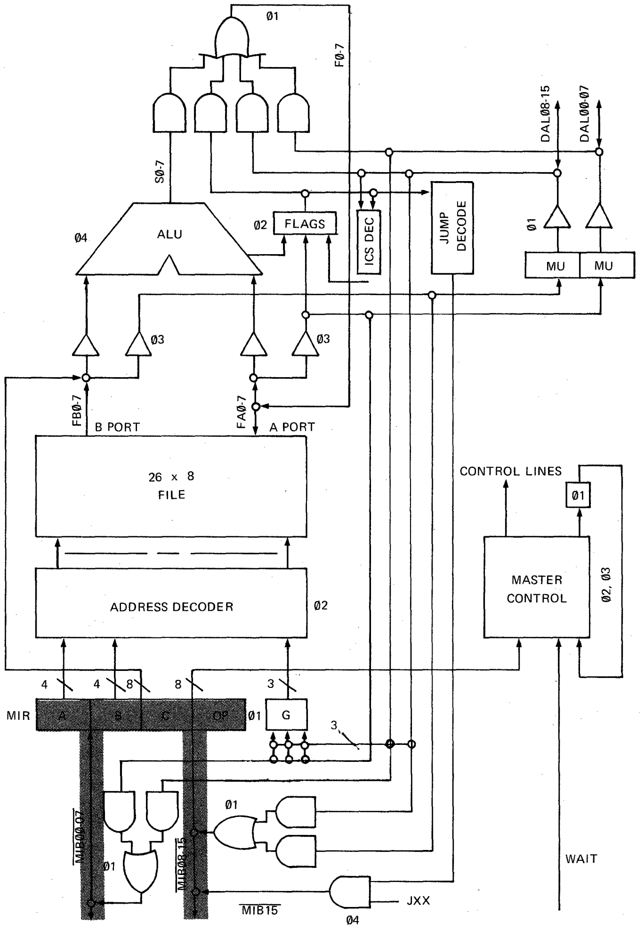


FIGURE 3-1
Ø1 DATA PATHS

At the end of Ø 1 (trailing edge of the clock), the contents of the MIR register have settled. The contents on the MIR register are held fixed until the next occurance of Ø 1 (four phase times).

At the beginning of Ø 2, the contents of the MIR register are presented to the various different functions of the CPU. The A and B fields are presented to the address decoder, as is the G register. The contents of the MIR register's C field and OP field are presented to the master control function of the Data Chip. During Ø 2 and Ø 3 the master control function decodes the C field and generates signals which control the gating of data to the various functions of the Data Chip. Also during Ø 2 the address decoder is interpreting the contents of the A field and the B field and accessing the appropriate register in the register file. Also during Ø 2 the condition codes reflecting the result of the last ALU operation become valid. They are presented to the FLAGS register which retains them for examination on the next occurance of Ø 4. Figure 3-2 illustrates Ø 2 data paths.

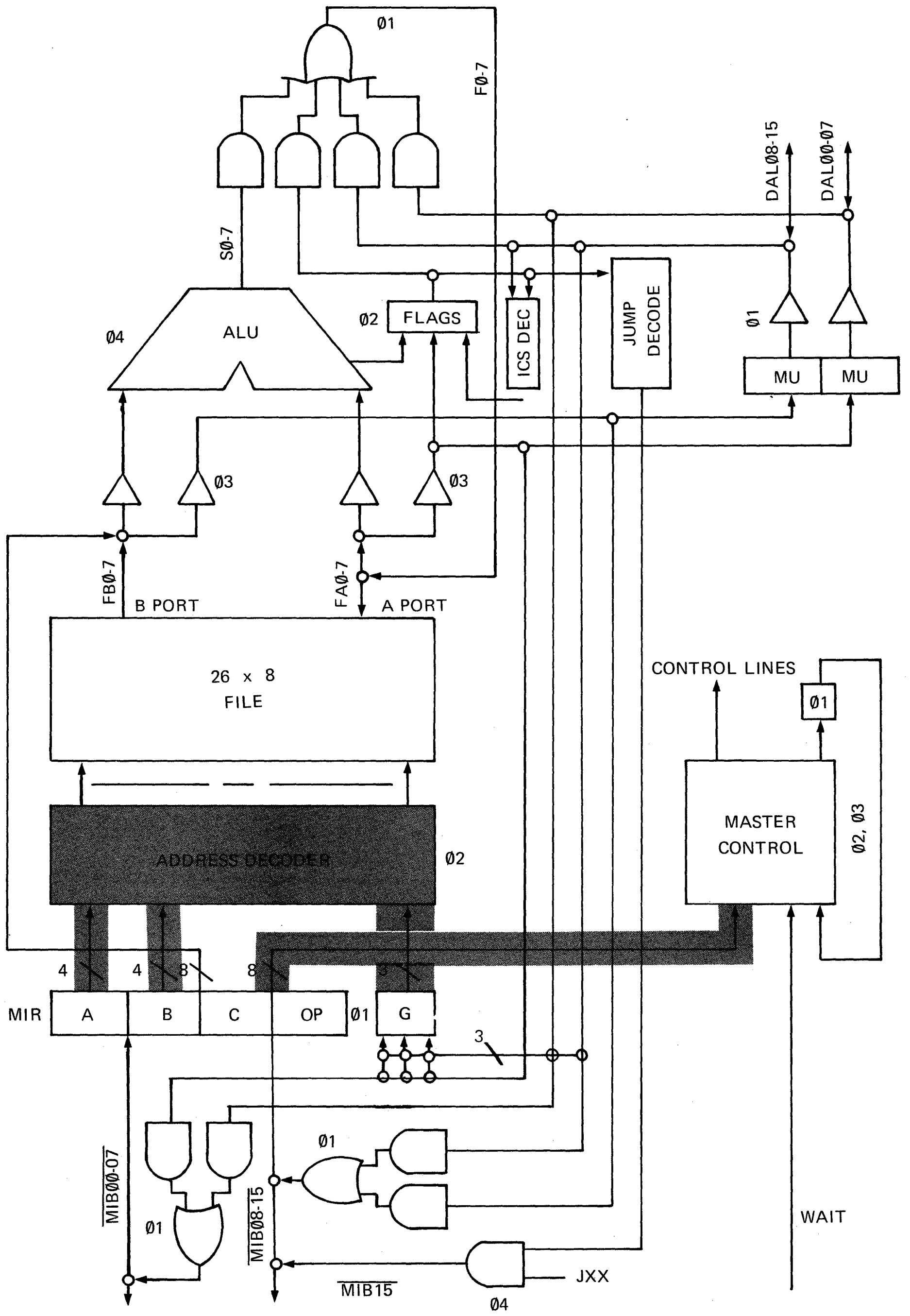


FIGURE 3-2
Ø 2 Data Paths

Phase 3 is the register access phase. During Ø 3 data is presented to the ALU by the A and the B output ports of the register file. Data may also be presented to the ALU B port by means of the literal path from the MIR register. Note the literal path (whose contents are the C and B field of the MIR) bypasses the address decoding logic and the register file. Also during Ø 3 the contents of the A and B register file output ports may be presented to the Data Access on line DALOO --DAL15. Also during this phase time the contents of the register file A port may be presented to the flag register. Finally, during this phase time, the contents of the register files A and B ports may be presented to the gating which controls access to the MIB bus. Figure 3-3 illustrates Ø 3 data paths.

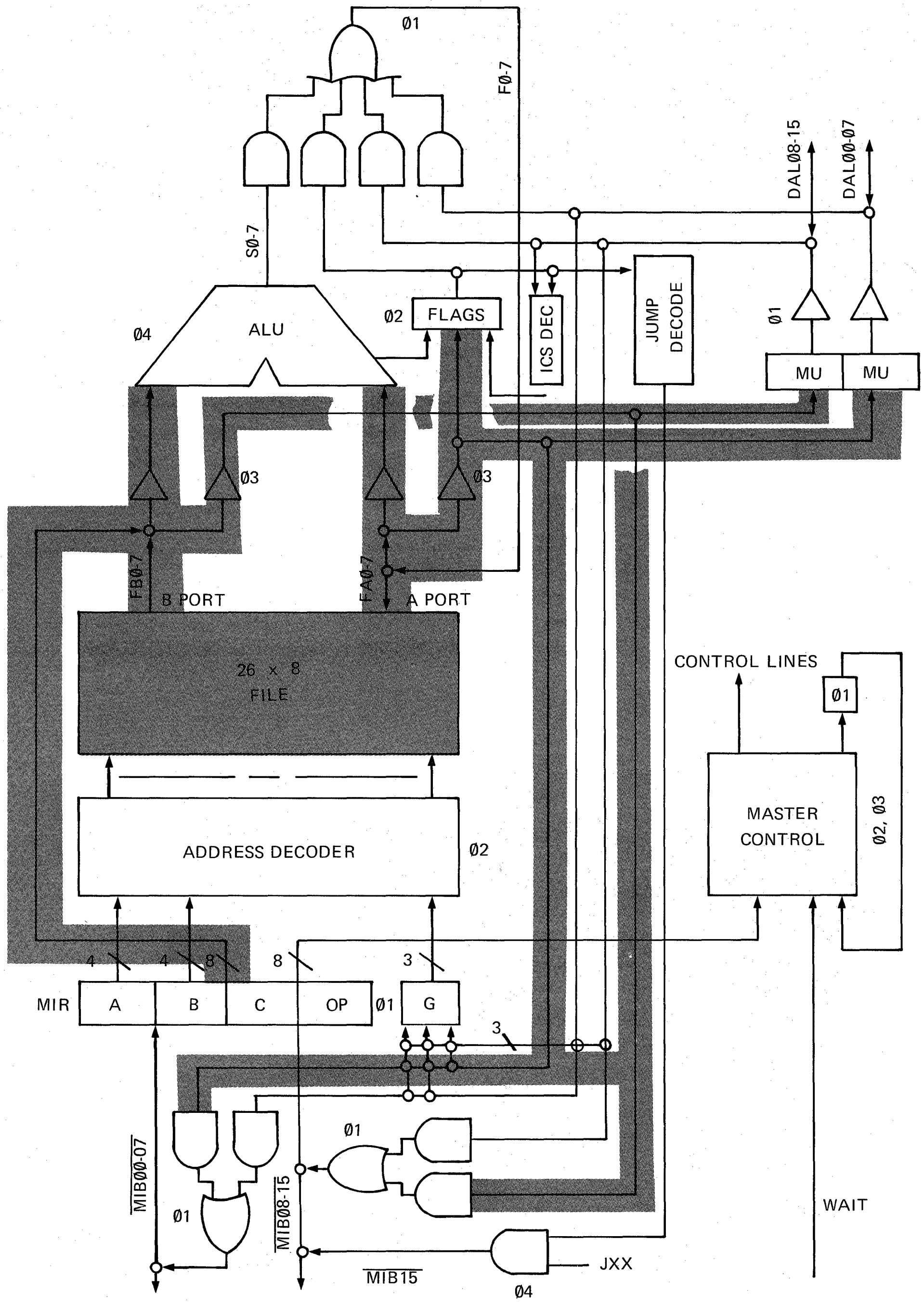


FIGURE 3-3
Ø 3 Data Paths

During Ø 4 the output of the ALU becomes valid. Also during Ø 4, input to the ALU may occur from the data bus. That is, data presented on lines DALOO--DAL15 may be presented to the gating structure which controls the output of the ALU. Finally, during Ø 4, the data available from the Data Access may be presented to the gating structure that controls access to the MIB bus. Figure 3-4 illustrates the Ø 4 data paths. Note that, during Ø 4 the jump control takes place. If the jump decoder indicates that all the proper jump conditions are met and if JXX is asserted, then MIB15 will cause the Control Chip to effect a jump on the next clock cycle.

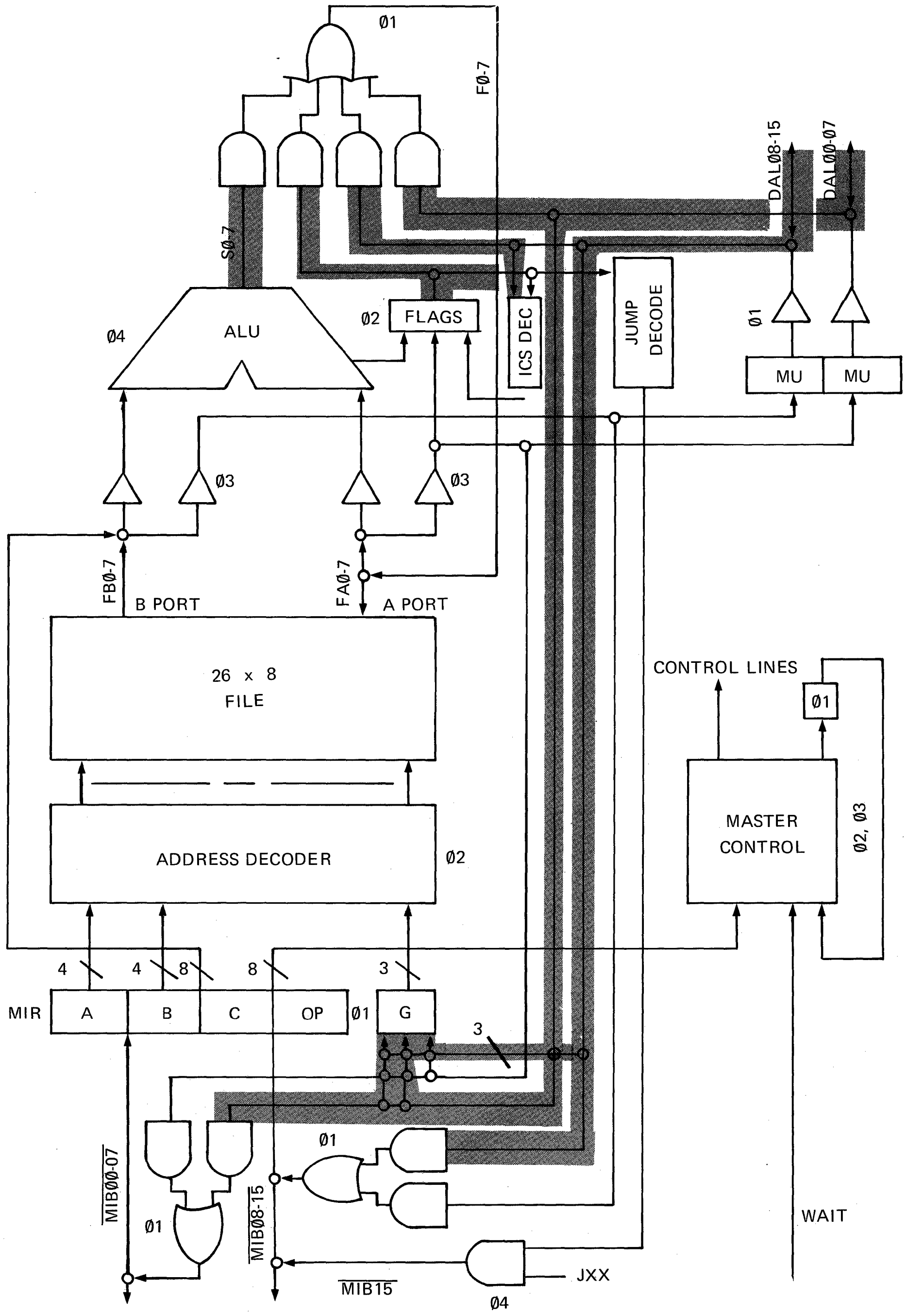


FIGURE 3-4
Ø 4 Data Paths

As the next Ø 1 clock occurs to start the next cycle of the instruction/execution, the data that was presented to the gating structure that controls access to the A input port of the register file is gated or not gated, depending upon a signal from the Master Control, into the register file. Figure 3-5 illustrates this path. If the instruction contained in the MIR register is one that requires only one clock cycle for execution, then a new instruction is fetched from the MICROM according to the address generated by the Control Chip and execution begins as in Figure 3-1.

If however, the instruction requires two cycles for execution, things proceed differently. At the beginning of Ø 1 of the second cycle, the contents of the MIR are preserved, no new data is gateo in. The low order bits of both the 'A' and 'B' fields are complemented to access adjacent slots in the register file to those previously addressed in the first cycle. Also during this phase, the CPU may output data to the MIB bus. Figure 3-5 illustrates this path.

Other than these exceptions, two cycle instructions proceed as do one cycle instructions.

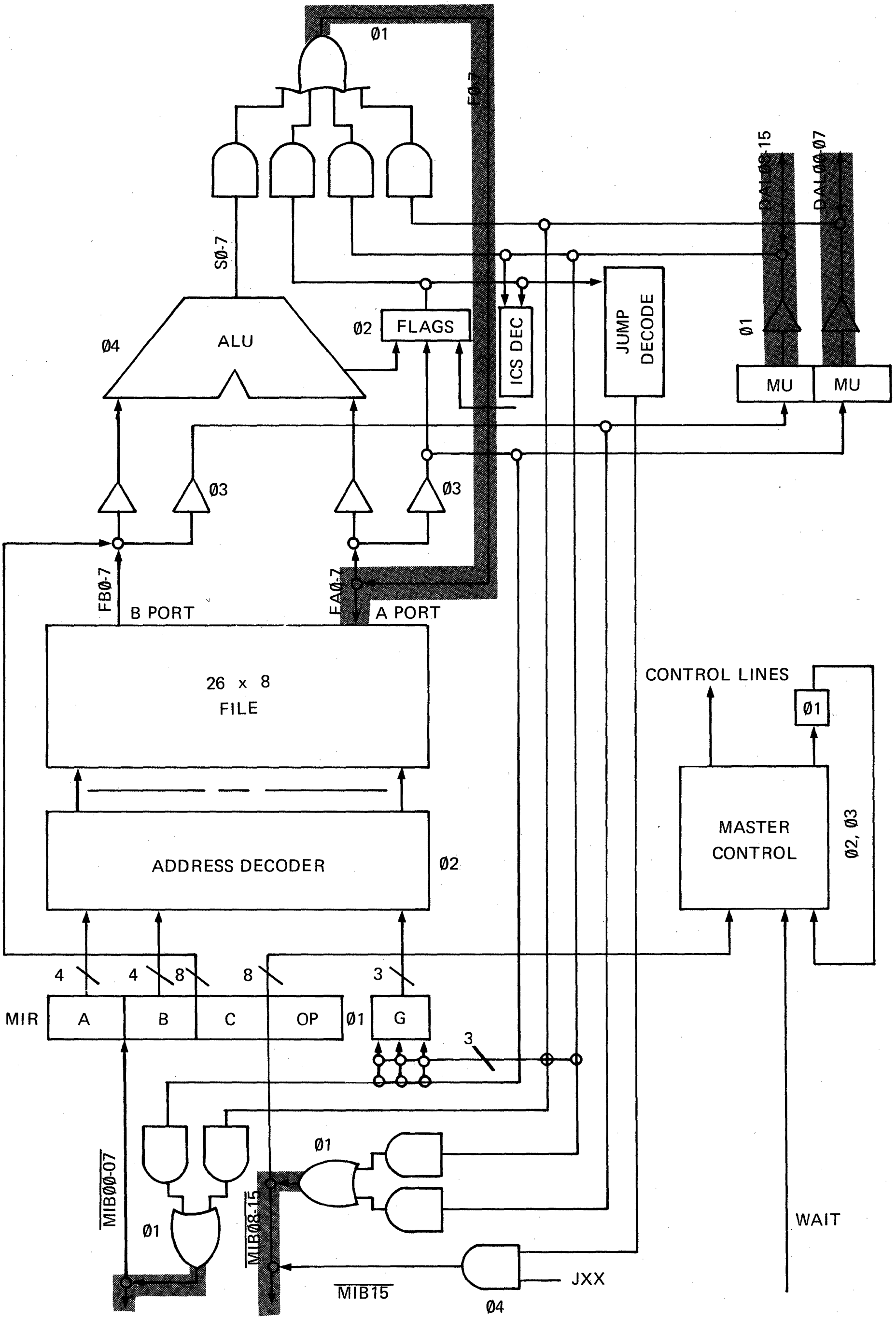


FIGURE 3-5
Ø1 Data Paths
Second Cycle

3.3 CONTROL LINES

During Ø 2 and Ø 3 the Master Control function is decoding the new contents of the MIR register in order to generate signals to control the data transfers described above. These control signals are generated on the basis of the contents of the C and OP fields of the MIR register. All of the control lines generated by the Master Control function are not described here because some of them serve timing and sequencing functions that are not necessary to the understanding of the CPU operation and how it affects the Data Access and Microinstruction Bus. However, there are several control lines which the designer may find helpful to understand. They are shown in the attached diagram, Figure 3-6, and, in order, they are:

3.3.1 REGISTER LOAD

This control line determines whether or not the output of the ALU will be gated back into the register file input port. It is invoked by register to register format instructions.

3.3.2 LOAD FLAGS

This line controls whether or not the output of the FLAGS register will be gated into the register file input port. It is generated by the Load Flags and Copy Flags instruction.

3.3.3 LOAD HI BYTE

This control line determines whether or not the contents of the DAL08-15 will be gated into the register file input port. This line is controlled primarily by the Input class of instructions.

3.3.4 LOAD LO BYTE

This control line determines whether the contents of DAL-07 will be gated into the register file input port. It is generated by input class instructions. Note that this signal and preceeding signal are generated by the specification in the B field of the appropriate input instruction. Note also that one can input the low order byte, the hi order byte, or both bytes. This latter case is an Input Word Instruction.

3.3.5 MODIFY LO

This control line is invoked by the Modify instruction and causes the output of the A register file port to be gated on to the low order 7 bits of the microinstruction bus. This control line is also invoked by the Load Translation Register instruction.

3.3.6 MODIFY HI

This control line is the high order analog of the previously described control line.

3.3.7 JXX

This control line is invoked by the jump instruction. It is made hi during the first phase four of the jump instruction, if the jump instructions have been met. It causes the contents of the MI register on the control chip to be placed into the LC register.

3.3.8 LOAD LIT

This instruction controls the multiplexer which selects the B input to the ALU. The ALU B input port may be fed from either the literal field of the MIR register or the B port output of the register file. This control signal is set by the literal class of instructions and causes the B input to the ALU to be selected from the literal path.

3.3.9 DOUBLE/SINGLE

This control line is generated by the Master Control for use by itself. It is invoked by instructions that require two cycles for execution. In the main, it controls whether or not a new instruction is gated into the MIR register at the next occurance of Ø 1.

3.3.10 LOAD TR

This control line is invoked by the Input Word instruction if either bits 4 or 5 of the B field are set. These bits being set will cause the data on the Data Access to be placed into the translation register on the Control Chip. It is in this fashion that new macroinstructions are fetched from the user memory for decoding.

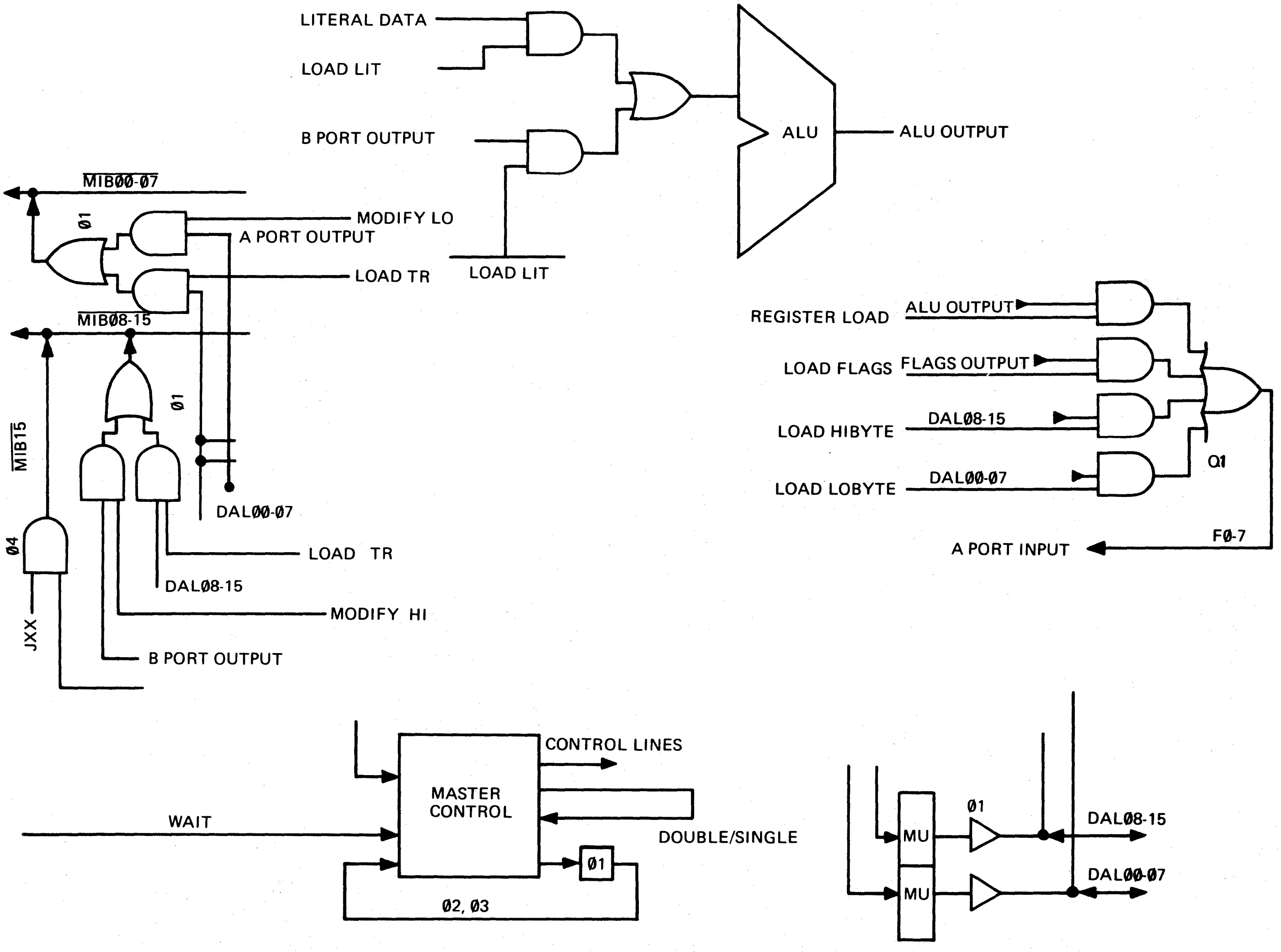


FIGURE 3-6
MAJOR CONTROL LINES

SECTION IV

PROGRAMMABLE TRANSLATION ARRAY (PTA)

4.1 GENERAL

This section describes the operation of the MCP1600 Microprocessor set's most unique feature, the Programmable Translation Array. The Programmable Translation array serves to generate new microinstruction fetch addresses as a function of several parameters. These parameters are those which are normally considered during the decode of a macroinstruction. While the Programmable Translation Array was designed specifically to eliminate most of the over head of macroinstruction translation, it is useful for other purposes. One could, for instance, conceive of a data driven processor that would determine the next operation to be performed on a data structure as a function of the present data and the present operation. The uses of the Programmable Translation Array are limited only by the system designer's imagination.

Briefly, the Programmable Translation Array consists of eight elements. They are:

- Location Counter.
- Array 1
- Array 2
- Array 3
- Translation Register
- Array 4
- Translation State Register
- Return Register

These elements give the capability of calculating microinstruction addresses other than sequentially, or as specified by a jump instruction. Figure 4-1 depicts their interconnection. The mapping (or new Location Counter value) that the PTA generates, when invoked, is a function of the value of the Location Counter, the contents of the Translation Register, the interrupts, and the state of the RNI line.

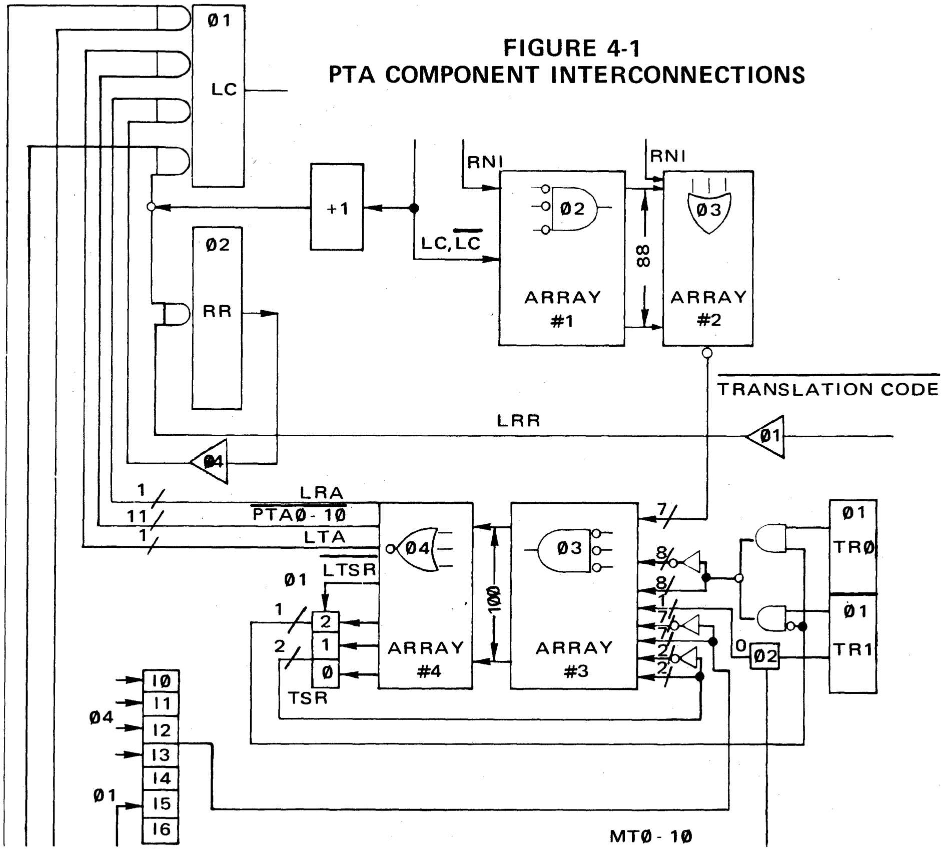


FIGURE 4-1
PTA COMPONENT INTERCONNECTIONS

4.2 PTA COMPONENTS

This section will describe the components of the Programmable Translation Array and their functions.

4.2.1 LOCATION COUNTER

It is the function of the Location Counter to hold the address of the next microinstruction . The Location Counter is 11 bits wide and has outputs to MIB00—MIB10 and to Array #1. It has inputs from the Master Control function, Array #4 and the Return Register. There is also an input to the Location Counter from an incrementer.

4.2.2 ARRAY #1

Array #1 is an 88 element array of active low input AND gates with 23 inputs. There are 11 true inputs and 11 complemented inputs from the Location Counter and the RNI bit (MIB17). Figure 4-2 illustrates the organization of Array #1. Figure 4-3 illustrates the concept, as opposed to implementation of a typical gate, one of 88, that make up Array #1. The outputs of these gates make up the bulk of the inputs of Array #2.

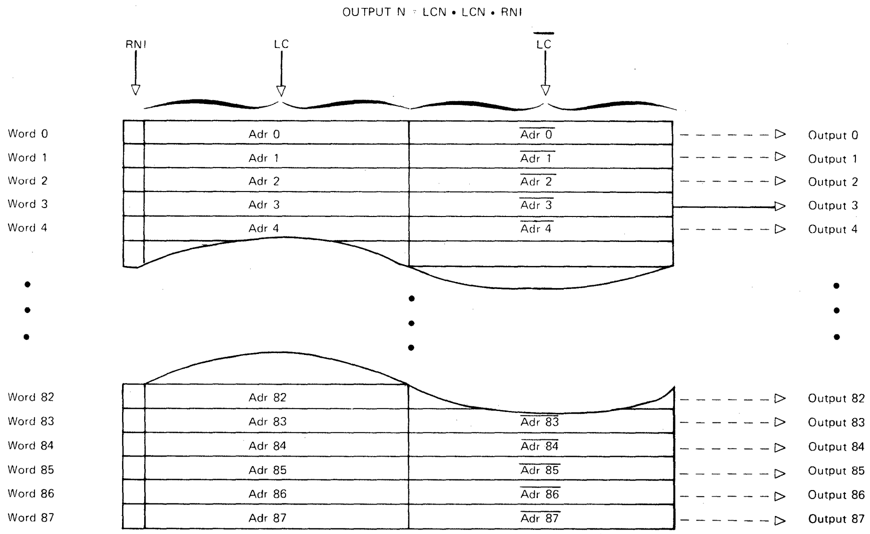


FIGURE 4-2 ARRAY #1 ORGANIZATION

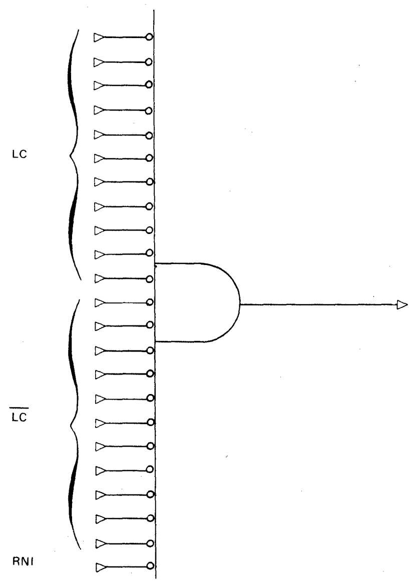


FIGURE 4-3 TYPICAL GATE - ARRAY #1

The purpose of Array #1 is to perform a selection function. When the inputs to Array #1 match one (or more) of the word values (addresses) the output associated with that word is asserted. This array is programmed with 6 hexidecimal digits as shown in Figure 4-4. Transistors are placed in the array for 1 bits in the data specification. To program a group of addresses the bits of the address to be left out are specified as Ø bits for the true and complement address inputs. This produces a 'DON'T CARE' effect for those bits and the result will always be a match. This lets the user match on modulo address.

The RNI line serves to inhibit the output of array #1. The reason for this is described later.

Ø	LC ADDRESS	Ø	$\overline{\text{LC ADDRESS}}$
23	22 12	11	10 0

Example:

Address = X' 5F8'

Array #1 Data Word = X' 5F8207'

FIGURE 4-4 ARRAY #1 DATA SPECIFICATION

4.2.3 ARRAY #2

Array #2 is an ORed array whose outputs are negated. The array has 89 inputs, 88 from Array #1 and one from MIB17, the RNI line. The seven outputs from Array #2 represent a translation state code and become inputs into array #3.

Figure 4-5 illustrates the orgainzation of Array #2. Figure 4-6 illustrates in some detail, the conceptual structure of the gate and interconnect structure of Array #2. The small circles present where the lines would normally cross represent connections that can or cannot be made. It is via the making or not making of the connections with transistors that this array is programmed.

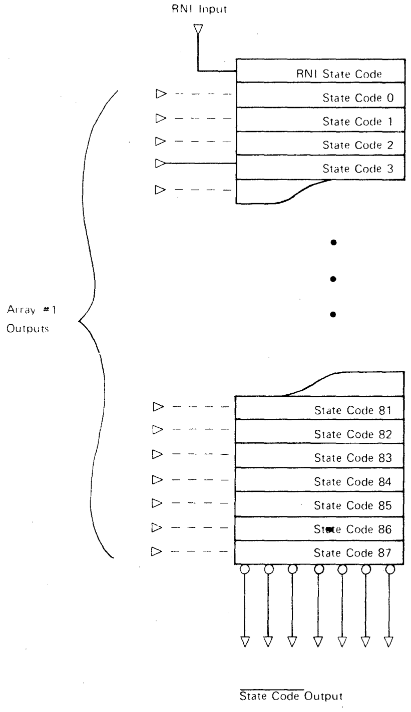


FIGURE 4-5 ARRAY #2 ORGANIZATION

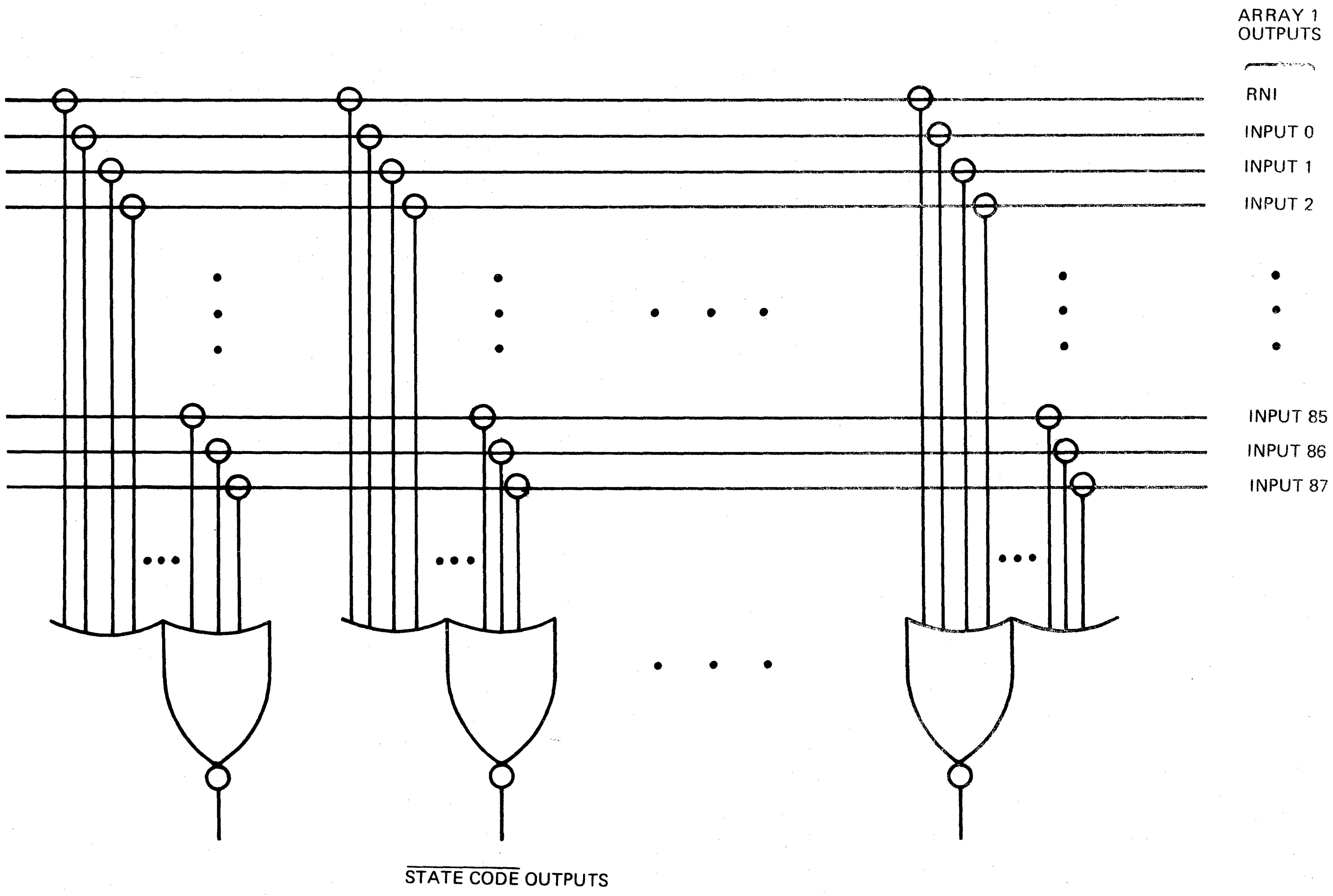


FIGURE 4-6
ARRAY #2 GATE AND INTERCONNECT STURCTURE

Array #2 is driven by the inputs from Array #1. When one or more outputs of Array #1 are asserted, Array #2 will make the corresponding word available as its outputs.

The purpose of Arrays #1 and #2 is to provide a mapping between the present value of the Location Counter and a preselected set of translation state codes.

Due to electrical considerations, there are restrictions placed on the configuration of Array #2. Normally this array is programmed in a 4 of 7 code (4 of 7 outputs may be selected to be high, the other 3 must be low), or more ideally, to reduce capacitive loading, a 5 of 7 code. The tables below describe the various state code outputs available in both the 4 of 7 and the 5 of 7 schemes.

ØF	2D	3C	56	69
17	2E	47	59	6A
1B	33	4B	5A	6C
1D	35	4D	5C	71
1E	36	4E	63	72
27	39	53	65	74
2B	3A	55	66	78

Table 4-1 Translation State Code Outputs -- 4 of 7

1F	3D	5B	6B	75
2F	3E	5D	6D	76
37	4F	5E	6E	79
3B	57	67	73	7A
				7C

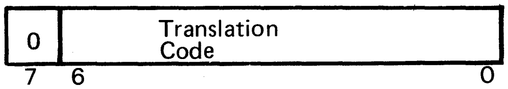
Table 4-2 Translation State Code Outputs -- 5 of 7

The outputs of Array #2 represent some of the inputs to Array #3.

The RNI line (MIB17) deserves mention at this point. It is the purpose of the RNI line to force a particular user defined state code from the mapping represented by Arrays #1 and #2. This state code will be independent of the contents of the Location Counter.

Accordingly, RNI inhibits the outputs of Array #1 and invokes that state code selected by the user in Array #2 and makes it present at the output of Array #2.

The programming of Array #2 is illustrated in Figure 4-7 below. Transistors are put in the arrays for zero bits as per the data specifications.



Example: X′ 56′

FIGURE 4-7 DATA SPECIFICATION

4.2.4 ARRAY #3

Array #3 consists of 100 active low input AND gates, and has 42 inputs. In contrast to Arrays #1 and #2, the inputs come from a variety of sources. The inputs to this array constitute a direct interface to the real world at the macromachine level, as one of the inputs is the translation register. The inputs to Array #3 are:

- Seven from Array #2. (Translation Code)
- Sixteen from the Translation register. Eight are true data and eight are complemented data.
- Fourteen inputs from the interrupt latches. Again, both true and complemented data are present.
- Two from the translation state register. Again, both true and complemented data are used.
- One input, called the Q signal, not generally usable.

It is worth noting, that, while there are 42 inputs in Array #3, they are not all present at the same time. The array is broken into two partitions. The first partition consists of words 0 through 15, and has as its inputs, the 14 bits from the interrupt latches. The second partition of the array, words 16 through 99 have, in these same slots, the 16 bits of inputs from the Translation Register.

The Figure 4-8 illustrates the structure of the first partition of the array. Figure 4-9 illustrates the second partition of the array. Figure 4-10 shows a conceptual picture of a typical gate in the array.

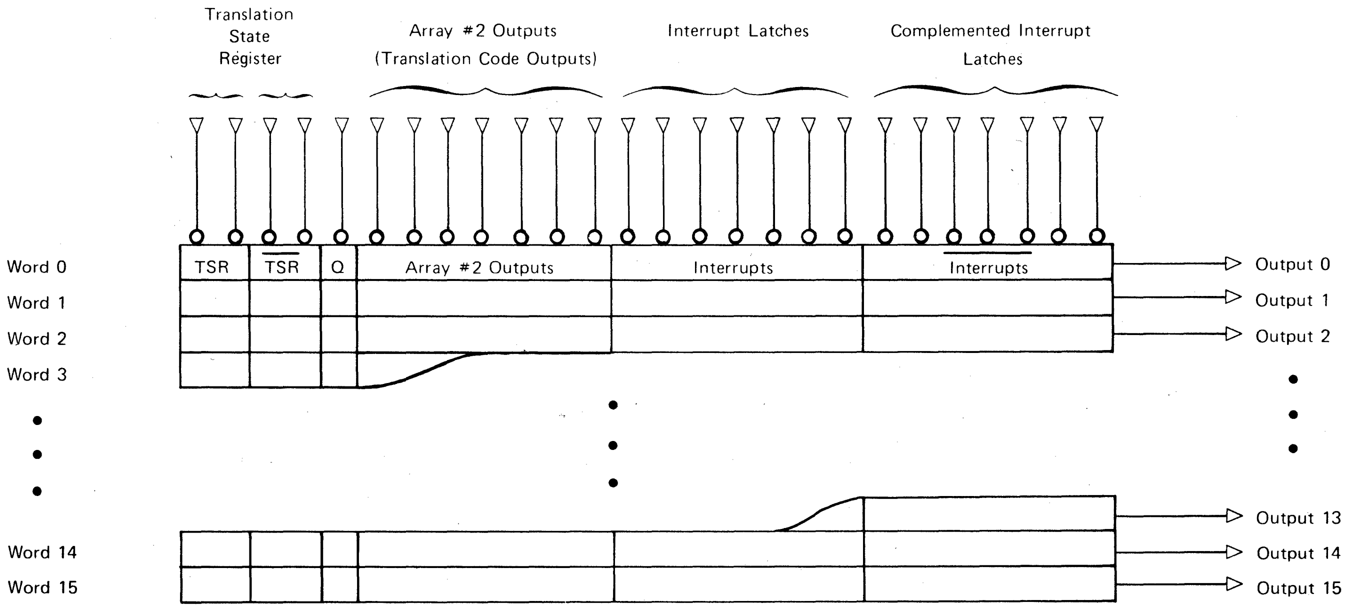


FIGURE 4-8 ARRAY #3 INTERRUPT ORGANIZATION

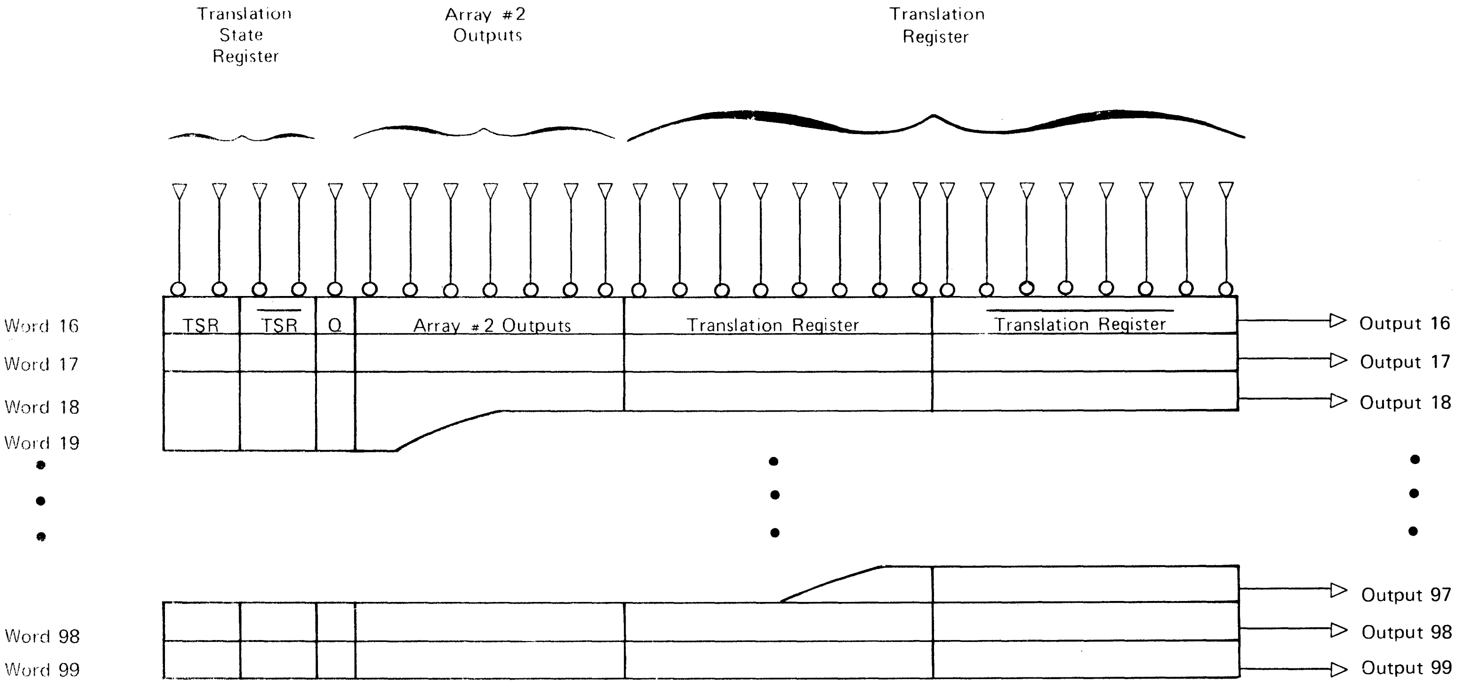


FIGURE 4-9 ARRAY #3 TRANSLATION REGISTER ORGANIZATION

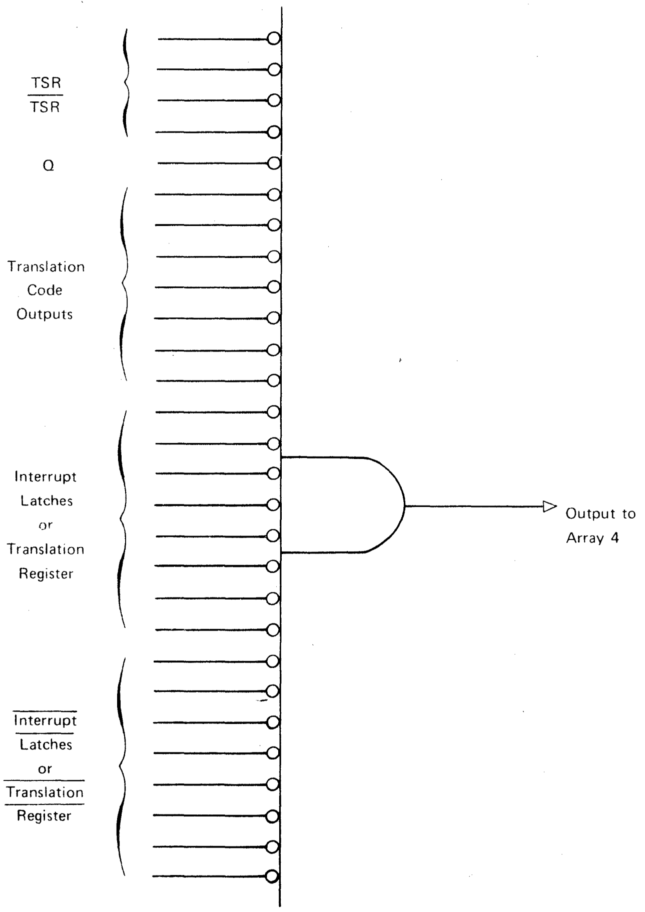

TSR
$\overline{TSR}$
Q
Translation Code Outputs
Interrupt Latches or Translation Register
$\overline{\text{Interrupt Latches or Translation Register}}$
Output to Array 4

FIGURE 4-10 TYPICAL GATE OF ARRAY #3

Array #3, being an ANDed array, will try to match the configuration of its inputs against one of the words that make up the array. If it finds a match, the output associated with that word will be asserted.

Array #3 is programmed with true data as eight hexadecimal digits. DON'T CAREs, which always result in matches, are created by placing both true and complemented bits to zero. Below is a data specification for the programming of Array #3.

AND Gates Ø- 15:

TSR	$\overline{\text{TSR}}$	Ø	Ø	Ø	Q	Ø	TRANSLATION CODE	Ø	INTERRUPTS	Ø	$\overline{\text{INTERRUPTS}}$
31	30 29	28	27		24	23	22 16	15	14 8	7	6 0

AND Gates 16-99:

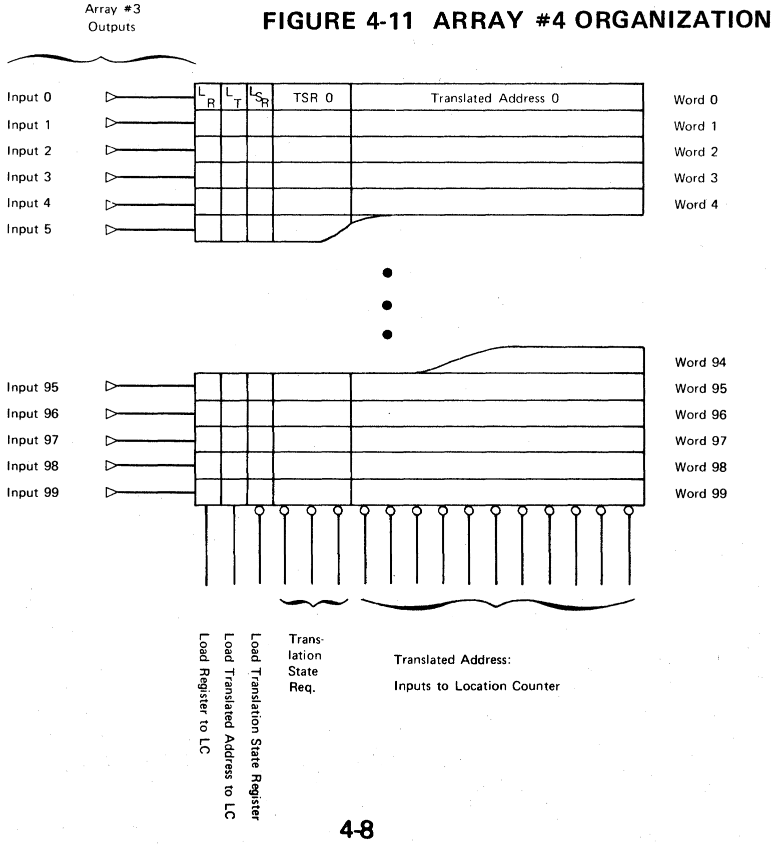
TSR	$\overline{\text{TSR}}$	Ø	Ø	Ø	Q	Ø	TRANSLATION CODE	TRANSLATION REGISTER	$\overline{\text{TRANSLATION REGISTER}}$
31	30 29	28	27		24	23	22 16	15 8	7 0

NOTE: Q= (($\overline{\text{TR14}}\cdot\overline{\text{TR13}}\cdot\overline{\text{TR12}}$) + (TR14 · TR13 · TR12))

4.2.5 ARRAY #4

Array #4 is a NORed array with 100 inputs which come from Array #3. The outputs generated by Array #4 include address data to be fed into the Location Counter, data to be fed into the Translation State Register, and three control lines. Two control lines determine whether the Location Counter will be loaded from the Return Register or the output of the translation array. The third control line determines whether or not the value presented to the Translation State Register will in fact be loaded into the Translation State Register. Figure 4-11 illustrates the orgainzation of this array. Note that not all the outputs are complemented. The two control signals LRA and LTA are true data.

FIGURE 4-11 ARRAY #4 ORGANIZATION



Array #4 will determine if any of its inputs are active. If they are, the word associated with the active input is placed on the output lines.

The array is programmed into terms of true data. Five hexadecimal digits are used to program each word.Transistors are placed in the array for 1 bits in the data specification. More than one active AND gate in Array 3 causes the outputs to OR.

Ø	L R A	L T A	L T S R	Ø		TRANSLATION ADDRESS
19	18	17	16	15	14 12	11 0

4.2.6 TRANSLATION STATE REGISTER

The Translation State Register serves two functions. One is to determine whether the Hi or the Lo order byte of the Translation Register is to be selected for input to Array #3; The other is to provide a four state feedback as an input of Array #3. The loading of the Translation State Register is controlled by an output (LTSR) of Array #4.

4.2.7 TRANSLATION REGISTER

The Translation Register holds the data presently serving as input to the Programmable Translation Array. It receives its inputs from the MicroInstruction Bus (MIB00 - MIB15) as a result of an LTR or IW instruction. Because of the fact that its output destination is 8 bits wide, only half of the Translation Register can be examined at any one time. The Translation State Register controls which half of the Translation Register is gated into the input of Array #3, its destination. It serves as a feedback term of information to the second half of the word.

Bit 2 of the TSR controls the selection of translation input from the TR. When this bit is a zero the upper byte of the TR is selected; when it is a one the lower byte is selected. TSR2 does not serve as an input to Array #3.

4.3 PROGRAMMABLE TRANSLATION ARRAY OPERATIONS

This section describes the operation of the Programmable Translation Array. The operation is based on the description of the components in the preceeding section that make up the Programmable Translation Array. The Programmable Translation Array operates as a loop. The beginning and end points of the loop can be considered as the Location Counter.

This loop is completed within one machine cycle. It is helpful to remember that the main components of the loop are:

- The Location Counter
- Arrays #1 and #2, which taken together perform a mapping of the Location Counter to a translation code.
- The translation register, which holds the macroinstruction undergoing the translation process.
- Arrays #3 and #4, which take the translation code and the macroinstruction presently undergoing translation and map them into an address.

This newly generated address can then point to the beginning of a macroroutine to interpret the macroinstruction presently accessed.

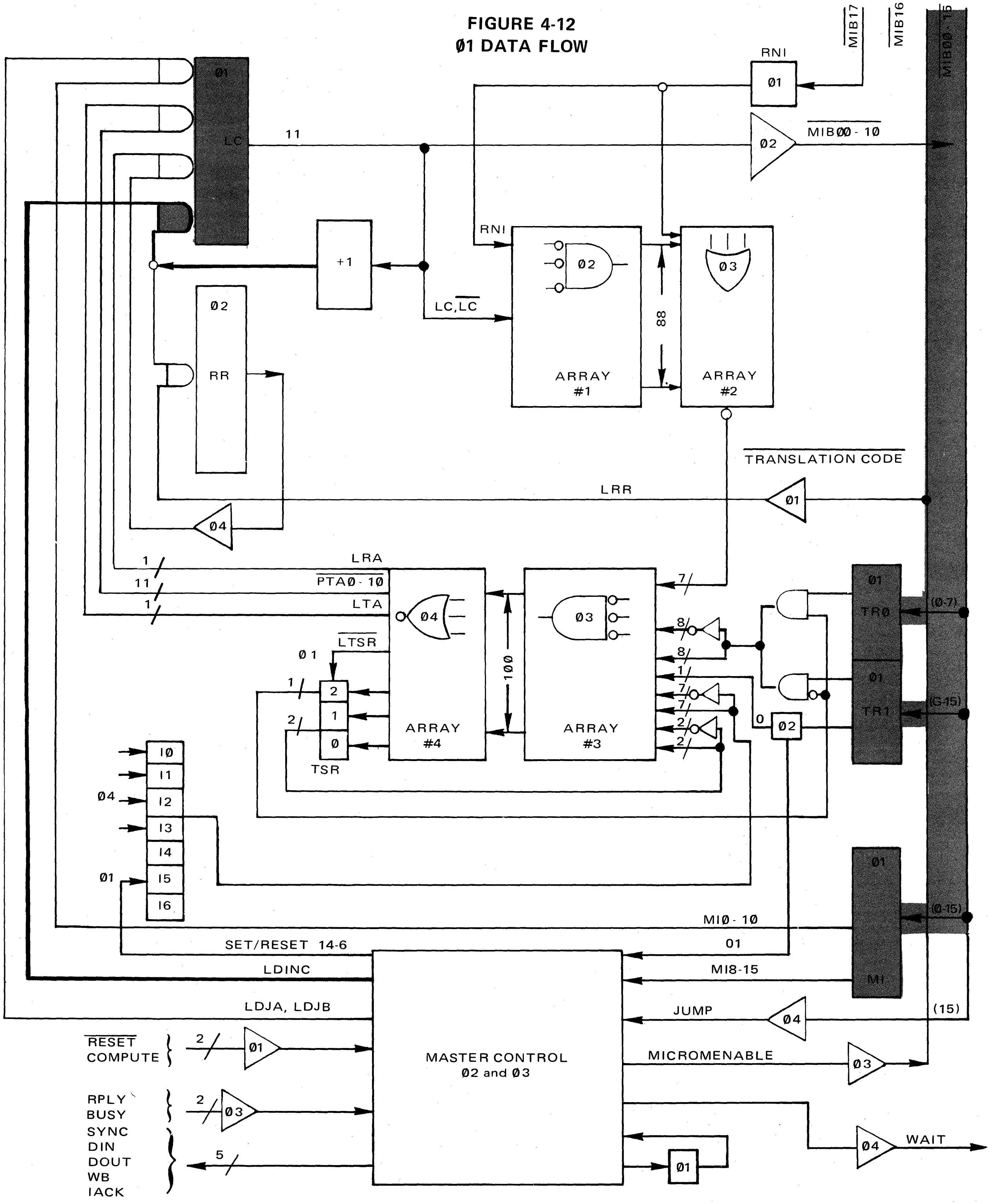
The following description can be best followed by reference to the attached diagrams which depict the flow of data in the Programmable Translation Array as a function of phase time clocks.

The operation of the Programmable Translation Array begins at Ø 1. During Ø 1 the location counter is loaded with its new value. The new value can come from any one of four sources. They are:

- From Array #4 of the Programmable Translation Array
- From the MI register
- From the incrementer
- From the Return Register

Figure 4-12 assumes, for the purposes of instruction, that the Location Counter is loaded from the incrementer. Concurrently, with the loading of the Location Counter at Ø1, the Translation Register is also being loaded if either a LTR or an IW instruction was executed.

FIGURE 4-12
Ø1 DATA FLOW



At the beginning of Ø2, (See Figure 4-13) the contents of the Location Counter are clocked out to the MIB bus (MIBOO - 10) and to Array #1. Both the true and complemented forms of the Location Counter are input to Array #1. Also input is the RNI line. Array #1 takes the input data word and determines whether or not it has a match. If it has a match, then the appropriate output is set at the end of Ø2. If there is no match, the translation proceeds no farther because no outputs are set.

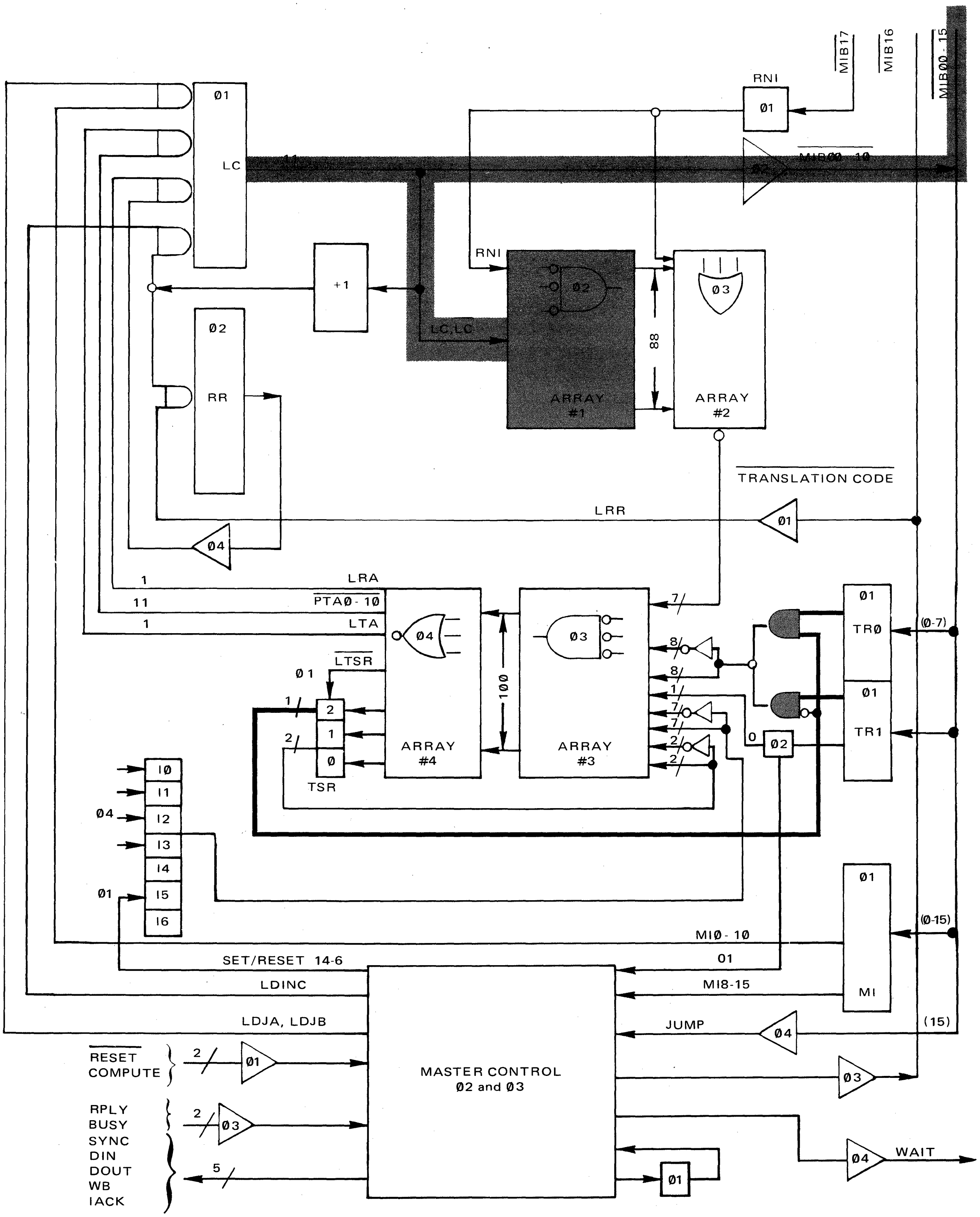


FIGURE 4-13
Ø2 DATA FLOW

At the beginning of Ø3, (See Figure 4-14), Array #2 samples the outputs from Array #1 and determines whether any are active. If there is a match, then the translation code is generated and fed to Array #3, also during Ø3.

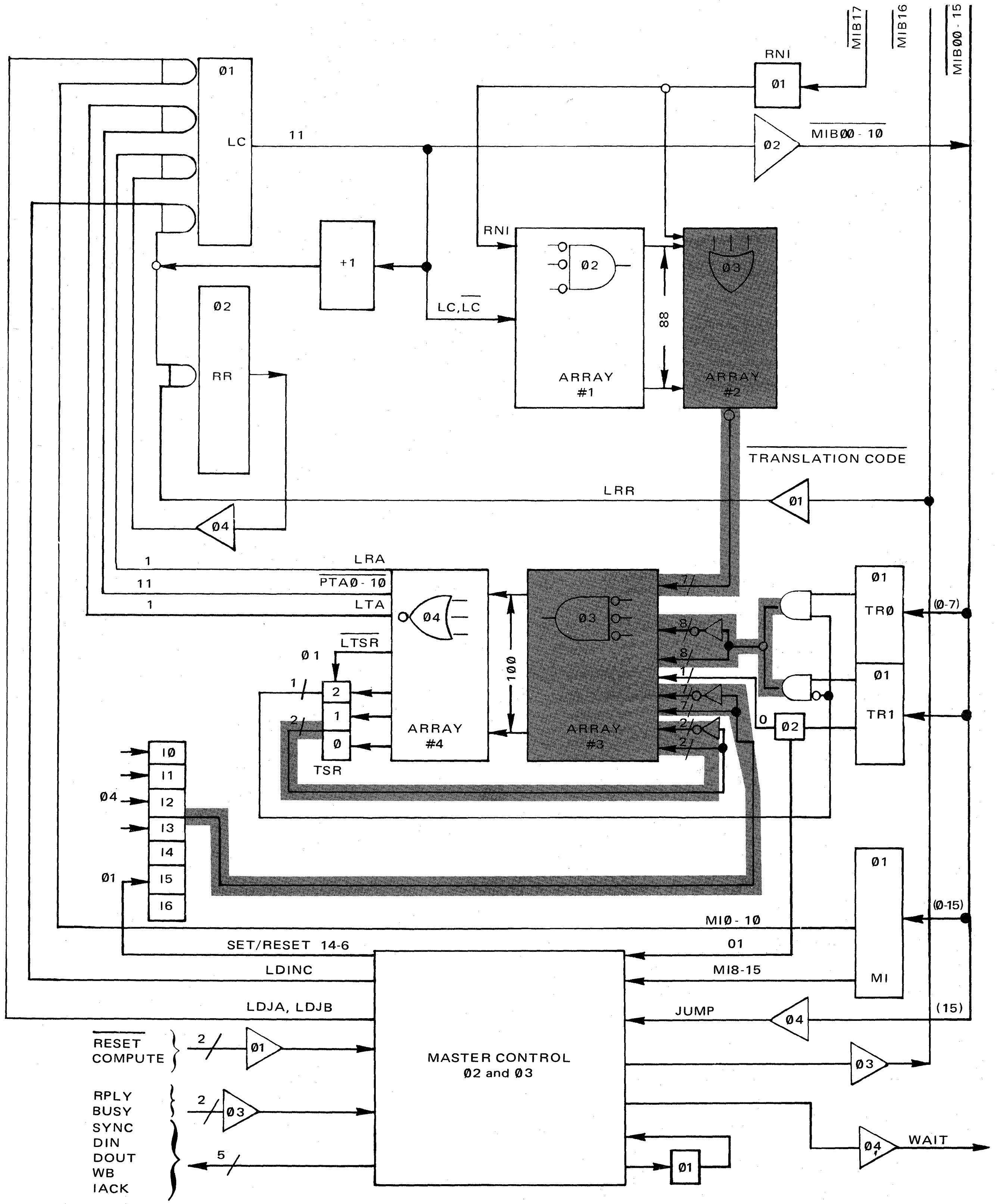


FIGURE 4-14
Ø3 DATA FLOW

During Ø3, the contents of the selected byte of the Translation Register are fed to Array #3 along with the contents of the interrupt register, the Translation State Register, and the Q bit. During the later portions of Ø3, Array #3 determines if there is a match and, if a match is found, at the end of Ø3 an output is generated.

This output is fed into Array #4 at the beginning of Ø 4. (See Figure 4-15). During Ø4, Array #4 determines whether there is an active input to it. If there is an active input, the contents of the word associated with the input are sent out at the end of Ø4. The Array #4 word may or may not cause the Translation State Register to be loaded, and may or may not cause the Location Counter to be loaded at the next occurance of Ø 1.

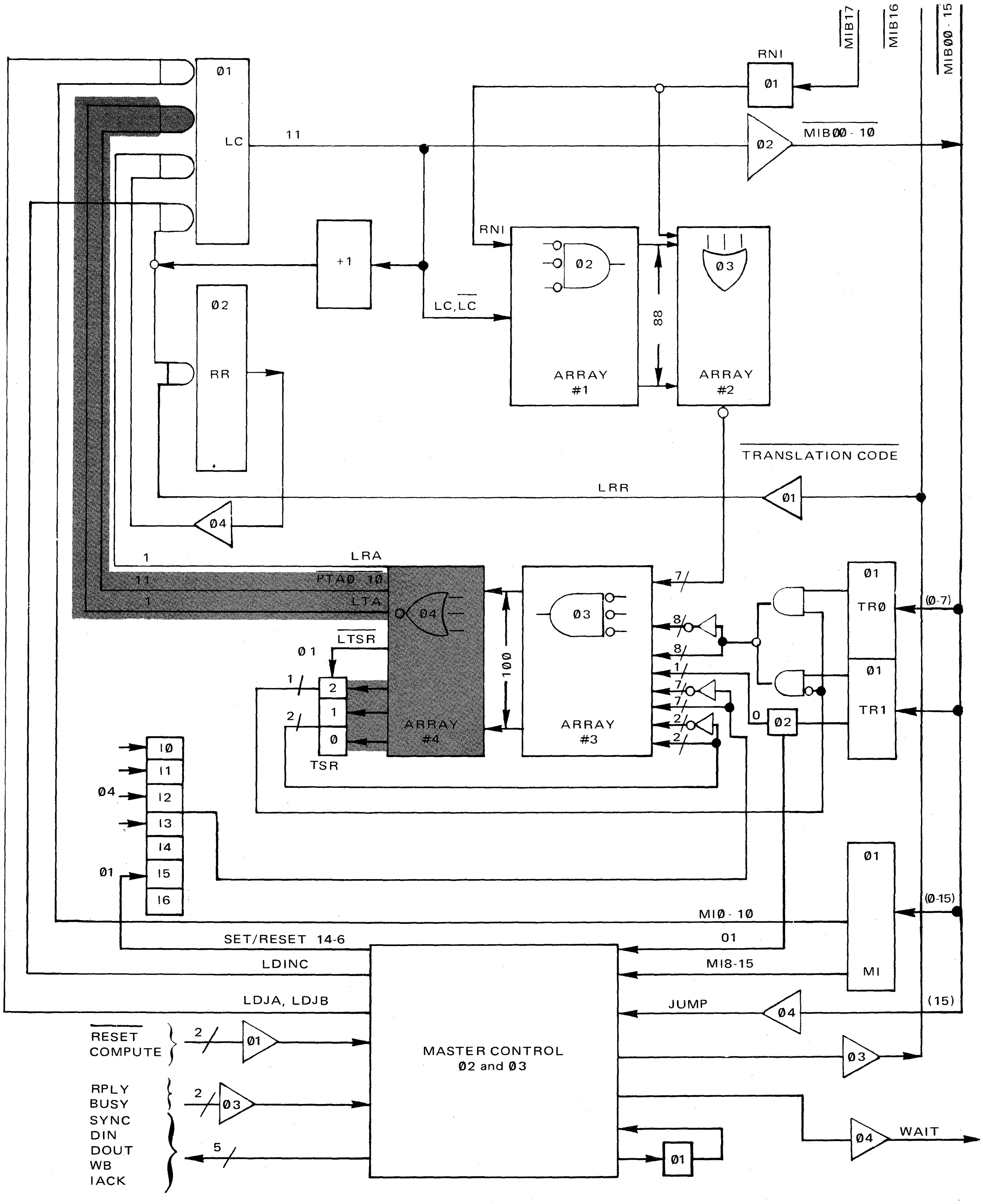


FIGURE 4-15
Ø4 DATA FLOW

SECTION V

MICRO INSTRUCTIONS

5.1 INSTRUCTION REPERTOIRE

This section contains descriptions of the microinstructions. With each description is a diagram showing the format of the instruction and its operation code, given in hexadecimal. Above each diagram is the instruction mnemonic operation code, the argument needed by the assembler, and the name of the instruction. Under each diagram is a description of the command and its timing in clock cycles. It should be noted that the descriptions of the instructions are rather perfunctory. For a more thorough description of the instructions from a programmer's standpoint, the user is invited to review the MCP1600 MICRO PROGRAMMER's MANUAL.

The purpose of this section then is to outline the structure of the instruction set as an aid to understanding the events on the two processor busses.

While the formats presented here describe 16 bits of the instruction word, be aware that the word is, in fact, 22 bits wide. The portion of the instruction described here control the operation of the processor. The other 6 bits are:

- bits 16 and 17 are involved with control of the Location Counter. Bit 16 determines whether or not the contents of the Return Register will be loaded into the Location Counter. Bit 17 determines whether or not a Read Next Instruction translation will be invoked.

  These two options are available on all instructions.

- bits 18 through 21 have nothing to do with control of the MCP1600 system. They are available for the user to program as TTL levels. These bits are available on all instructions.

5.2 INSTRUCTION TYPES

All of the MCP1600 microinstructions are 22 bits long. Of the 22 bits, the 16 least significant effect the operation of the CPU. The purposes of the other 6 bits are covered in either the Programmed Translation Array description or the Micro Instruction Bus description.

The instructions for the MCP1600 are divided into four classes as follows:

JUMP FORMAT

OP	R	Address
15 12	11	10 0

This format provides an 11-bit Control ROM address. It causes an unconditional jump to any location in the permissable MICROM address space by always asserting the jump control line (JXX on the Data Chip or MIB 15 on the Micro Instruction Bus). The unconditional jump instruction is the only one that utilizes this format.

Unconditional jump instructions always take two clock cycles; one to decode the instruction, the second to load the Location Counter on the Control Chip.

A special case of this format is the Return From Subroutine instruction, covered later.

CONDITIONAL JUMP FORMAT

OP	C	ADDRESS
15 12	11 8	7 0

This format provides a Jump address within a page. This instruction is substantially the same as the Unconditional Jump format, with exception of the reduced address space. It is also a two cycle instruction, with the jump decision (based upon the contents of the C Field) being made by the jump decode logic on the Data Chip during the first cycle. This decision is communicated to the Control Chip via MIB15 (JXX on the Data Chip). During the second cycle the jump is affected.

LITERAL FORMAT

OP	Literal	a
15 12	11 4	3 0

This format provides 8-bit literal data. The "a" field usually specifies a file register. These instructions invoke the 'literal path' to fetch data around the register file for the 'b' input to the ALU. All of these instructions execute in one clock cycle.

REGISTER FORMAT

OP	b	a
15 8	7 4	3 0

This format is by far the predominant one. In it, 'b' and 'a' are register file designators. Generally, depending upon the operation, there can be two meanings to the register designators:

$$\text{'b'}\ \xrightarrow{\text{(operation)}}\ \text{'a'}$$

as, for instance, in a shift or move instruction

$$\text{'b' (operation) 'a'} \longrightarrow \text{'a'}$$

as, for instance, in an arithmetic instruction.

'b' and 'a' can specify either a single byte data quantum or a word (double byte) data quantum. In this latter case, the designators usually point to the even addressed member of a register pair. While they may both point to an odd addressed pair, note carefully the instructions description for the effect.

If the designators apply to a word data quantum, the instruction takes two cycles to execute. The following sections describe MCP1600 instructions by classes of function.

5.3 MCP 1600 MICRO-INSTRUCTION SET

The following micro-instructions are contained in the MCP 1600. The symbology used is listed in the table below:

Symbols	Meaning
→	is transferred to
( )	contents of location or register
Λ	Logic Product (AND)
V	Inclusive "or"
⊻	Exclusive "or"
Rx:Ry	Forms extended register Ry = LSB, Rx = MSB
Ra	The register specified by the micro-instruction "a" field
Rb	The register specified by the micro-instruction "b" field
Flag Setting	Designates no category
X	Don't care condition
Ø	Flag cleared (set to Ø)
1	Flag set
–	Flag not affected
*	Set according to function (see flag description in Section 5.3.2)

5.3.1 TABLE OF MICRO-INSTRUCTIONS

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Jump	Ø(Ø)	JMP	2	(MIR1Ø-ØØ)→(LC) Jump unconditionally to microm location specified by MIR1Ø-ØØ
Return from Subroutine	Ø(1)	RFS	2	(RR)→(LC) Restore return address from subroutines
Jump if ƵB False	1Ø	JƵBF	2	If ZB=Ø, (MIR Ø7-ØØ)→(LC7-Ø) jump conditionally within microm page
Jump if ƵB True	11	JƵBT	2	If ƵB=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C8 False	12	JC8F	2	If C8=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C8 True	13	JC8T	2	If C8=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if ICS False	14	JIF	2	If ICS=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if ICS True	15	JIT	2	If ICS=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if NB False	16	JNBF	2	If NB=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if NB True	17	JNBT	2	If NB-1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if Ƶ False	18	JƵF	2	If Ƶ=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if Ƶ True	19	JƵT	2	If Ƶ=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C False	1A	JCF	2	If C=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C True	1B	JCT	2	If C=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if V False	1C	JVF	2	If V=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if V True	1D	JVT	2	If V=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if N False	1E	JNF	2	If N=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if N True	1F	JNT	2	If N=1, (MIRØ7-ØØ)→(LC7-Ø)
Add Literal	2	AL	1	(Ra)+Literal→Ra the 8 bit constant is added to Ra
Compare Literal	3	CL	1	(Ra)-Literal. The 8 bit result of the literal is compared against Ra and the approprate flags set.
And Literal	4	NL	1	(Ra)∧Literal→(Ra) The 8 bit result of a logical product of Ra and the literal are coded into Ra.
Test Literal	5	TL	1	(Ra)∧ Literal The 8 bit result of the logical product of Ra and the literal set condition Flags. The contents of Ra are unaffected.
Load Literal	6	LL	1	Literal→(Ra). The 8 bit literal is loaded into Ra.
Reset Interrupts	7Ø	RI	1	The three software interrupts I4, I5, and I6 are reset as indicated in the instruction's b field.
Set Interrupts	7I	SI	1	The three software interrupts I4, I5, and I6 are set as indicated in the instruction's b field.
Copy Condition Flags	72	CCF	1	Flags→(Ra) The four condition flags and four ALU status flags are loaded in Ra
Load Condition Flags	73	LCF	1	(Ra)→Flags. The contents of Ra are transferred to the ALU status flags unconditionally and to the condition flags under control of the b field.
Reset TSR	74	RTSR	1	Ø→(TSR). The three TSR bits in the control chip are reset.
Load G Low	75	LGL	1	(Ra) 2-Ø→(G). The three least significant bits of Ra are loaded into the G register.
Conditionally Increment Byte	76	CIB	1	If C8 is set, (Ra)+I → (Ra)
Conditionally Decrement Byte	77	CDB	I	If C8 is set, (Ra)-I → (Ra)

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Move Byte	8Ø/81	MB	1	(Rb)→(Ra). The 8 bit contents of Rb are transferred to Ra. Rb is unaffected.
Move Word	82/83	MW	2	(Rb)→(Ra). The 16 bit contents ofRb+I:Rb are transferred to Ra+1:Ra. Rb+I and Rb are unaffected.
Conditionally Move Byte	84/85	CMB	1	(Rb)→(Ra), if C=I. The 8 bit contents of Rb are transferred to Ra if the carry flag is set from a previous operation.
Conditionally Move Word	86/87	CMW	2	(Rb)→(Ra), if C=I. The 16 bit contents of Rb+I:Rb are transferred into Ra+I: Ra if the carry flag is set from a previous operation.
Shift Left Byte with Carry	88/89	SLBC	1	(Rbm)→(Ram+I), (carry)→(Raø). The 8 bit contents of Rb are shifted left one bit and loaded into Ra. The contents of the carry flag is inserted into the vacated low position of Ra.
Shift Left Word with Carry	8A/8B	SLWC	2	(Rbm)→(Ram+I), (carry)→(Ra ø). The 16 bit contents of Rb+I:Rb are shifted left one bit and loaded into Ra+I:Ra. The contents of the carry flag is inserted into the vacated low position of Ra.
Shift Left Byte	8C/8D	SLB	1	(Rbm)→(Ram+I) The 8 bit contents of Rb are shifted left one bit and loaded into Ra.
Increment Byte by 1	9Ø/9I	ICBI	1	(Rb)+I→(Ra). The 8 bit contents of Rb are incremented by one and the result is transferred to Ra.
Increment Word by 1	92/93	ICWI	2	(Rb)+I→(Ra). The 16 bit contents of Rb+I:Rb are incremented by one and the result transferred to Ra+I:Ra.
Increment Byte by 2	94/95	ICB2	1	(Rb)+2→(Ra). The 8 bit contents of Rb are incremented by two and the result transferred to Ra.
Increment Word by 2	96/97	ICW2	2	(Rb)+2→(Ra). The 16 bit contents of Rb+I:Rb are incremented by two and the result transferred to Ra+I:Ra.
Twos Complement Byte	98/99	TCB	1	($\overline{Rb}$)+I→Ra. The 8 bit contents of Rb are two's complemented and transferred to Ra.
Twos Complement Word	9A/9B	TCW	2	($\overline{Rb}$)+I→Ra. The 16 bit contents of Rb+I Rb are two's complemented and transferred to Ra+I:Ra.
Ones Complement Byte	9C/9D	OCB	1	($\overline{Rb}$)→Ra. The 8 bit contents of Rb are one's complemented and transferred to Ra.
One's Complement Word	9E/9F	OCW	2	($\overline{Rb}$)→Ra. The 16 bit contents of Rb+I: Rb are one's complemented and transferred to Ra+I:Ra.
Add Byte	AØ/AI	AB	1	(Rb)+(Ra)→(Ra). The 8 bit contents of Rb are added to Ra and loaded into Ra.
Add Word	A2/A3	AW	2	(Rb)+(Ra)→(Ra). The 16 bit contents of Rb+I:Rb are added to Ra+I:Ra and loaded into Ra+I:Ra.
Conditionally Add Byte	A4/A5	CAB	1	(Rb)+(Ra)→(Ra), if C=I. The 8 bit contents of Rb are added to Ra and the result is transferred to Ra if the carry flag is set from a previous operation.
Conditionally Add Word	A6/A7	CAW	2	(Rb)+(Ra)→(Ra), if C=I. The 16 bit contents of Rb+1:Rb are added to Ra+I:Ra and the result is transferred to Ra+I;Ra if the carry flag is set from a previous operation.
Add Byte with Carry	A8/A9	ABC	1	(Rb)+(Ra)+C→(Ra). The 8 bit sum of the contents of Rb plus the carry flag are added to Ra and the result is transferred to Ra.

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Add Word with Carry	AA/AB	AWC	2	$(Rb)+(Ra)+C\rightarrow(Ra)$. The 16 bit sum of the contents of Rb+1:Rb plus the carry flag are added to Ra+1:Ra and the result is transferred to Ra+1:Ra.
Conditionally Add Digits	AC	CAD	1	$(Rb)_{3\text{-}o}+(Ra)_{3\text{-}o}\rightarrow(Ra)_{3\text{-}o}$, if C4=0 $(Rb)_{7\text{-}4}+(Ra)_{7\text{-}o}\rightarrow(Ra)_{7\text{-}4}$, if C8=0 The 4 least significant and/or 4 most significant bits of Rb are added to their corresponding bits in Ra if C4 or C8 are reset respectively from a previous operation. This allows for decimal arithmetic corrections.
Conditionally Add Word on Indirect Condition Status	AE/AF	CAWI	2	$(Rb)+(Ra)\rightarrow(Ra)$, if ICS=I. The 16 bit contents of Rb+I:Rb are added to Ra+I:Ra and the results transferred to Ra+I:Ra if the ICS code is set from a prior operation. This instruction allows address displacement addition on Branch conditions.
Subtract Byte	BØ/BI	SB	1	$(Ra)-(Rb)\rightarrow(Ra)$. The 8 bit contents of Rb are subtracted from Ra and the result loaded into Ra.
Subtract Word	B2/B3	SW	2	$(Ra)-(Rb)\rightarrow(Ra)$. The 16 bit contents of Rb+I:Rb are subtracted from Ra+I:Ra and the results loaded into Ra+I:Ra.
Compare Byte	B4/B5	CB	1	$(Ra)-(Rb)$. The 8 bit difference between the contents of Ra and Rb is used to set the status flags at all times and the condition flags if OP code B5 is selected Registers Ra and Rb are not changed.
Compare Word	B6/B7	CW	2	$(Ra)-(Rb)$. The 16 bit difference between the contents of Ra+I:Ra and Rb+I:Rb are used to set the status flags at all times and the condition flags if OP Code B7 is selected. Registers Ra+I:Ra and Rb+I:Rb are not changed.
Subtract Byte with Carry	B8/B9	SBC	1	$(Ra)-(Rb)-C\rightarrow(Ra)$. The difference of the 8 bit contents of Rb subtracted from Ra minus the contents of C flag is loaded into Ra.
Subtract Word with Carry	BA/BB	SWC	2	$(Ra)-Rb)-C\rightarrow(Ra)$. The difference of the 16 bit contents of Rb+I:Rb subtracted from Ra+I:Ra minus the contents of the C flag is loaded into Ra+I:Ra.
Decrement Byte by I	BC/BD	DBI	1	$(Rb)-1\rightarrow Ra$. The 8 bit contents of Rb minus one are loaded into Ra.
Decrement Word by I	BE/BF	DWI	2	$(Rb)-I\rightarrow Ra$. The 16 bit contents of Rb+I:Rb minus one are loaded into Ra+I:Ra.
And Byte	CØ/CI	NB	1	$(Rb)\wedge(Ra)\rightarrow(Ra)$. The 8 bit logical product of Rb and Ra is loaded into Ra.
And Word	C2/C3	NW	2	$(Rb)\wedge(Ra)\rightarrow(Ra)$. The 16 bit logical product of Rb+I:Rb and Ra+I:Ra is loaded into Ra+I:Ra.
Test Byte	C4/C5	TB	1	$(Rb)\wedge(Ra)$. The 8 bit logical product of Rb and Ra sets the status flags at all times and the condition flags if OP code C5 is selected. Ra and Rb are unchanged.
Test Word	C6/C7	TW	2	$(Rb)\wedge(Ra)$. The 16 bit logical product of Rb+I:Rb and Ra+I:Ra sets the status flags at all times and the condition flags if OP code C7 is selected. Ra and Rb are unchanged.
Or Byte	C8/C9	OB	1	$(Rb)\vee(Ra)\rightarrow(Ra)$. The 8 bit logical OR operation is performed between the contents of Ra and Rb and the result are transferred to Ra.

Instruction	OP Code	Mnemonie	Microcycles	Description of Operation
Or Word	CA/CB	OW	2	(Rb)∨(Ra)→(Ra). The 16 bit logical OR operation is performed between the contents of Ra+I:Ra and Rb+I:Rb and the results are transferred to Ra+I:Ra.
Exclusive-Or Byte	CC/CD	XB	1	(Rb)⊻(Ra)→Ra. The 8 bit logical exclusive OR operation is performed between the contents of Rb and Ra and the result is transferred to Ra.
Exclusive-Or Word	CE/CF	XW	2	(Rb)⊻(Ra)→Ra. The 16 bit logical exclusive OR operation is performed between the contents of Rb+I:Rb and Ra+I:Ra and the result is transferred to Ra+I:Ra.
And Complement Byte	DØ/DI	NCB	1	$(\overline{Rb})$∧(Ra)→(Ra). The 8 bit logical product of the inverse of Rb and the contents of Ra is loaded into Ra.
And Complement Word	D2/D3	NCW	2	$(\overline{Rb})$∧(Ra)→(Ra). The 16 bit logical product of the inverse of Rb+I:Rb and the contents of Ra+I:Ra is loaded into Ra+I:Ra.
Shift Right Byte with Carry	D8/D9	SRBC	1	(Rbm+I)→(Ram), (carry)→(Ra7). The 8 bit contents of Rb are shifted right one bit and loaded into Ra. The carry flag is inserted into the high order position of Ra.
Shift Right Word with Carry	DA/DB	SRWC	2	(Rbm+I)→(Ram), (carry)→(Ra15). The 16 bit contents of Rb+I:Rb are shifted right one bit and loaded into Ra+I:Ra. The carry flag is inserted into the high order position of Ra+I.
Shift Right Byte	DC/DD	SRB	1	(Rbm+I)→(Ram). The 8 bit contents of Rb are, shifted right one bit and loaded into Ra.
Shift Right Word	DE/DF	SRW	2	(Rbm+I)→(Ram). The 16 bit contents of Rb+I:Rb are shifted right one bit and loaded into Ra+I:Ra.
Input Byte	EØ/EI	IB	1 (min)	(DAL)→(Ra). An 8 bit byte on the DAL is loaded into the specified Ra. The b field in this instruction selects read or read-modify-Write operation and selects upper (Bit 15-8) or lower (Bits 7-0) as the byte to be input from the Dal lines.
Input Word	E2/E3	IW	2 (min)	(DAL)→(Ra). The 16 bit word contained on the DAL is loaded into Ra+I:Ra. The b field in this instruction selects read or read-modify-write operation, selectively updates the G register, and selectively sets the ICS bit.
Input Status Byte	E4/E5	ISB	1	(DAL)→(Ra). The 8 bit byte from the DAL line, as specified by the b fold is input to register Ra, regardless of the state of REPLY or BUSY signal.
Input Status Word	E6/E7	ISW	2	(DAL)→(Ra). The 16 bit word from the DAL line is loaded into Ra+I:Ra regardless of the state of the REPLY or BUSY signal.
Modify Instruction	EC/ED	MI	1	(MIB)∨(Rb:Ra). The 16 bit contents of registers Rb:Ra are ORed with the contents of next microm micro-instruction on the MIB lines to modify any or all of the next micro-instruction. This instruction can be used to make on-line changes to micro-program flow.

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Load Translation Register	EE/EF	LTR	2	(Rb:Ra)→(TR). The 16 bit contents of registers Rb:Ra are transferred to the Translation Register on the chip. This allows a translation of resultant data into a micro-instruction.
Read and Increment Byte by I	FØ	RIBI	1	(Rb:Ra)→DAL (Ra)+I→(Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The contents of Ra are incremented by one. Rb is unchanged.
Write and Increment Byte by I	FI	WIBI	1	(Rb:Ra)→DAL, (Ra)+I→(Ra). The 16 bit address located in Rb;Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The contents of Ra are incremented by one. Rb is unchanged.
Read and Increment Word by I	F2	RIWI	2	(Rb:Ra)→DAL, (Ra+I:Ra)+I→(Ra+I:Ra). The 16 bit address located in Rb;Ra is transferred to the DAL lines and a DATA READ operation is initiated. The contents of Ra+I:Ra are incremented by one.
Write and Increment Word by I	F3	WIWI	2	(Rb:Ra)→DAL, (Ra+I:Ra)+I→(Ra+I:Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The contents of Ra+I:Ra are incremented by one.
Read and Increment Byte by 2	F4	RIB2	1	(Rb:Ra)→DAL, (Ra)+2→(Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The contents of Ra are incremented by 2. Rb remains unchanged.
Write and Increment Byte by 2	F5	WIB2	1	(Rb:Ra)→DAL, (Ra)+2→(Ra). The 16 bit address located in Rb:Ra is transferred to the DAL line and a DATA WRITE operation is initiated. The contents of Ra are incremented by 2. Rb is not changed.
Read and Increment Word by 2	F6	RIW2	2	(Rb:Ra)→DAL, (Ra+I:Ra)+2→(Ra+I:Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The content of Ra+I:Ra are incremented by 2.
Write and Increment Word by 2	F7	WIW2	2	(Rb:Ra)→DAL, (Ra+I:Ra)+2→(Ra+I:Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The contents of Ra+I:Ra are incremented by 2.
Read	F8	R	1	(Rb:Ra)→DAL. The 16 bit address located in Rb:Ra is transferred to the DAL line and a DATA READ operation is initiated.
Write	F9	W	1	(Rb:Ra)→DAL. The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated.
Read Acknowledge	FA	RA	1	(Rb:Ra)→ DAL, 1→IACK. The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The interrupt acknowledge line is set high.
Write Acknowledge	FB	WA	1	(Rb:Ra)→DAL, 1→IACK. The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The Interrupt Acknowledge line is set high.

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Output Byte	FC	OB	1 (min.)	(Rb:Ra)→DAL, 1→DOUT. The 16 bit data contents of Rb:Ra are transferred to the DAL lines. Registers Ra and Rb are unchanged. The DATA OUT signal is activated. This instruction is completed when the REPLY signal is received from the addressed unit. For byte operations Rb must equal Ra so that the same byte is placed in both 8 bit positions on the DAL.
Output Word	FD	OW	1 (min.)	(Rb:Ra)→DAL, 1→DOUT. The 16 bit data contents of Rb:Ra are transferred to the DAL lines. The DATA OUT signal is activated. Registers Ra and Rb are unchanged. This instruction is complete when the REPLY signal is received from the addressed unit.
Output Status	FE	OS	1	(Rb:Ra)→DAL. The 16 bit contents of registers Rb:Ra are transferred to the DAL lines and take place regardless of the state of the REPLY or BUSY signal. DOUT is not activated and SYNC is not terminated. Registers Rb and Ra are unchanged.
No Operation	FF	NOP	1	This instruction causes no operation.

5.3.2 SUMMARY OF MICRO-INSTRUCTION AND STATUS FLAGS

					ALU Status Flags				Condition Flags			
Mnemonic	OP Code			Microcycles	NB	ZB	C4	C8	N	Ƶ	V	C
A. Jump Format	Bit 15-12	II	1Ø-Ø									
JMP	Ø	Ø	ADDR	2	–	–	–	–	–	–	–	–
RFS	Ø	I	----------	2	–	–	–	–	–	–	–	–
B. Conditional Jump Format												
	Bit 15-12	II-8	7-0									
JƵBF	I	0	ADDR	2	–	–	–	–	–	–	–	–
JƵBT	1	1	ADDR	2	–	–	–	–	–	–	–	–
JC8F	1	2	ADDR	2	–	–	–	–	–	–	–	–
JC8T	1	3	ADDR	2	–	–	–	–	–	–	–	–
JIF	1	4	ADDR	2	–	–	–	–	–	–	–	–
JIT	1	5	ADDR	2	–	–	–	–	–	–	–	–
JNBF	1	6	ADDR	2	–	–	–	–	–	–	–	–
JNBT	1	7	ADDR	2	–	–	–	–	–	–	–	–
JƵF	1	8	ADDR	2	–	–	–	–	–	–	–	–
JƵT	1	9	ADDR	2	–	–	–	–	–	–	–	–
JCF	1	A	ADDR	2	–	–	–	–	–	–	–	–
JCT	1	B	ADDR	2	–	–	–	–	–	–	–	–
JVF	1	C	ADDR	2	–	–	–	–	–	–	–	–
JVT	1	D	ADDR	2	–	–	–	–	–	–	–	–
JNF	1	E	ADDR	2	–	–	–	–	–	–	–	–
JNT	1	F	ADDR	2	–	–	–	–	–	–	–	–
C. Literal Format												
	Bit 15-12	11-4	3-0									
AL	2	Lit	a	1	*	*	*	*	–	–	–	–
CL	3	Lit	a	1	*	*	–	–	–	–	–	–
NL	4	Lit	a	1	*	*	–	–	–	–	–	–
TL	5	Lit	a	1	*	*	–	–	–	–	–	–
LL	6	Lit	a	1	*	*	–	–	–	–	–	–

					ALU Status Flags				Conditon Flags			
Mnemonic	OP Code			Microcycles	NB	ZB	C4	C8	N	Z	V	C
D. Register Format	Bit 15-8	7-4	3-0									
RI	7Ø	b	x	1	-	-	-	-	-	-	-	-
SI	71	b	x	1	-	-	-	-	-	-	-	-
CCF	72	x	a	1	-	-	-	-	-	-	-	-
LCF	73	b	a	1	-	-	-	-	-	-	-	-
RTSR	74	x	x	1	-	-	-	-	-	-	-	-
LGL	75	x	a	1	-	-	-	-	-	-	-	-
CIB	76	x	a	1	*	*	*	*	-	-	-	-
CDB	77	x	a	1	*	*	*	*	-	-	-	-
MB	8Ø/81	b	a	1	*	*	-	-	*	*	0	-
MW	82/83	b	a	2	*	*	-	-	*	*	0	-
CMB	84/85	b	a	1	(*	*	-	-	*	*	0	-).C
CMW	86/87	b	a	2	(*	*	-	-	*	*	0	-).C
SLBC	88/89	b	a	1	*	*	*	*	*	*	*	*
SLWC	8A/8B	b	a	2	*	*	*	*	*	*	*	*
SLB	8C/8D	b	a	1	*	*	*	*	*	*	*	*
SLW	8E/8F	b	a	2	*	*	*	*	*	*	*	*
ICB1	9Ø/91	b	a	1	*	*	*	*	*	*	*	*
ICW1	92/93	b	a	2	*	*	*	*	*	*	*	*
ICB2	94/95	b	a	1	*	*	*	*	*	*	*	*
ICW2	96/97	b	a	2	*	*	*	*	*	*	*	*
TCB	98/99	b	a	1	*	*	*	*	*	*	*	*
TCW	9A/9B	b	a	2	*	*	*	*	*	*	*	*
OCB	9C/9D	b	a	1	*	*	0	0	*	*	0	1
OCW	9E/9F	b	a	2	*	*	0	0	*	*	0	1
AB	AØ/A1	b	a	1	*	*	*	*	*	*	*	*
AW	A2/A3	b	a	2	*	*	*	*	*	*	*	*
CAB	A4/A5	b	a	1	(*	*	*	*	*	*	*	*).C
CAW	A6/A7	b	a	2	(*	*	*	*	*	*	*	*).C
ABC	A8/A9	b	a	1	*	*	*	*	*	*	*	*
AWC	AA/AB	b	a	2	*	*	*	*	*	*	*	*
CAD	AC	b	a	1	*	*	*	*	*	*	*	*
CAWI	AE/AF	b	a	2	(*	*	*	*	*	*	*	*).ICS
SB	BØ/B1	b	a	1	*	*	*	*	*	*	*	*
SW	B2/B3	b	a	2	*	*	*	*	*	*	*	*
CB	B4/B5	b	a	1	*	*	*	*	*	*	*	*
SW	B6/B7	b	a	2	*	*	*	*	*	*	*	*
SBC	B8/B9	b	a	1	*	*	*	*	*	*	*	*
SWC	BA/BB	b	a	2	*	*	*	*	*	*	*	*
DBI	BC/BD	b	a	1	*	*	*	*	*	*	*	*
DWI	BE/BF	b	a	2	*	*	*	*	*	*	*	*
NB	CØ/CI	b	a	1	*	*	-	-	*	*	0	-
NW	C2/C3	b	a	2	*	*	-	-	*	*	0	-
TB	C4/C5	b	a	1	*	*	-	-	*	*	0	-
TW	C6/C7	b	a	2	*	*	-	-	*	*	0	-
ORB	C8/C9	b	a	1	*	*	-	-	*	*	0	-
ORW	CA/CB	b	a	2	*	*	-	-	*	*	0	-
XB	CC/CD	b	a	1	*	*	-	-	*	*	0	-
XW	CE/CF	b	a	2	*	*	-	-	*	*	0	-

					ALU Status Flags				Condition Flags			
Mnemonic	OP Code			Microcycles	NB	ZB	C4	C8	N	Ƶ	V	C
D. Register Format	Bit 15-8	7-4	3-0									
NCB	DØ/D1	b	a	1	*	*	—	-	*	*	0	—
NCW	D2/D3	b	a	2	*	*	—	—	*	*	0	—
SRBC	D8/D9	b	a	1	*	*	0	*	*	*	0	*
SRWC	DA/DB	b	a	2	*	*	0	*	**	*	0	*
SRB	DC/DD	b	a	1	*	*	0	*	*	*	0	*
SRW	DE/DF	b	a	2	*	*	0	*	*	*	0	*
IB	EØ/E1	b	a	1	*	*	—	—	*	*	0	—
IW	E2/E3	b	a	2	*	*	—	—	*	*	0	—
ISB	E4/E5	b	a	1	*	*	—	—	*	*	0	—
ISW	E6/E7	x	a	2	*	*	—	—	*	*	0	—
MI	EC/ED	b	a	1	—	—	—	—	—	—	—	—
LTR	EE/EF	b	a	2	—	—	—	—	—	—	—	—
RIB1	FØ	b	a	1	*	*	*	*	—	—	—	—
WIB1	F1	b	a	1	*	*	*	*	—	—	—	—
RIW1	F2	b	a	2	*	*	*	*	—	—	—	—
WIW1	F3	b	a	2	*	*	*	*	—	—	—	—
RIB2	F4	b	a	1	*	*	*	*	—	—	—	—
WIB2	F5	b	a	1	*	*	*	*	—	—	—	—
RIW2	F6	b	a	2	*	*	*	*	—	—	—	—
WIW2	F7	b	a	2	*	*	*	*	—	—	—	—
R	F8	b	a	1	—	—	—	—	—	—	—	—
W	F9	b	a	1	—	—	—	—	—	—	—	—
RA	FA	b	a	1	—	—	—	—	—	—	—	—
WA	FB	b	a	1	—	—	—	—	—	—	—	—
OB	FC	b	a	1	—	—	—	—	—	—	—	—
OW	FD	b	a	1	—	—	—	—	—	—	—	—
OS	FE	b	a	1	—	—	—	—	—	—	—	—
NOP	FF	x	x	1	—	—	—	—	—	—	—	—

SECTION VI

MICROINSTRUCTION BUS OPERATION

6.1 GENERAL

The Microinstruction Bus is responsible for interconnecting the three different circuits that make up a MCP1600 Microprocessor system. Connected to the Microinstruction Bus can be one CP1611B Data Chip, one CP1621B Control Chip, and as many as four CP1631B Microinstruction ROM Chips. There may also be user supplied TTL logic connected to the Microinstruction Bus in a fashion to be defined below.

The discussion below describes the 22 different lines on the Microinstruction Bus from the standpoint of each of the various types of devices attached to it and from the standpoint of user attached devices. Figure 6-1 illustrates the system interconnections.

6.2 MICROINSTRUCTION BUS OPERATION

The Microinstruction Bus consists of 22 lines. These lines are precharged by each of the microinstruction ROMs attached to the bus. Most generally, the lines are precharged high during Ø4 but MIB15 is precharged high during Ø3 and MIB16 is charged high during both Ø2 and Ø4. The microinstruction bus conveys its information by conditional discharge at the appropriate phase times. The microinstruction bus is an MOS compatible 4 phase bidirectional bus and data on the bus is in logical complement form.

The table on the next page (6-1) defines the meaning of each of the lines on the bus at each phase time on the bus. Note that some lines have more than one meaning, depending upon the phase time.

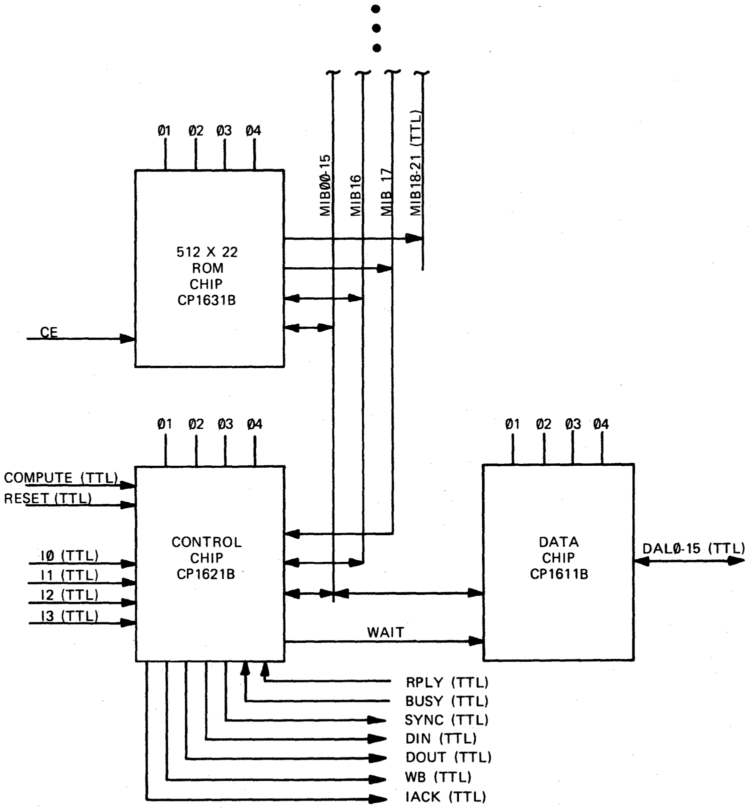


FIGURE 6-1. SYSTEM INTERCONNECTIONS MICROPROCESSOR SET

TABLE 6-1. MICROBUS TIMING

	Ø1	Ø2	Ø3	Ø4
MIB00-MIB10	– Microinstruction from ROM to Data Chip and Control – Second Cycle of 2-Cycle Instruction: Data from Data Chip to Control Chip.	– Address Data to MICROM	– Address must remain valid	– Precharge by MICROM
MIB11-MIB14	– Microinstruction from ROM to Data Chip and Control Chip – Second Cycle of 2-Cycle Instruction: Data from Data Chip to Control Chip.			– Precharge by MICROM
MIB15	– Microinstruction from ROM to Data Chip and Control Chip. – Second Cycle of 2-Cycle Instruction: Data from Data Chip to Control Chip.		– Precharge by MICROM	– Conditional Jump Results
MIB16	– Load Return Register	– Precharge by MICROM.	– Disable MICROM outputs at next Ø1	– Precharge by MICROM
MIB17	– RNI			– Precharge by MICROM
MIB18-MIB21	– TTL Outputs Valid			– Precharge by MICROM

6.3 MICROINSTRUCTION BUS ELEMENT DESCRIPTION

The MicroInstruction Bus is divided into seven partitions. These partitions can carry data in both directions. They assume different meanings at different points in the clock cycle of the processor set. This section will describe each element and its meaning as a function of the phase time in each clock cycle.

6.3.1 MIB00-MIB10

These lines serve to carry data bidirectionally between microinstruction ROMs and the Data and Control Chips. They are unconditionally precharged Hi at Ø4 by the MICROMS attached to the bus. At the following Ø1 they may have two different meanings, depending upon whether the Microinstruction presently being executed is a one cycle or a two cycle instruction.

If the Microinstruction executes one cycle, then the next Ø1 conveys the next microinstruction from the Microinstruction ROM to the Data Chip and Control Chip. If the Microinstruction takes two cycles, the second occurrence of Ø1 may result in data being transferred from the Data Chip to the Control Chip.

At Ø2, address data is transferred from the Location Counter in the Control Chip to the MICROM. At Ø3 the address data remains valid on the bus. The processor cycle is completed by an unconditional precharge of these lines by the MICROM at Ø4.

6.3.2 MIB11-MIB14

These lines serve to convey Microinstruction data from the Microinstruction ROM to the Data and Control Chips in much the same fashion as the preceeding lines do. These lines are unconditionally precharged Hi by the MICROM at Ø4. At Ø1 of a single cycle they are conditionally discharged to represent Microinstruction Data from the Microinstruction ROM to the Data Chip and the Control Chip.

If the preceeding Microinstruction was of the 2-cycle variety, the second occurrence of Ø1 on these lines can carry data from the Data Chip to the Control Chip. These lines have no other significance during Ø2 and Ø3.

6.3.3 MIB15

MIB15 is used to carry Microinstruction data from the ROM to the Data Chip and Control Chip and also to transfer the results of conditional jump tests from the Data Chip to the Control Chip.

This line is precharged Hi unconditionally by the Microinstruction ROM at Ø3. At Ø4 it can be conditionally discharged by the results of a conditional jump test.

At Ø1 this line conveys the microinstruction data from the ROM to the Data and Control Chip. In the case of a microinstruction whose execution takes two cycles, the second occurrence of Ø1 may serve to convey data from the Data Chip to the Control Chip. The contents of MIB15 are not significant at Ø2. The cycle is completed by an unconditional precharge of MIB15 at Ø3.

6.3.4 MIB16

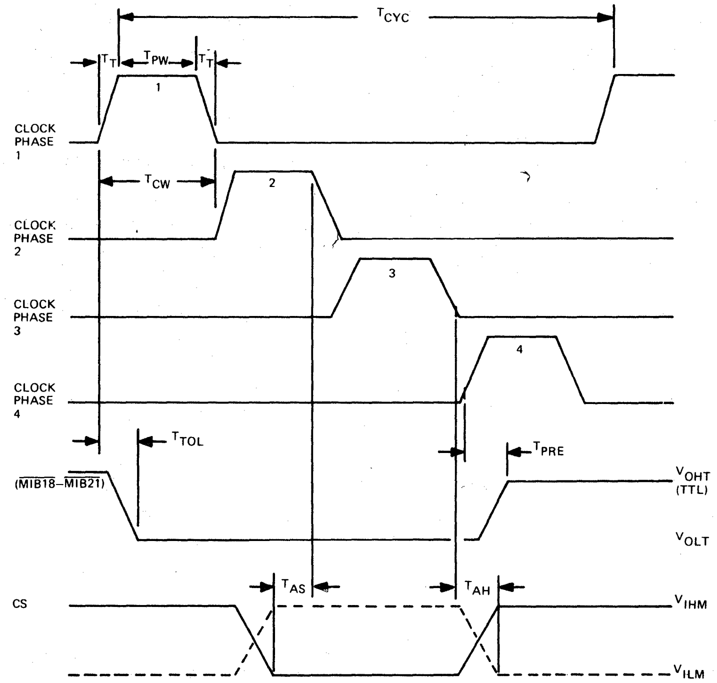
This control line conveys data from the ROM to the Control and Data Chips and from the Control Chip to the Microinstruction ROM. It is unconditionally precharged HI at both Ø2 and Ø4. at Ø1 following Ø4 it may conditionally discharge low by the MICROM in which case the signal is interpreted by the Control Chip as a command to load the subroutine Return Register with the incremented content of the Location Counter. On Ø2 the lines are unconditionally precharged Hi. At Ø3, the Control Chip may conditionally discharge this line. If it does so, this is an instruction to the selected MICROM (Microinstruction ROM) to disable its outputs at the next Ø1. In this fashion, 2 cycle instructions inhibit the transfer of new microinstructions from the MICROM to the Control Chip and the Data Chip.

6.3.5 MIB17

MIB17 has but one purpose. This is to convey the READ NEXT INSTRUCTION imperative from the MICROM to the Control Chip. This line is unconditionally precharged by the MICROMs attached to the Microinstruction Bus and conditionally discharged at Ø1. A discharge indicates that the RNI imperative is required.

6.3.6 MIB18-MIB21

These lines are not part of the Microinstruction Bus, in that they do not transfer data between the Microinstruction ROM and the Data Chip and the Control Chip. They are TTL level outputs capable of driving one TTL load per line. They represent four bits in the Microinstruction word, and are made valid at the same time as the other outputs of the MICROM, Ø1. The lines are unconditionally precharged Hi by the MICROM at Ø4 and conditionally discharged Low according to the contents of the word, at Ø1. The user may set these outputs in any fashion he chooses. They remain valid in the inclusive interval Ø1 to Ø3. Figure 6-2 illustrates their timing.



NOTE: TTL OUTPUTS ($\overline{\text{MIB18}}$–$\overline{\text{MIB21}}$) ARE UNCONDITIONALLY DRIVEN HIGH AT Ø4 AND CONDITIONALLY DRIVEN LOW AT Ø1 THIS CONDITIONAL LOW WILL BE VALID UNTIL THE NEXT Ø4.

FIGURE 6-2. TTL OUTPUT TIMING CP1631B

6.3.7 WAIT LINE

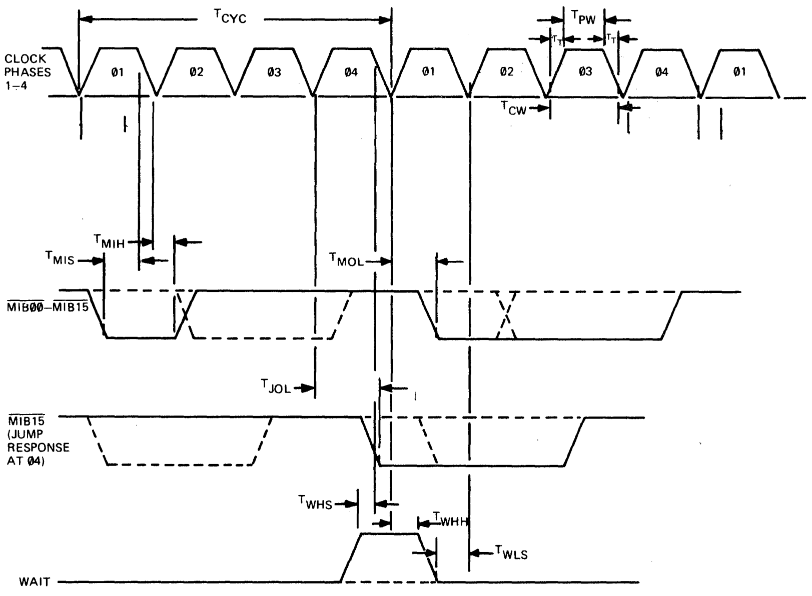
The WAIT Line from the Control Chip to the Data Chip establishes whether or not the Data Chip is in the RUN or WAIT mode. Whenever the WAIT control line is in the Low state, the Data Chip is in the RUN mode and the Microinstruction will be loaded into the MIR register and executed. This line is normally Low and must be driven Hi during Ø4 to cause the Data Chip to enter the WAIT state. It always returns to Low when the beginning edge of the Ø1 clock appears.

6.4 DATA CHIP

The CP1611B Data Chip interacts with a subset of the Microinstruction Bus. Lines MIB00-MIB15 are present at the Data Chip. They convey Microinstructions from the Microinstruction ROM to the Data Chip and they convey data from the Data Chip to the Control Chip. MIB15 also has a use as a control line conveying the results of conditional jump sets.

During the first Ø1 of a two cycle instruction, data is transferred from the microinstruction ROM into the MIR register of the Data Chip. The microinstruction bus is not sampled during Ø2 and Ø3. At Ø4 the Data Chip may conditionally discharge MIB15 to indicate the results of a conditional jump test. At the second occurrence of Ø1 of a two cycle instruction, the Data Chip may or may not transfer data to the Control Chip. If the two cycle instruction was of the Jump class or word operation class the Data Chip will not transfer data to the Control Chip. If the two cycle instruction was a LTR or IW instruction (with the appropriate bits in the control field properly set) then the Data Chip will transfer 16 bits of data into the Control Chip. This data will be gated into the Translation Register on the Control Chip.

During the execution of a one cycle microinstruction data is only transferred from the microinstruction ROM to the Data Chip. This transfer occurs at Ø1. Figure 6-3 illustrates the timing of the Microinstruction bus interface as seen by the Data Chip.



NOTE: SWITCHING TIMES ARE MEASURED AT 10% AND 90% OF SPECIFIED LEVELS.

FIGURE 6-3. MICROINSTRUCTION BUS TIMING

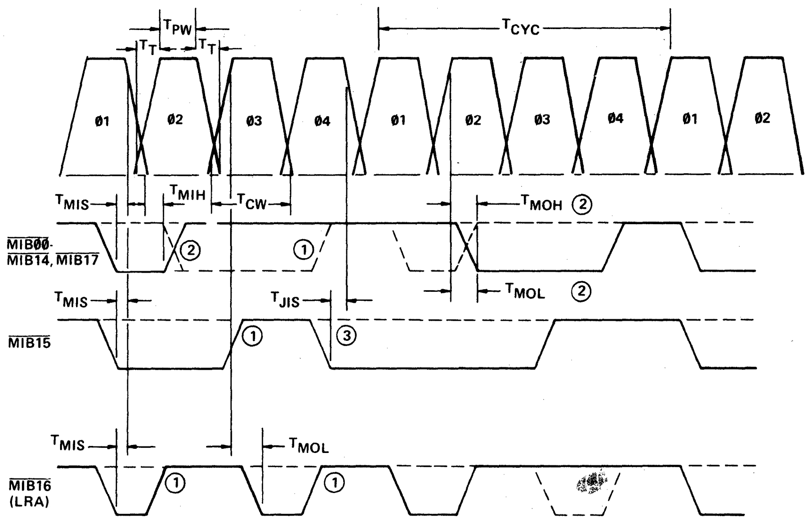
6.5 CONTROL CHIP

The CP1621B Control Chip interfaces with MIB00-MIB17 as the Data bus. Data can come to the Control Chip from both the Data Chip and the Microinstruction ROM. From the Microinstruction ROM the Control Chip can receive microinstructions into the MI register. There these microinstructions are used to drive the Master Control function and are also present in the event that the Microinstruction is of the jump variety and a jump is required. Data from the microinstruction ROM also is presented on MIB16, which controls whether or not the Return Register will be loaded with the incremented contents of the Location Counter. MIB17 from the Microinstruction ROM controls whether or not the RNI translation will be invoked. Data from the Data Chip to the Control Chip can be passed over MIB00-MIB15 as a result of an LTR instruction or an IW instruction. MIB15 is also used by the Data Chip to send the results to the Control Chip's Master Control function informing it of the results of a conditional jump test.

The Control Chip can generate a WAIT signal which goes to the Data Chip, and causes Data Chip operations to be suspended pending the completion of an I/O operation. The Control Chip also generates a signal placed on MIB16 which enables or disables the outputs of selected microinstruction ROM. This is used during the second cycle of the execution of two cycle instructions to avoid conflicts on the microinstruction bus. The timing of the microinstruction bus as seen from the Control Chip is illustrated in Figure 6-4.

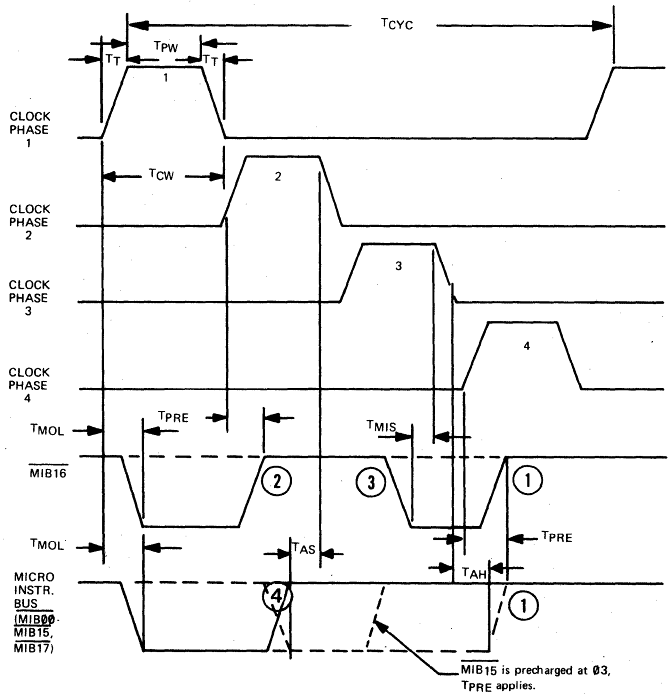
6.6 MICROINSTRUCTION ROM

The Microinstruction ROM interfaces with all 22 lines of the Microinstruction bus. It receives Microinstruction addresses on MIB00-MIB10 from the Location Counter in the Control Chip. It also receives an ENABLE/DISABLE signal on MIB16 from the Control Chip. It sends data on MIB00-MIB17 to the Control Chip and the Data Chip. MIB00-MIB15 are presented to both the Control Chip and the Data Chip. These 16 bits comprise microinstruction data for the Data Chip and the Control Chip to interpret. MIB16 is properly part of the microinstruction and controls whether or not the Return Register will be loaded with the incremented contents of the Location Counter. MIB17, also properly part of the microinstruction, controls whether or not the matrix represented by Arrays 1 and 2 will force a user specified State code that will cause the 'READ NEXT INSTRUCTIONS' translation to be performed. The interface of the microinstruction ROM to the microinstruction bus is illustrated in Figure 6-5.



NOTES:
1. BUS PRECHARGING IS PERFORMED BY THE CP1631B.
2. ADDRESS OUTPUT SWITCHING TIME (T_{MOH} AND T_{MOL}) APPLIES TO $\overline{\text{MIBØØ}}$-$\overline{\text{MIB1Ø}}$ ONLY.
3. JUMP RESPONSE FROM DATA CHIP CP1611B.
4. SWITCHING TIMES ARE MEASURED AT 10% AND 90% OF SPECIFIED LEVELS.

FIGURE 6-4. CONTROL CHIP MICROINSTRUCTION BUS TIMING



T_{CYC}
T_{PW}
T_T
T_T
1
CLOCK PHASE 1
T_{CW}
2
CLOCK PHASE 2
3
CLOCK PHASE 3
4
CLOCK PHASE 4
T_{MOL}
T_{PRE}
T_{MIS}
$\overline{\text{MIB16}}$
②
③
①
T_{MOL}
T_{AS}
T_{PRE}
T_{AH}
MICRO INSTR. BUS ($\overline{\text{MIBØØ}}$- $\overline{\text{MIB15}}$, $\overline{\text{MIB17}}$)
④
①
$\overline{\text{MIB15}}$ is precharged at Ø3, T_{PRE} applies.

NOTES:
1. MIB LINES EXCEPT $\overline{\text{MIB15}}$ ARE PRECHARGED HIGH AT PHASE 4.
2. $\overline{\text{MIB16}}$ IS ALSO PRECHARGED HIGH AT PHASE 2.
3. $\overline{\text{MIB16}}$ IS DISCHARGED TO DISABLE THE $\overline{\text{MIBØØ}}$-$\overline{\text{MIB15}}$ AND $\overline{\text{MIB18-MIB21}}$ OUTPUTS AT Ø1 FROM DISCHARGING LOW.
4. ADDRESS INPUT TRANSISTIONS OCCUR ONLY ON $\overline{\text{MIBØØ}}$-$\overline{\text{MIB1Ø}}$ LINES AND CHIP SELECT.
5. SWITCHING TIMES ARE MEASURED AT 10% AND 90% OF SPECIFIED LEVELS.

FIGURE 6-5. MICROINSTRUCTION BUS TIMING CP1631B

SECTION VII

DATA ACCESS

7.1 GENERAL

Communications between the MCP 1600 system and attached peripheral devices or memory is conducted via a path called the Data Access Bus. The Data Access Bus consists of 16 data lines (DAL00-DAL15), 7 control lines and 4 interrupt lines. The operation of these elements is controlled by the Input/Output class of instructions.

The Data Access provides a single 16 bit bidirectional path to and from the processor for transfer of data and addresses. The user may configure a number of different types of bus arrangements with the data access. Types such as:

- Common data and address
- Separate data and address
- Separate input and output buses

are easily implemented.

The Data Access Bus provides for 16 bit address and either 8 or 16 bit data (addresses are conventionally deemed to be byte addresses) and provide a maximum addressing capability of 65k bytes.

In accessing data, no distinction is made between memory and peripheral units or between instructions, data, control, or status. Each byte or word of information, regardless of function, is assigned an address and is referenced by means of this address.

In essence, then the Data Access Bus can be conceived to consist of the following elements:

- Data Lines (DAL00-DAL15)
- Control Lines (SYNC, REPLY, DATA-OUT or DOUT, DATA-IN or DIN, WRITE/BYTE, IACK, BUSY)
- Processor Control Lines (I0, I1, I2, I3, COMPUTE, RESET)

The remainder of this section describes each of these elements of the data bus and their interactions and timings.

7.2 INPUT/OUTPUT INSTRUCTIONS

This section breaks the I/O Instruction class into 5 classes. These classes are:

- Control Instruction consisting of the READ and WRITE INSTRUCTIONS that serve mainly to operate the Control Lines on the Data Access and to provide addresses.
- Data Transfer Instruction, consisting of the INPUT and OUTPUT instructions which serve to pass data along the Data Access lines.
- ACKNOWLEDGE instructions which serve to respond to interrupts.

Figure 7-1 illustrates the condition testing performed prior to executing the Input/Output instructions. The execute function is described in the following sections.

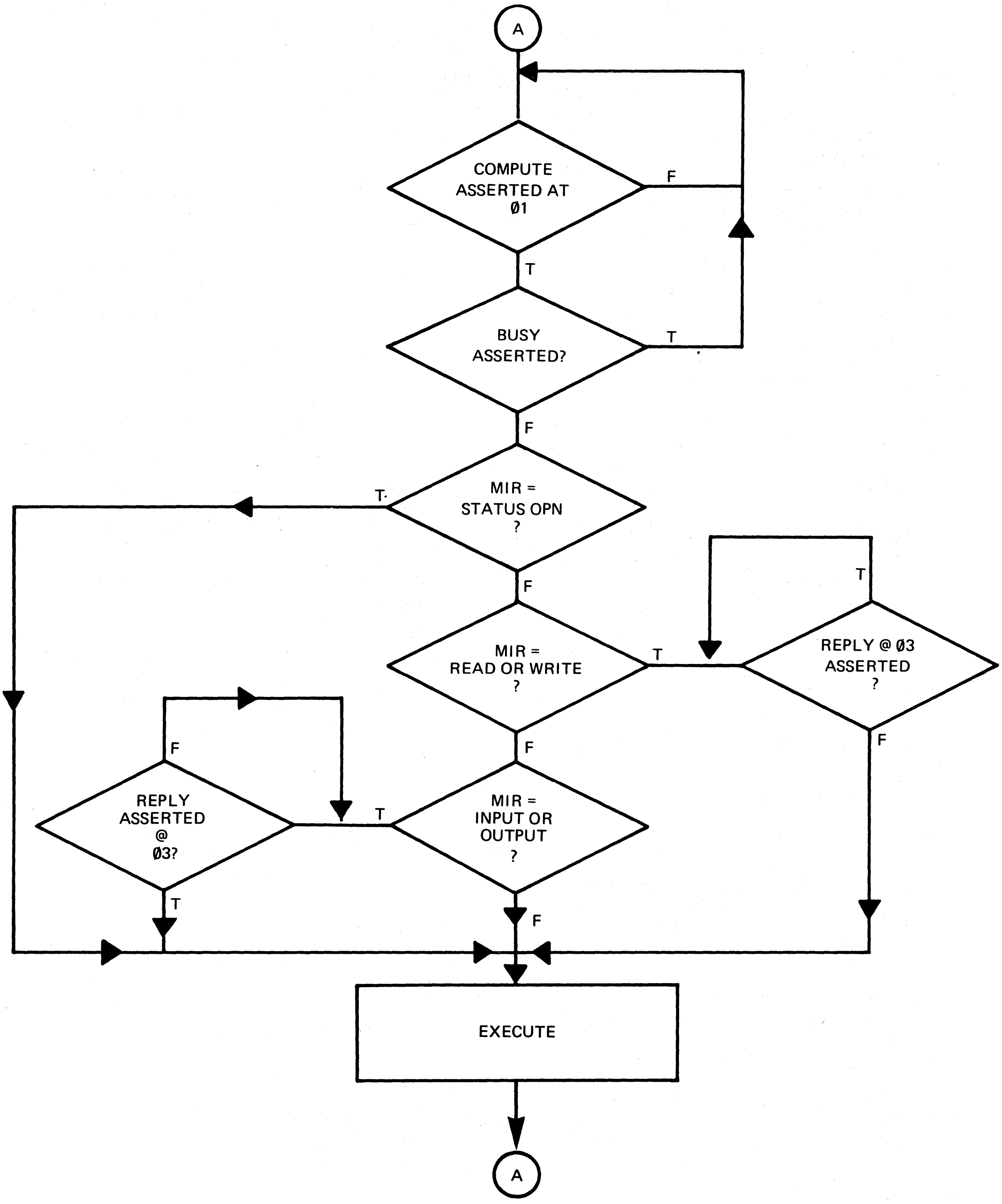


FIGURE 7-1. I/O INSTRUCTION CONDITION TESTING

7.2.1 READ INSTRUCTIONS

The READ instructions take a 16 bit address from a designated register pair and transfer it to the M register. When the address becomes valid on the bus, (during the following Ø1) the SYNC line is made high. The variations on the READ instruction are primarily for address manipulation and easing the coding of I/O routines. They cause the address source registers to be modified in some fashion. As far as the system designer is concerned, the pertinent operation of this instruction class is that it causes the selected address to be placed on the Data Access lines (DAL00-DAL15) and the SYNC line to be raised. The addressed device will assert the REPLY Line when ready for the data transfer. A flow diagram of the READ instruction operation is shown in Figure 7-2.

R b, a READ

F8	b	a
15	8	4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. Registers Rb and Ra are not changed.

Timing: 1 cycle

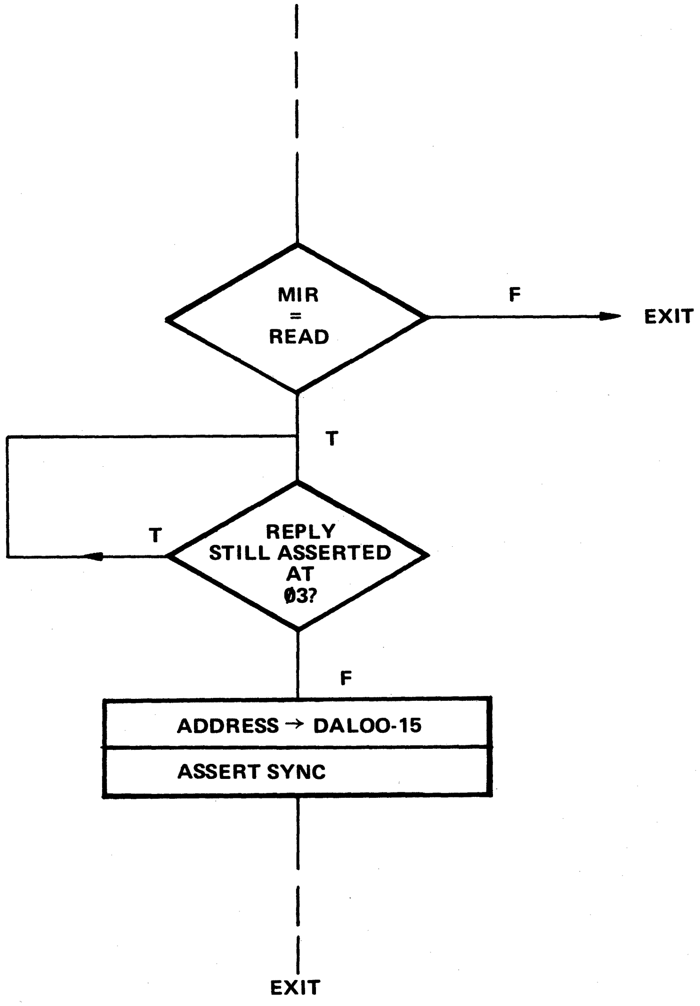


FIGURE 7-2. EXECUTION OF READ INSTRUCTION

RIB1 b, a READ AND INCREMENT BYTE BY 1

F0	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The contents of Register Ra are incremented by 1. Register Rb is not changed.

Timing: 1 cycle

RIB2 b, a READ AND INCREMENT BYTE BY 2

F4	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The contents of Register Ra are incremented by 2. Register Rb is not changed.

Timing: 1 cycle

RIW1 b, a READ AND INCREMENT WORD BY 1

F2	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The word in Ra+1:Ra is incremented by 1.

Timing: 2 cycles

RIW2 b, a READ AND INCREMENT WORD BY 2

F6	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The word in Ra+1:Ra is incremented by 2.

Timing: 2 cycles

7.2.2 WRITE INSTRUCTIONS

This class of instructions causes an address in a designated register pair to be placed on the Data Access lines DAL00-DAL15. The SYNC line is raised as the address becomes valid at the next occurrence of Ø1. Also the WRITE-BYTE Line is raised during the next occurrence of Ø1. When the addressed device is ready to transfer data, it will assert the REPLY line.

The comments about address source register manipulation pertain to this class of instructions as they do to the READ class of instructions described above.

A DATA WRITE operation is distinguished from a DATA READ operation by the assertion of WRITE/BYTE at the same time the address becomes valid on the bus. Note carefully that this signal later is used to describe data length when the data becomes valid on the bus.

A flow diagram of the WRITE instruction operation is shown in Figure 7-3.

W b, a WRITE

F9	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. Registers Rb and Ra are not changed.

Timing: 1 cycle

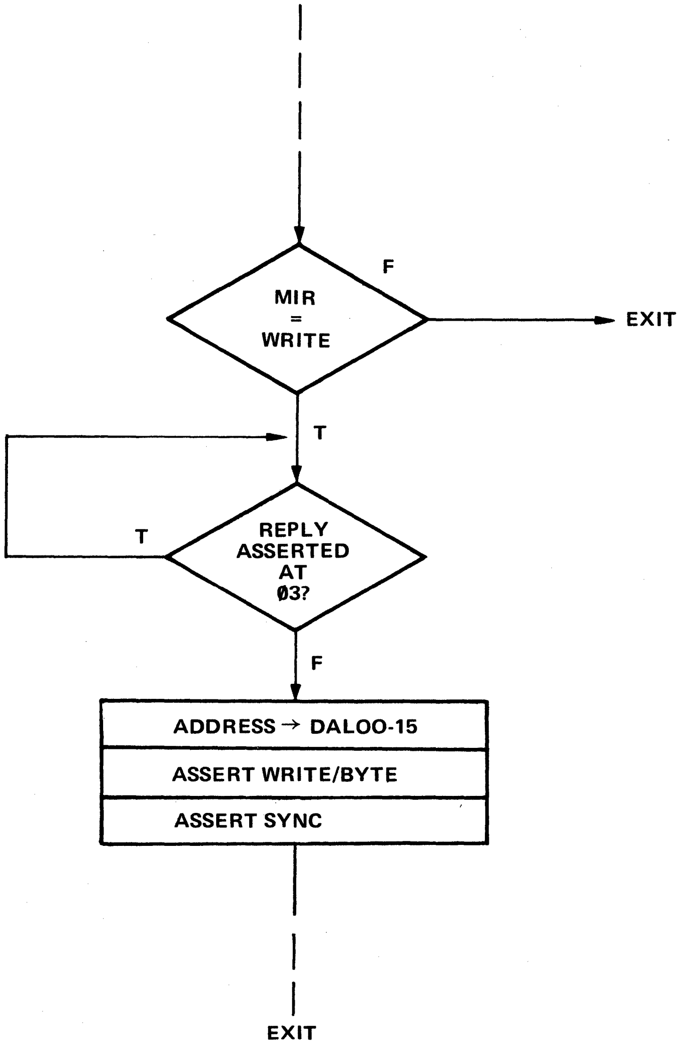


FIGURE 7-3. EXECUTION WRITE INSTRUCTION

WIB1 b, a WRITE AND INCREMENT BYTE BY 1

F1	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The contents of Register Ra are incremented by 1. Register Rb is not changed.

Timing: 1 cycle

WIB2 b, a WRITE AND INCREMENT BYTE BY 2

F5	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The contents of Register Ra are incremented by 2. Register Rb is not changed.

Timing: 1 cycle

WIW1 b, a WRITE AND INCREMENT WORD BY 1

F3	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The word in Ra+1:Ra is incremented by 1.

Timing: 2 cycles

WIW2 b, a WRITE AND INCREMENT WORD BY 2

F7	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The word in Ra+1:Ra is incremented by 2.

Timing: 2 cycles

7.2.3 INPUT INSTRUCTIONS

The INPUT Instructions control transfer of data from external devices on the Data Access to the processor. The data present on the data access is input by the instruction to the specified register or register pair. The INPUT class of instructions will not execute until a REPLY signal has been received from the device addressed by the previous READ instructions. Refer to a description of the REPLY signal below for the timing required. When this instruction is executed it sets DATA-IN high to inform the addressed device that it should place its data on the bus. This instruction terminates by making SYNC and DATA-IN Low on the Ø2 after completion.

Note that the instructions INPUT STATUS BYTE and INPUT STATUS WORD will execute regardless of the state of the REPLY signal and they do not affect SYNC and DIN. Both ISB and ISW can be stand alone microinstructions.

Note also that there are no timing restrictions after the selection of a device. As long as the operations are executed in the proper sequence, no timeouts or arbitrary disciplines bother the system designer.

The flow diagram in Figure 7-4 illustrates the operation of the INPUT instruction.

IB b, a INPUT BYTE

E0/E1	b	a

15 8 4 0

The 8-bit byte from the Data Lines, as specified by b, is placed in Register Ra. Code E1 causes the condition flags, except C, to be updated. The Read Data Access operation is terminated unless Bit 6

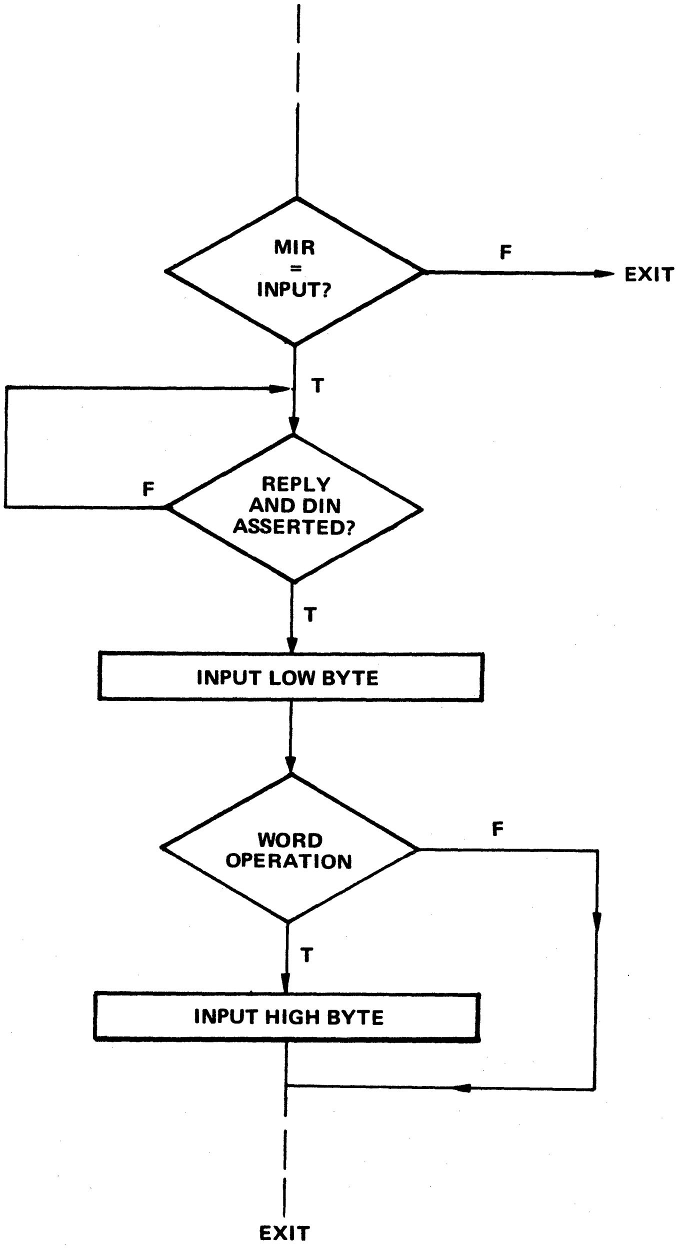


FIGURE 7-4. EXECUTION OF INPUT INSTRUCTION

is a one which allows a Read-Modify-Write (RMW) requiring termination by an output instruction. The instruction will not execute until after a Reply signal has been received from the addressed unit. The optional inputs are listed below:

b = 0	Upper Byte (Bits 15-8)
b = 1	Lower Byte (Bits 7-0)
b = 2	Upper Byte if M(0) = 1; Lower Byte if M(0) = 0
b = 3	Lower Byte if M(0) = 1; Upper Byte if M(0) = 0
b = 4	Upper Byte (Bits 15-8); RMW
b = 5	Lower Byte (Bits 7-0); RMW
b = 6	Upper Byte if M(0) = 1; Lower Byte if M(0) = 0; RMW
b = 7	Lower Byte if M(0) = 1; Upper Byte if M(0) = 0; RMW

Timing: 1 cycle (minimum)

IW b, a INPUT WORD

E2/E3	b	a

15 8 4 0

The 16-bit word from the Data Lines is placed in Registers Ra+1:Ra. Code E3 causes the condition flags, except C, to be updated. The Read Data Access operation is terminated unless Bit 6 is a one, which allows a Read-Modify-Write (RMW) operation requiring termination by an Output instruction. If Bit 4 or 5 is a one the word on the Data Lines is loaded in the Translation Register and at the same time either Bits 6-4 or Bits 8-6 of the DAL are loaded into the G Register. The instruction will not execute until after a Reply signal has been received from the addressed unit. The Lower Byte is loaded before the Upper Byte. The b options are listed below:

b = 0	
b = 1	Load TR; DAL 6-4 to GR; Sets ICS
b = 2	Load TR; DAL 8-6 to GR; Sets ICS
b = 3	Load TR; Sets ICS
b = 4	RMW
b = 5	Load TR; DAL 6-4 to GR; RMW; Sets ICS
b = 6	Load TR; DAL 8-6 to GR; RMW; Sets ICS
b = 7	Load TR; RMW; Sets ICS

Timing: 2 cycles (minimum)

NOTE: For both IB and IW a RMW sequence is allowed if and only if the Q1 flag is false. Q1 looks at the current contents of the translation register. The equation is as follows:

$Q1 = (\overline{TR14} \cdot TR13) + (\overline{TR14} \cdot \overline{TR12} \cdot TR11 \cdot \overline{TR10} \cdot TR9 \cdot TR8 \cdot TR7 \cdot TR6)$

$RMW = (B6 \cdot \overline{Q1})$.

Another version of the Control Chip exists without the Q1 flag. It is designated as CP1661B.

ISB b, a INPUT STATUS BYTE

E4/E5	b	a

15 8 4 0

The 8-bit byte from the Data Lines, as specified by b, is placed in Register Ra. Code E5 causes the condition flags, except C, to be updated. The instruction will input regardless of the state of the Reply signal. These optional inputs are listed below:

b = 0	Upper Byte (Bits 15-8)
b = 1	Lower Byte (Bits 7-0)
b = 2	Upper Byte if M(0) = 1; Lower Byte if M(0) = 0
b = 3	Lower Byte if M(0) = 1; Upper Byte if M(0) = 0

Timing: 1 cycle

ISW b, a INPUT STATUS WORD

E6/E7	b	a

15 8 4 0

The 16-bit word from the Data Lines is placed in Registers Ra+1:Ra. Code E7 causes the condition flags, except C, to be updated. This instruction will input regardless of the state of the Reply signal. The Lower Byte is loaded before the Upper Byte.

Timing: 2 cycles

7.2.4 OUTPUT INSTRUCTIONS

OUTPUT Instructions cause data to be transferred from the processor to the addressed peripheral devices. The OUTPUT instructions needs the REPLY signal to be asserted before execution will proceed. DATA-OUT is set Hi by the processor during the first phase of the instruction to inform the peripheral device that data is presently available on the Data Access. Also during this time, the WRITE/BYTE signal may be set to indicate the length of data on the bus.

The OUTPUT STATUS command operates as the OUTPUT instructions except that it does not pay attention to the state of the REPLY line, and does not affect SYNC and DOUT. It also can be a stand alone microinstruction.

After the execution of the OUTPUT instruction, data becomes valid on the bus. Also, at this time, WRITE/BYTE will be asserted if the data size is one byte. If the data size is a word, WRITE/BYTE will become passive.

Refer to Figure 7-5 for the flow diagram of the OUTPUT instruction operation.

OB b, a OUTPUT BYTE

FC	b	a

15 8 4 0

The 16-bit contents of Registers Rb:Ra are transferred to the M Register and the Data Lines. The Data Out signal is activated. Registers Rb and Ra are not changed. To provide proper operation with a 16-bit

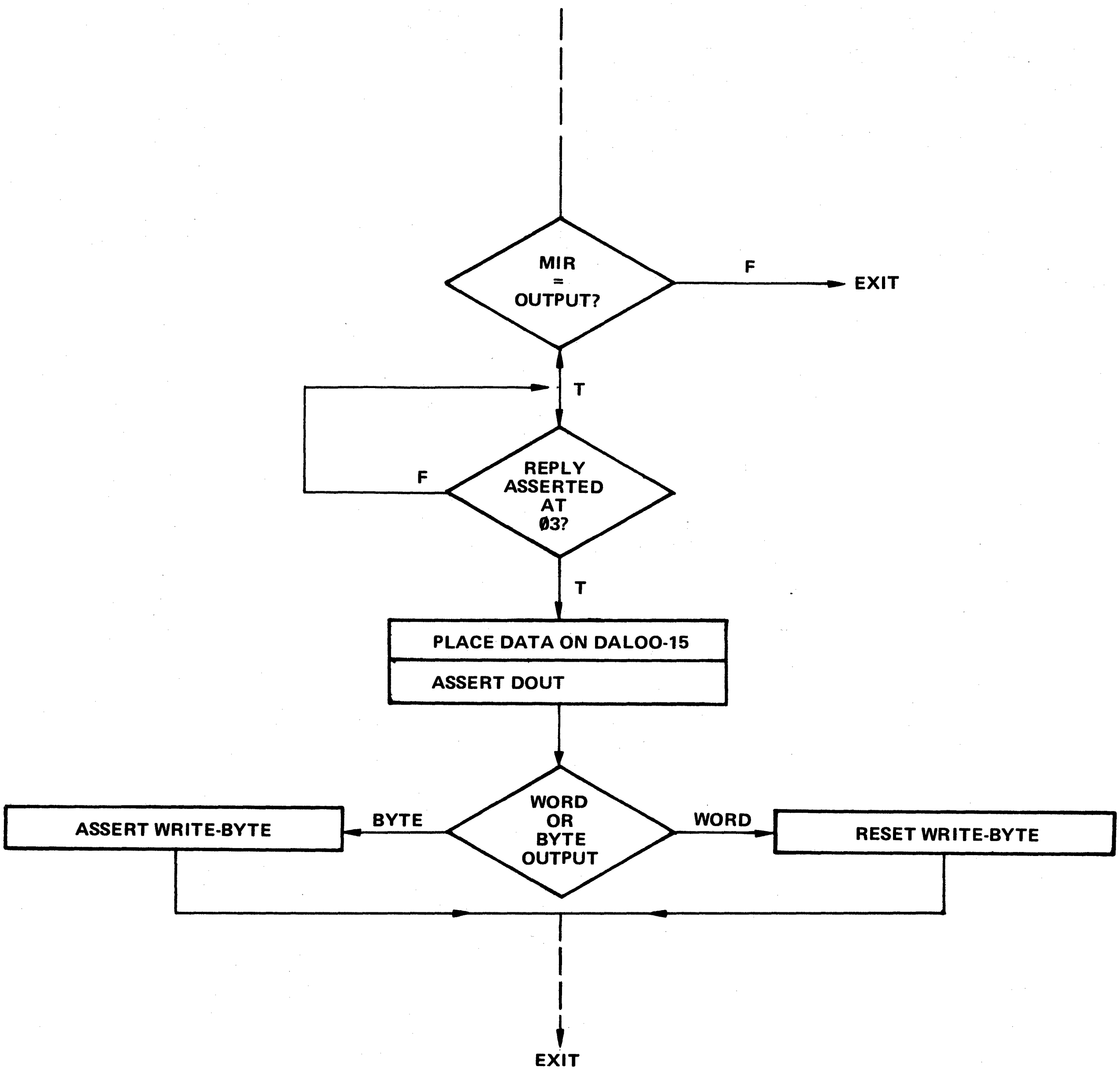


FIGURE 7-5. EXECUTION OF OUTPUT INSTRUCTION

Data Path b must equal a so that the same byte is placed in both byte positions of the M Register. Output does not take place until Reply has been received from the addressed unit.

Timing: 1 cycle (minimum)

OW b, a OUTPUT WORD

FD	b	a

15 8 4 0

The 16-bit contents of Registers Rb:Ra are transferred to the M Register and the Data Lines. The Data Out signal is activated. Registers Rb and Ra are not changed. Output does not take place until Reply has been received from the addressed unit.

Timing: 1 cycle (minimum)

OS b, a OUTPUT STATUS

FE	b	a

15 8 4 0

The 16-bit contents of Registers Rb:Ra are transferred to the M Register and the Data Lines. Registers Rb and Ra are not changed. Output takes place regardless of the state of the Reply signal. This instruction is normally used without a Write instruction, and cannot turn off Sync.

Timing: 1 cycle

7.2.5 INTERRUPT ACKNOWLEDGE INSTRUCTIONS

These instructions perform in identical fashion to the READ/WRITE instructions mentioned in Sections 7.2.1 and 7.2.2. Additionally, they raise the Interrupt Acknowledge (IACK) line. The Interrupt Acknowledge signal and a predetermined address placed on the Data Access bus by this instruction can be used as a signal to the I/O devices (if the system designer wishes) to inform the I/O set that the device requesting service should place his device number on the bus for transmission to the processor. Depending upon whether or not a READ acknowledge or a WRITE acknowledge was executed, the succeeding instruction may be an INPUT or OUTPUT Instruction respectively.

RA b, a READ ACKNOWLEDGE

FA	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated with the Interrupt Acknowledge line high. Registers Rb and Ra are not changed. The Interrupt Acknowledge signal along with one or more address bits is used to form a signal which polls I/O units for the one interrupting the processor on the Interrupt line corresponding to the address. The processor inputs a byte or word containing the identification of the interrupting unit.

Timing: 1 cycle

WA b, a WRITE ACKNOWLEDGE

FB	b	a

15 8 4 0

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated with the Interrupt Acknowledge line high. Registers Rb and Ra are not changed. The Interrupt Acknowledge signal along with one or more address bits is used to form a signal which polls I/O units for the one interrupting the processor on the Interrupt line corresponding to the address. This unit receives the next data output.

Timing: 1 cycle

7.3 DATA/ADDRESS LINES

Sixteen Data/Address Lines, denoted DAL00-DAL15, are used to transfer addresses and data from the processor and to receive data into the processor. This bus is TTL compatible and signals on it are logical true data.

An address is output by the Data Chip as a result of executing a READ or a WRITE instruction. It appears on the bus at Ø1 following the execution of the instruction. The address is valid on the bus during phases 2, 3 and 4.

Data is output by the processor as a result of executing an OUTPUT instruction. It appears on the bus at the first Ø1 after the microinstruction is executed and remains on the bus for a minimum of one cycle. Data is received from the Data Access into the processor by the INPUT instruction. It is clocked in at Ø4 of the instruction cycle.

7.4 CONTROL LINES

This section describes the Control lines generated by the CP1621 Control Chip as a result of I/O instruction execution. It provides a functional description of the control lines. Their timing and interactions with the Data Access lines are further defined in Section 7.6.

Sync (TTL)

The SYNC is a Control signal used to initiate and signify the length of a Data Access operation. SYNC is made high as soon as an address becomes valid. This occurs at Ø2 following a READ or WRITE. It remains high until the termination of the operation.

Reply (TTL)

The REPLY is a Control signal used by the addressed unit to respond to the Processor's Data Access signals.

The REPLY signal must be high during Ø3 of the INPUT or OUTPUT microinstruction execution cycle in order for this operation to complete. The REPLY signal is also interrogated by READ and WRITE micro-instructions and it must be low during Ø3 in order for these operations to take place.

Data-In (TTL)

The DATA-IN (DIN) is a Control signal from the Processor to cause the address unit to gate its Read data on the Data lines. It is made high at the time the address is removed from the lines, or one cycle after the SYNC is made high (the second Ø2 of the INPUT) and is a function of the READ instruction. The DATA-IN is made low at the end of the Input Byte or Input Word instruction or when SYNC is made low. This signal can be used to control the enabling of external TTL Tri-State Bus Driver/Receivers.

Data-Out (TTL)

The DATA-OUT (DOUT) is a Control signal from the Processor which is made high at the same time as the Write data (Ø1 following the OUTPUT) is placed on the DAL bus by the Processor. It remains high for the duration of the OUTPUT instruction, dropping one phase before the data is taken off the DAL bus.

Write/Byte (TTL)

The WRITE/BYTE (WB) is a Control signal from the Processor which is high during the time the address is on the bus to signify a WRITE rather than a READ operation; and is high during Data-Out to signify a Byte output rather than a Word output. To indicate an Output, it comes up at Ø1 following a WRITE.

Interrupt Acknowledge (TTL)

The IACK is a Control signal from the Processor which signifies that the Processor is responding to an Interrupt. This signal is made high at the same time the SYNC is made high as a result of either 'RA' or 'WA' instructions, and stays high as long as SYNC is high.

Busy (TTL)

The BUSY is a Control signal from an external unit to the Processor requesting access to the bus. The signal can be used, for example, by a DMA unit to access the memory. The BUSY signal is interrogated at Ø3 by the Processor every time READ or WRITE instructions are taking place. Whenever the BUSY signal is found to be one, the Processor enters a WAIT state inhibiting any access operation from taking place. The Processor will resume normal operation as soon as BUSY is turned off.

7.5 INTERRUPT, RESET, AND COMPUTE

This section describes signals which would, in the normal scheme of things, be defined as processor control signals.

7.5.1 RESET

RESET is a TTL level line that may be controlled by an external device. Activation of the RESET line causes the Microprocessor to force 001 into the Location Counter. A NOP is also forced into the MIR and the MI registers. SYNC and DATA-IN are both reset. The RESET line can be wired to a POWER ON reset or it may be used by the program or its own purposes.

Note that the activation of RESET is a hard action in that everything stops and the above mentioned conditions are forced.

7.5.2 COMPUTE

COMPUTE is also a TTL level signal and it controls the processor's execution of microinstructions. The processor examines COMPUTE during every Ø1 to determine whether or not it should execute the present microinstruction. In the case of a two cycle instruction COMPUTE need be high only during Ø1 of the first cycle. Among other things COMPUTE may be used to control single stepping of microinstructions. This line should not be confused with the WAIT signal on the MIB bus.

7.5.3 INTERRUPTS

The external INTERRUPT lines of the MCP 1600 system are IO-I3. These are microprogrammable. These lines provide inputs to Array 3 of the Programmable Translation Array and thus may be checked at certain user defined addresses in the Microprogram. As the examination of these lines is controlled by the contents of the Location Counter and the Macroinstruction being translated, it is easy for the microprogram to examine them for instance, before the FETCH cycle of every macroinstruction. There is no discipline associated with the use of these interrupt lines. Their state may be changed at any time. The system designer may implement his own interrupt control scheme by use of these lines and the ACKNOWLEDGE instructions described in Section 7.2.5 above.

7.6 INPUT/OUTPUT OPERATIONS

The Data Access of the MCP1600, because of its flexibility, can be used to configure a variety of I/O schemes. Several of the schemes that can be implemented were mentioned in the introduction. This section will describe the interactions of the elements that comprise the Data Access and then define some canned operations that have been found useful. The combination of these canned operations and a discipline that structures interrupts and device addresses can result in a fast and powerful I/O structure.

7.6.1 STANDARD I/O SEQUENCES

There are five standard I/O sequences that have been developed for the MCP1600 Processor System. These sequences provide for the orderly transfer of data to and from the processor. Two sequences provide for normal READs and WRITEs. One sequence provides a READ/MODIFY/WRITE capability which is useful for controlling random access memory devices. The last two sequences are READ/WRITE INTERRUPT ACKNOWLEDGE sequences.

These sequences are as described both verbally and with timing diagrams. Note that in the timing diagrams, the first clock cycle (Ø1, - - - Ø4) is devoted to instruction interpretation. Bus operations don't begin to occur until the second clock cycle.

7.6.1.1 WRITE/OUTPUT SEQUENCE

This sequence consists of two instructions:

WRITE	Device Number
OUTPUT	Data Source

The Write operation transfers data from the processor to the addressed unit. The Write is initiated by a Write instruction which transfers a 16-bit address to the Data Access port. The address is present on the lines for one cycle. Data is transferred from the processor registers to the Data Access port by an Output Byte or Output Word instruction. The operation is terminated after the data has been on the lines for a minimum of one cycle. When outputting a byte with a 16-bit data path, the same byte must be placed in both the upper and the lower bytes of the port and the addressed unit takes care of storing the byte in proper half of the word as selected by the low-order address bit.

The time period between selection of the device by the WRITE instruction and the transfer of the data is not critical if the device controller latches the selection. The sequence is graphically shown in Figure 7-6 below.

7.6.1.2 READ/INPUT SEQUENCE

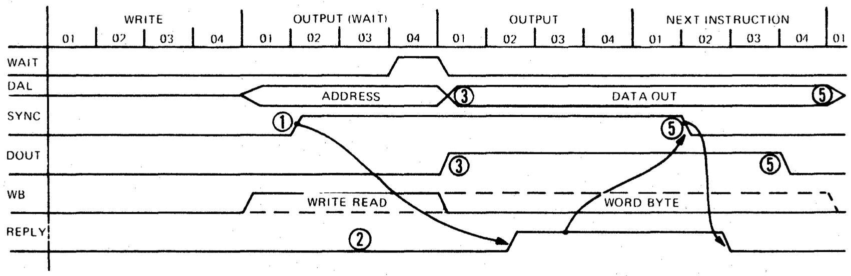
This sequence consists of two instructions which select the device, then transfer data.

READ	Device Number
INPUT	Data Destination

The Read operation transfers data from the addressed unit to the processor. The Read is initiated by a Read instruction which transfers a 16-bit address to the Data Access port. The address is present on the lines for one cycle and then the processor signals the addressed unit to put its data on the lines. Data is input to the processor registers by an Input Byte or Input Word instruction. The Read operation is terminated by either Input or Output instructions. An Input Byte instruction allows for selection of the Upper Byte, Lower Byte or the Byte selected by the lower-order bit of the address.

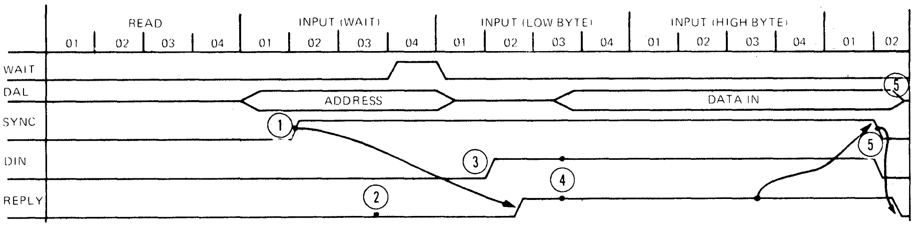
The time period between execution of the READ and INPUT INSTRUCTIONS is not critical if selection latching is used in the device controller.

Figure 7-7 illustrates the READ/INPUT sequence.



1. 'SYNC' IS MADE HIGH AS SOON AS A VALID ADDRESS IS ON THE BUS (02).
2. 'REPLY' BEING LOW ON 03 CAUSES THE 'OUTPUT' INSTRUCTION TO BE EXECUTED REPEATEDLY.
3. 'DOUT' IS MADE HIGH ON 01 AS 'DATA OUT' IS PLACED ON THE BUS.
4. 'OUTPUT' INSTRUCTION IS TERMINATED BY RAISING 'REPLY' BEFORE 03.
5. 'SYNC' IS MADE LOW ON 02, 'DOUT' IS MADE LOW ON 04, AND 'DAL' IS TRI STATED ON 01
REPLY CAN BE TURNED OFF AS SOON AS 'SYNC' IS MADE LOW.

FIGURE 7-6. WRITE/OUTPUT SEQUENCE



1 'SYNC' IS MADE HIGH AS SOON AS A VALID ADDRESS IS ON THE BUS (02).
2. 'REPLY' BEING LOW ON 03 CAUSES THE 'INPUT' INSTRUCTION TO BE EXECUTED REPEATEDLY
3. 'DIN' IS MADE HIGH AS SOON AS THE ADDRESS IS TAKEN OFF THE BUS (02) AS FUNCTION OF READ MICROINSTRUCTION
4 'REPLY' AND 'DIN' MUST BE HIGH ON 03 IN ORDER FOR 'INPUT' INSTRUCTION TO TAKE PLACE.
5 UPON TERMINATION OF 'INPUT' INSTRUCTION, 'SYNC' AND 'DIN' ARE MADE LOW ON 02

FIGURE 7-7. READ/INPUT SEQUENCE

7.6.1.3 READ/MODIFY/WRITE SEQUENCE (See Figure 7-8)

The length of this sequence is variable. At least three instructions are required:

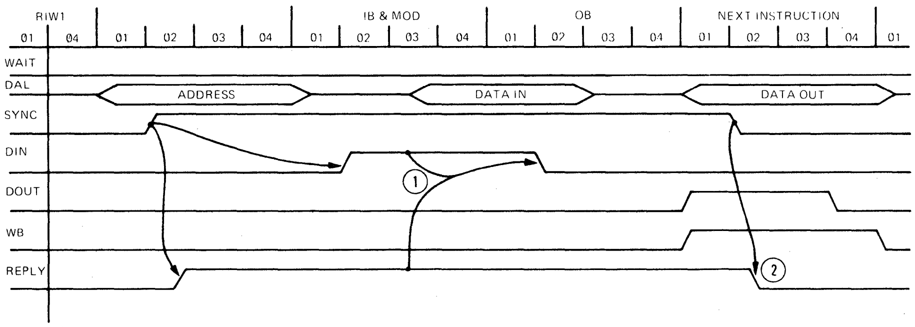
READ Device Number

INPUT Destination Register

Microprogram can modify data

OUTPUT Source Register

A Read/Modify/Write operation is both a Read and Write in a single Data Access operation. This provides for inputting data from an addressed unit, modifying it, the outputting the changed data to the same addressed unit. The Read/Modify/Write is initiated by a Read instruction which transfers a 16-bit address to the Data Access port. The address is present on the lines for one cycle and then data is placed on the lines by the addressed unit. Data is input to the processor registers by an Input Byte or Input Word instruction. The Data Access operation is not terminated as in the case of a normal Read, but continues for a subsequent output, as specified by the Input instruction. This suspended period can be used to modify the data, if desired. An example might be an Increment Memory instruction. After modification, data is transferred from the processor registers to the Data Access port by an Output Byte or Output Word instruction. The operation is terminated after the data has been on the line for a minimum of one cycle.



1. 'INPUT & MOD' WILL TURN 'DIN' OFF, BUT WILL KEEP 'SYNC' ON.
2. 'REPLY' CAN BE MADE LOW AS SOON AS'SYNC' AND 'DIN' ARE OFF.

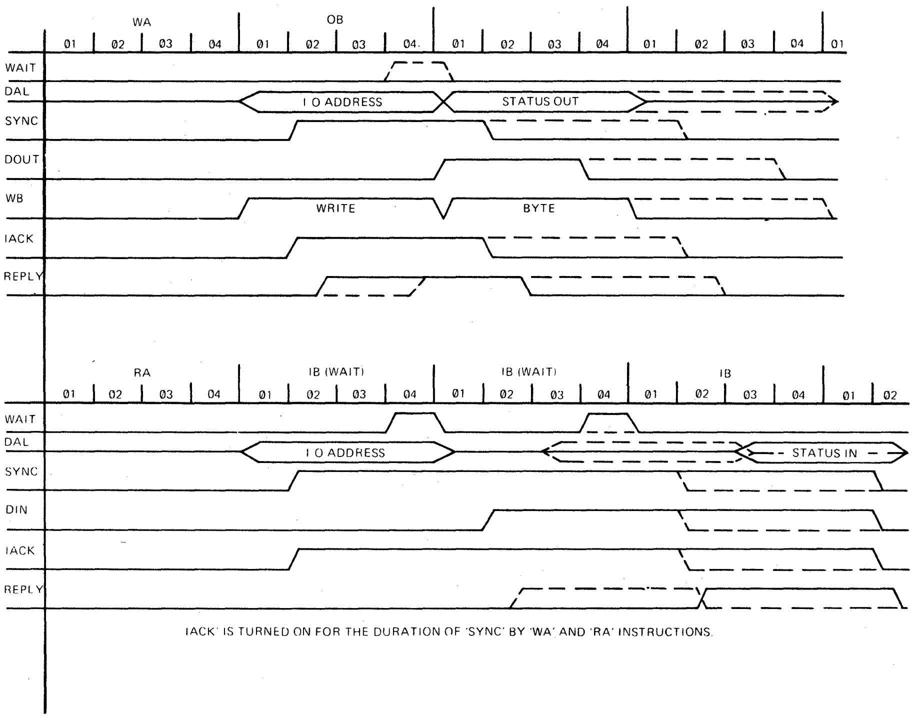
FIGURE 7-8. READ/MODIFY/WRITE SEQUENCE

7.6.1.4 INTERRUPT ACKNOWLEDGE SEQUENCES

These two sequences provide the system designer with the ability to structure his own interrupt system. In addition to the normal READ or WRITE sequences, the processor will raise IACK at the specified time.

The combination of IACK and a special reserved device number on DAL00-15 could, for example, order the device controller requesting attention to return its device number on the subsequent INPUT instruction.

Figure 7-9 illustrates the operation of both the READ and WRITE Interrupt Acknowledge.



IACK' IS TURNED ON FOR THE DURATION OF 'SYNC' BY 'WA' AND 'RA' INSTRUCTIONS.

FIGURE 7-9. INTERRUPT ACKNOWLEDGE SEQUENCE

SECTION VIII

ELECTRICAL SPECIFICATIONS

8.1 GENERAL

This section describes the electrical characteristics of the parts making up the WDC MCP 1600 Microprocessor Set. Each of the three required devices is parameterized.

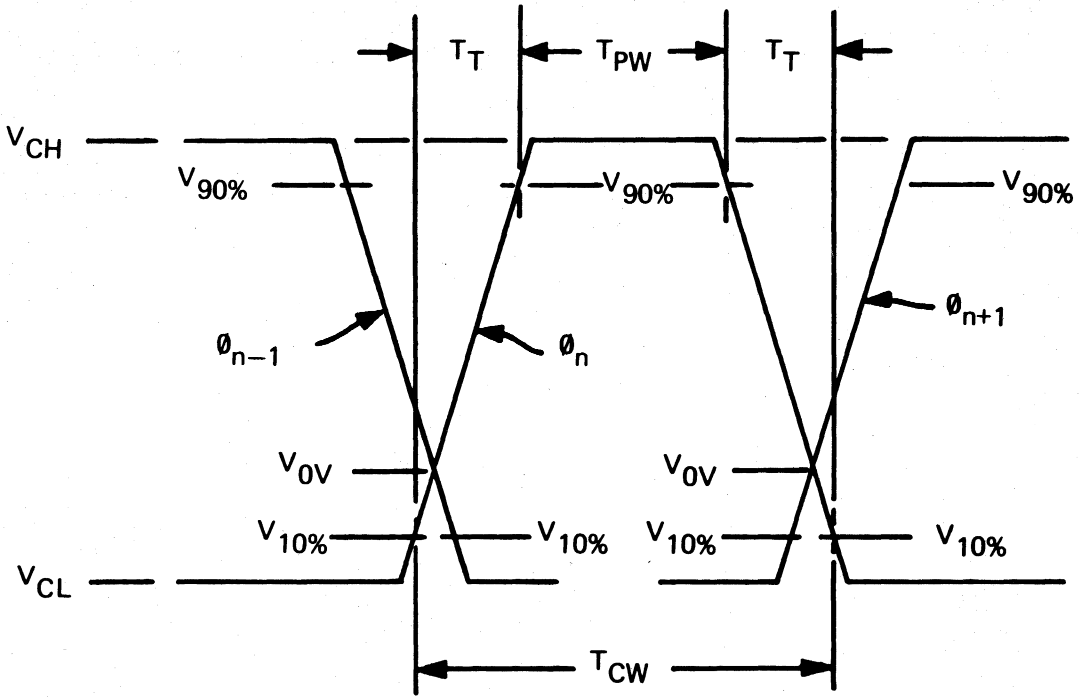
8.2 CP1611B MICROPROCESSOR DATA CHIP

This section describes the electrical interface requirements of the CP1611B Microprocessor Data Chip. Adherence to nominal values will ensure reliable operation.

This is a preliminary specification with tentative device parameters and may be subject to change after final product characterization is completed.

Information furnished by Western Digital Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Western Digital Corporation for its use; nor any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Western Digital Corporation. Western Digital Corporation reserves the right to change said circuitry at any time without notice.

CLOCK DIAGRAM



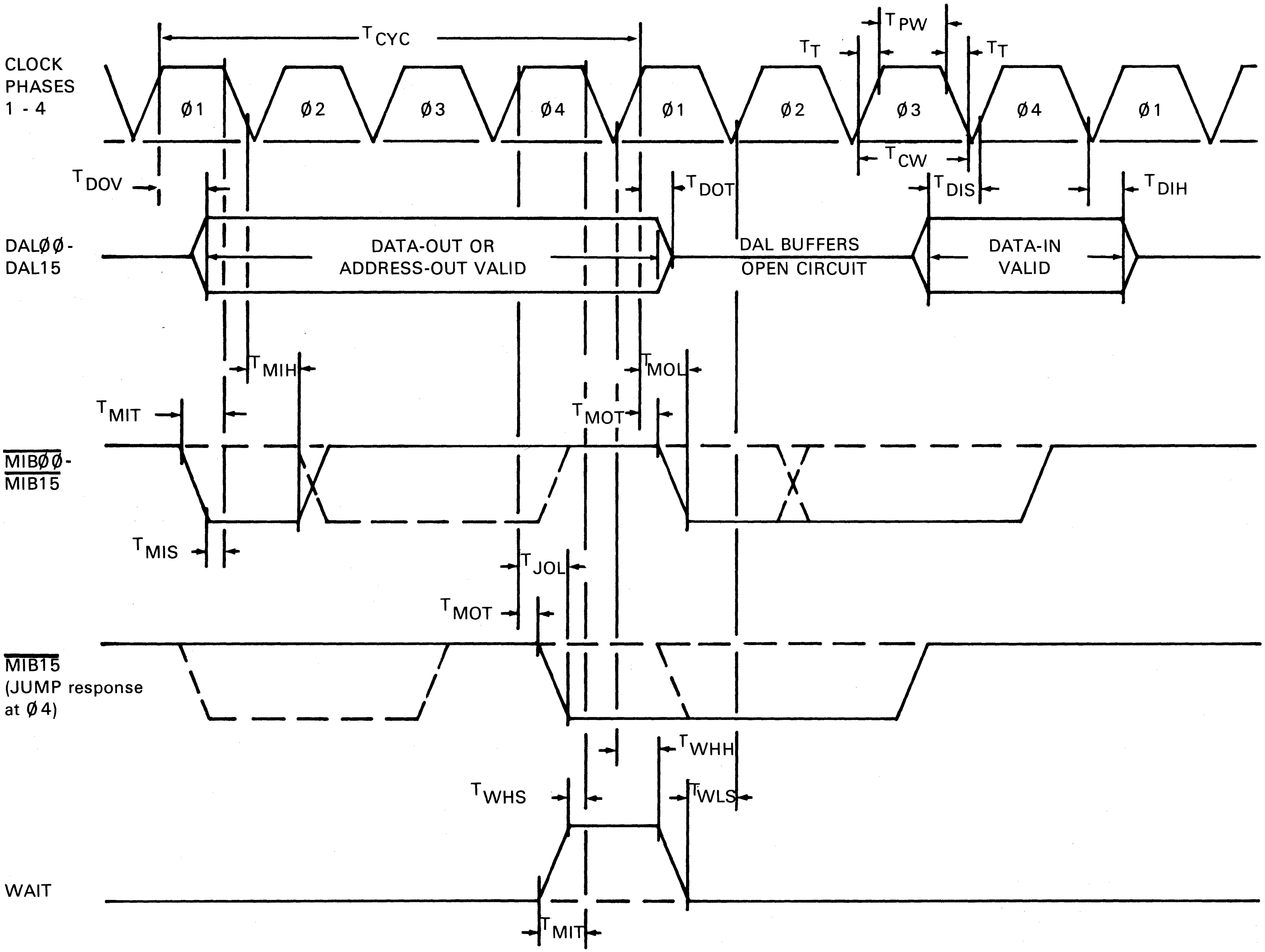
DEFINITION OF 10% AND 90% VOLTAGE POINTS
FOR CLOCK, INPUTS AND OUTPUTS

$$V_{10\%} = V_{LOW\ (MAX.)} + .1\left[V_{HIGH\ (MIN.)} - V_{LOW\ (MAX.)}\right]$$

$$V_{90\%} = V_{LOW\ (MAX.)} + .9\left[V_{HIGH\ (MIN.)} - V_{LOW\ (MAX.)}\right]$$

This definition applies to clock, input and output pins.

FIGURE 8-1
MCP16ØØ CLOCK DIAGRAM



NOTE: SWITCHING TIMES ARE MEASURED AT 10% AND 90% OF SPECIFIED LEVELS.

FIGURE 8-2
1611 INTERFACE TIMING

ABSOLUTE MAXIMUM RATINGS OVER FREE AIR TEMPERATURE RANGE (Unless Otherwise Noted)*

Supply Voltage V_{DD} (See NOTE)	–0.5V to 15V
Supply Voltage V_{BB} (See NOTE)	–10V to 1.0V
All Other Pin Voltages (See NOTE)	–1.0V to 15V
Clock Voltage (See NOTE)**	–1.0V to 15V
Operating Free Air Temperature Range	0°C to 125°C
Storage Temperature Range	–55°C to 125°C

NOTE: These voltage values are with respect to V_{SS} Supply Voltage. If V_{BB} is more positive than any other voltage, then I_{BB} must be limited to 10ma.

*Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum-rating conditions for extended periods may affect device reliability.

**The other three clock voltages must be between 0.5V and –0.6V except for switching overlaps. Not more than one clock may be high at any one time.

Applying power to the part may be any sequence of conditions that do not violate the maximum ratings specified on this sheet.

OPERATING CHARACTERISTICS

T_{CASE} = Ø°C to 70°C, V_{DD} = +12.0V± .6V, V_{BB} = –3.9V± .25V, V_{SS} = ØV

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
I_L	Leakage Current For Any Pin Other Than Clocks or Power			±10	μA	V_{IN} – 5.25V/0V
I_{BB}	V_{BB} Supply Current			–500	μA	V_{BB} = –5.25V
I_{LC}	Clock Leakage Current			±100	μA	V_{CLOCK} = 13.7V/0V
$I_{DD_{AVE}}$	Average V_{DD} Operating Current		20.0	35.0	mA	T_{CYC} = 300 nsec C_L = 50 pf
V_{IHM}	Input High Voltage $\overline{\text{MIBØØ-MIB15}}$, WAIT	4.1		5.5	V	
V_{ILM}	Input Low Voltage $\overline{\text{MIBØØ-MIB15}}$, WAIT	0.0		0.8	V	
V_{OLM}	Output Low Voltage $\overline{\text{MIBØØ-MIB15}}$	0.0		0.4	V	I_O = 100 μA during Ø1 only
V_{OV}	Overlap Voltage of Any Two Adjacent Clock Phases	0.0		3.0	V	
V_{CH}	Clock High Voltage (SEE NOTE)	11.8 12.0		13.0 13.7	V	V_{DD} = 11.4 V_{DD} = 12.6
V_{CL}	Clock Low Voltage (SEE NOTE)	–0.6		0.5	V	
V_{OHT}	TTL Output High Voltage DALØØ-DAL15	2.4		5.5	V	I_O = –100 μA
V_{OLT}	TTL Output Low Voltage DALØØ-DAL15	0.0		0.4	V	I_O = 1.8 mA
V_{IHT}	TTL Input High Voltage	2.4		5.5	V	$I_I \leqslant 10$ μA
V_{ILT}	TTL Input Low Voltage	–1.0		0.8	V	$I_I \geqslant -10$ μA

NOTE: Linear interpolation applies for V_{CH} when V_{DD} is between 11.4V and 12.6V. No overshoot or Undershoot allowable.

AC CHARACTERISTICS

T_{CASE} = Ø°C to 70°C, VDD = +12V ± .6V, V_{BB} = −3.9V ± .25V, V_{SS} = ØV

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
T_{PW}	Clock Width High (All Phases)	55		240	nsec	
T_{CW}	Clock Width Low (All Phases)	75		300	nsec	
T_{CYC}	Clock Period (All Phases)	300		1000	nsec	
T_{T}	Clock Transition Time (All Phases)	5			nsec	
T_{MIS}	$\overline{\text{MIBØØ-MIB15}}$ Input Set-Up Time At Ø1	20			nsec	
T_{MIH}	$\overline{\text{MIBØØ-MIB15}}$ Hold Time After Ø1	5			nsec	
T_{MOL}	Time to Discharge $\overline{\text{MIBØØ-MIB15}}$ to a Low by the Data Chip at Ø1			35	nsec	C_L = 50 pf
T_{JOL}	Time to Discharge $\overline{\text{MIB15}}$ Low at Ø4 by the Data Chip for the JUMP Response			35	nsec	C_L = 50 pf
T_{DOV}	Address or Data Out Switching Time from Ø1 to High or Low Valid			70	nsec	Fig. 8-2
T_{DOT}	Address or Data Out Tri-State Time from Ø1			55	nsec	Fig. 8-2
T_{DIS}	Data In Set-Up Time to Ø4	15			nsec	
T_{DIH}	Data In Hold Time After Ø4	10			nsec	
T_{WHS}	Wait Input High Set-Up Time	20			nsec	
T_{WHH}	Wait Input High Hold Time	5			nsec	
T_{WLS}	Wait Input Low Set-Up Time	20			nsec	
TMIT	Input Transition Start Set-up Time	35			nsec	
TMOT	Output Transition Start Delay Time			20	nsec	C_L = 50 pf

CAPACITANCE

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
$C_{\emptyset 1}$	Clock Phase 1 Capacitance		40	60	pf	V_{IN} = ØV, V_{SS} = ØV, V_{DD} = ØV, V_{BB} = −3.9V f = 1 MHz
$C_{\emptyset 2}$	Clock Phase 2 Capacitance		40	60	pf	
$C_{\emptyset 3}$	Clock Phase 3 Capacitance		40	60	pf	
$C_{\emptyset 4}$	Clock Phase 4 Capacitance		40	60	pf	
C_D	Data Input/Output Pin Capacitance		6.0	15.0	pf	
C_C	Clock to Clock Capacitance		4.0	7.0	pf	

8.3 CP1621 MICROPROCESSOR CONTROL CHIP

This section describes the electrical interface requirements of the CP1621B Microprocessor Control Chip. Adherence to nominal values as specified will ensure reliable operation.

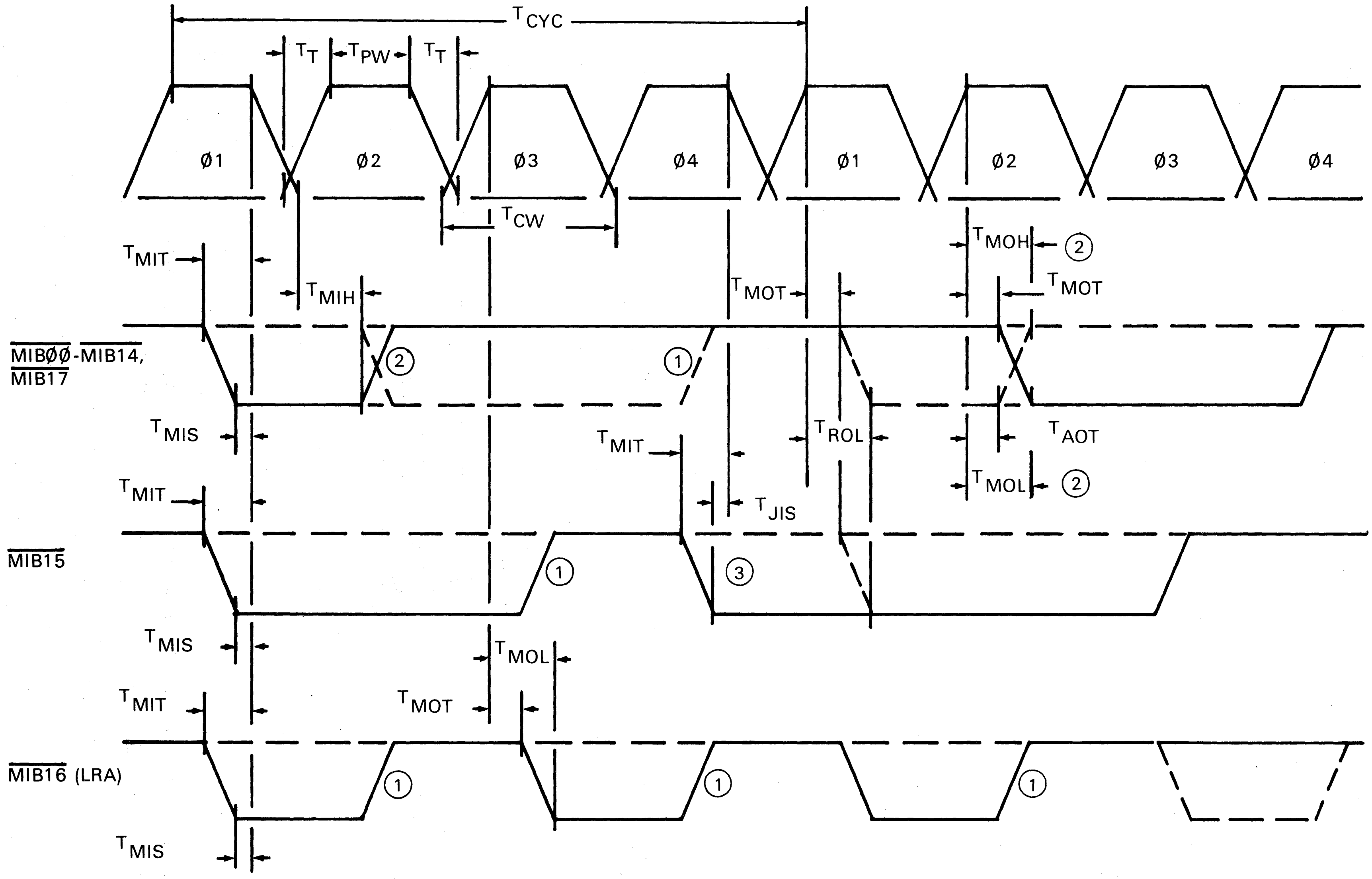
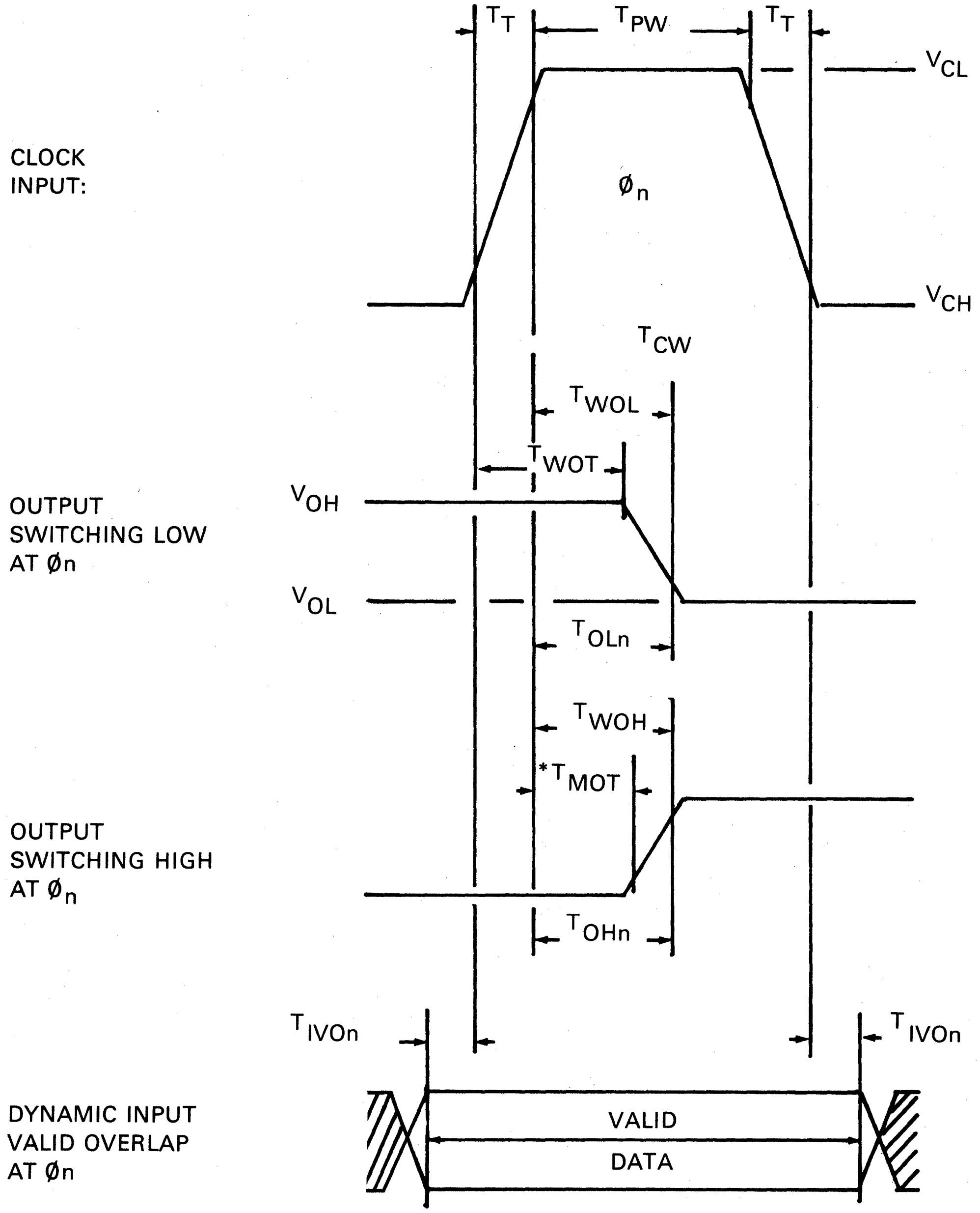


FIGURE 8-3
1621 MIB TIMING

NOTES:

1. BUS PRECHARGING IS PERFORMED BY THE CP1631B.
2. ADDRESS OUTPUT SWITCHING TIME (T_{MOH} AND T_{MOL}) APPLIES TO $\overline{\text{MIBØØ}}$-$\overline{\text{MIB10}}$ ONLY.
3. JUMP RESPONSE FROM DATA CHIP CP1611B.
4. SWITCHING TIMES ARE MEASURED AT 10% AND 90% OF SPECIFIED LEVELS.



NOTE: SWITCHING TIMES ARE MEASURED AT 10% AND 90% OF SPECIFIED LEVELS.
*NOTE: T_{MOT} only applies to the wait signal in these drawings.

FIGURE 8-4
1621 CONTROL SIGNALS TIMING

ABSOLUTE MAXIMUM RATINGS OVER FREE AIR TEMPERATURE RANGE (Unless Otherwise Noted)*

Supply Voltage V_{DD} (See NOTE)	–0.5V to 15V
Supply Voltage V_{CC} (See NOTE)	–0.5V to 15V
Supply Voltage V_{BB} (See NOTE)	–10V to 1.0V
All Other Pin Voltages (See NOTE)	– 1.0V to 15V
Clock Voltage (See NOTE)**	– 1.0V to 15V
Operating Free Air Temperature Range	0°C to 125°C
Storage Temperature Range	–55°C to 125°C

NOTE: These voltage values are with respect to V_{SS} Supply Voltage. If V_{BB} is more positive than any other voltage, THEN I_{BB} must be limited to 10ma.

*Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum-rating conditions for extended periods may affect device reliability.

**The other three clock voltages must be between 0.5V and – 0.6V except for switching overlaps. Not more than one clock may be high at any one time.

Applying power to the part may be any sequence of conditions that do not violate the maximum ratings specified on this sheet.

OPERATING CHARACTERISTICS

T_{CASE} = Ø°C to 70°C, V_{DD} = + 12.0V ± .6V, V_{BB} = – 3.9V ± .25V, V_{SS} = ØV, V_{CC} = + 5V ± .25V

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
I_{LI}	Leakage Current For Any Pin Other Than Clocks or Power			±10	µA	V_{IN} = 5.25V/ØV
I_{BB}	V_{BB} Supply Current			– 500	µA	V_{BB} = – 5.25V
I_{LC}	Clock Leakage Current			±100	µA	V_{CLOCK} = 13.7V/ØV
ICC_{AVE}	Average V_{CC} Operating Current		7.0	12.0	mA	T_{CYC} = 300 nsec C_L = 50pf
IDD_{AVE}	Average V_{DD} Operating Current		20.0	35.0	mA	T_{CYC} = 300 nsec C_L = 50 pf
V_{IHM}	MOS Input High Voltage	4.1		V_{CC}	V	
V_{ILM}	MOS Input Low Voltage	0.0		0.8	V	
V_{OHM}	MOS Output High Voltage	4.25		V_{CC}	V	I_O = – 30 µA
V_{OLM}	MOS Output Low Voltage	0.0		0.4	V	I_O = 100 µA
V_{OV}	Overlap Voltage of Any Two Adjacent Clock Phases	0.0		3.0	V	
V_{CH}	Clock High Voltage (SEE NOTE)	11.8 12.0		13.0 13.7	V	V_{DD} = 11.4V V_{DD} = 12.6V
V_{CL}	Clock Low Voltage (SEE NOTE)	– 0.6		0.5	V	
V_{OHT}	TTL Output High Voltage	2.4		V_{CC}	V	I_O = – 100 µA
V_{OLT}	TTL Output Low Voltage	0.0		0.4	V	I_O = 1.8 mA
V_{IHT}	TTL Input High Voltage	2.7		V_{CC}	V	$I_1 \leq 10$ µA
V_{ILT}	TTL Input Low Voltage	–1.0		0.8	V	$I_1 \geq -10$ µA

NOTE: Linear interpolation applies for V_{CH} when V_{DD} is between 11.4V and 12.6V. No overshoot or undershoot allowable.

OPERATING CHARACTERISTICS

T_{CASE} = Ø°C to 70°C, V_{DD} = +12V ± .6V, V_{BB} = −3.9V ± .25V, V_{SS} = ØV, V_{CC} = +5V ± .25V

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
T_{PW}	Clock Width High (All Phases)	55		240	nsec	
T_{CW}	Clock Width Low (All Phases)	75		300	nsec	
T_{T}	Clock Transition Time (All Phases)	5			nsec	
T_{CYC}	Clock Period (All Phases)	300		1000	nsec	
T_{MIS}	MIBØØ-MIB15 Input Set-Up Time at Ø1	20			nsec	
T_{MIH}	MIBØØ-MIB15 Hold Time After Ø1	5			nsec	
T_{MOL}	Output Switching Low at Ø2 For MIBØØ-10 and at Ø3 For MIB16			35	nsec	C_L = 50 pf
T_{ROL}	Output Switching Low at Ø1 For MIBØ8-15 by RESET			35	nsec	C_L = 50 pf
T_{JIS}	Time to Set-Up MIB15 Low at Ø4 by The Data Chip For the JUMP Response	20			nsec	
T_{MOH}	Output Switching High at Ø2			50	nsec	C_L = 50 pf
T_{WOL}	WAIT Switching Low at Ø1			35	nsec	C_L = 20 pf
T_{OL1}	Output Switching Low at Ø1; WORD/BYTE			60	nsec	Figure 8-4
T_{OH1}	Output Switching High at Ø1; WORD/BYTE, DATA-OUT			60	nsec	Figure 8-4
T_{OL2}	Output Switching Low at Ø2; DATA-IN, IACK, SYNC	5*		60	nsec	Figure 8-4
T_{OH2}	Output Switching High at Ø2; SYNC, DATA-IN, IACK	5*		60	nsec	Figure 8-4
T_{OL4}	Output Switching Low at Ø4; DATA-OUT			60	nsec	Figure 8-4
T_{WOH}	Output Switching High at Ø4; WAIT			35	nsec	C_L = 20 pf
T_{IVO3}	Input Valid Overlap at Ø3; REPLY, BUSY	10			nsec	
T_{IVO1}	Input Valid Overlap at Ø1; RESET, COMPUTE	10			nsec	
T_{IVO4}	Input Valid Overlap at Ø4; INTØ-INT3	10			nsec	
T_{MOT}	Output Transition start delay Time			20	nsec	C_L = 50 pf
T_{MIT}	Input Transition start Set-Up Time	35			nsec	
T_{WOT}	WAIT Output Hold Time	5			nsec	C_L = 20 pf
T_{AOT}	Address Output Transition Start Delay Time			30	nsec	C_L = 50 pf

*Minimums apply to "SYNC" signal only with C_L = 10 pf.

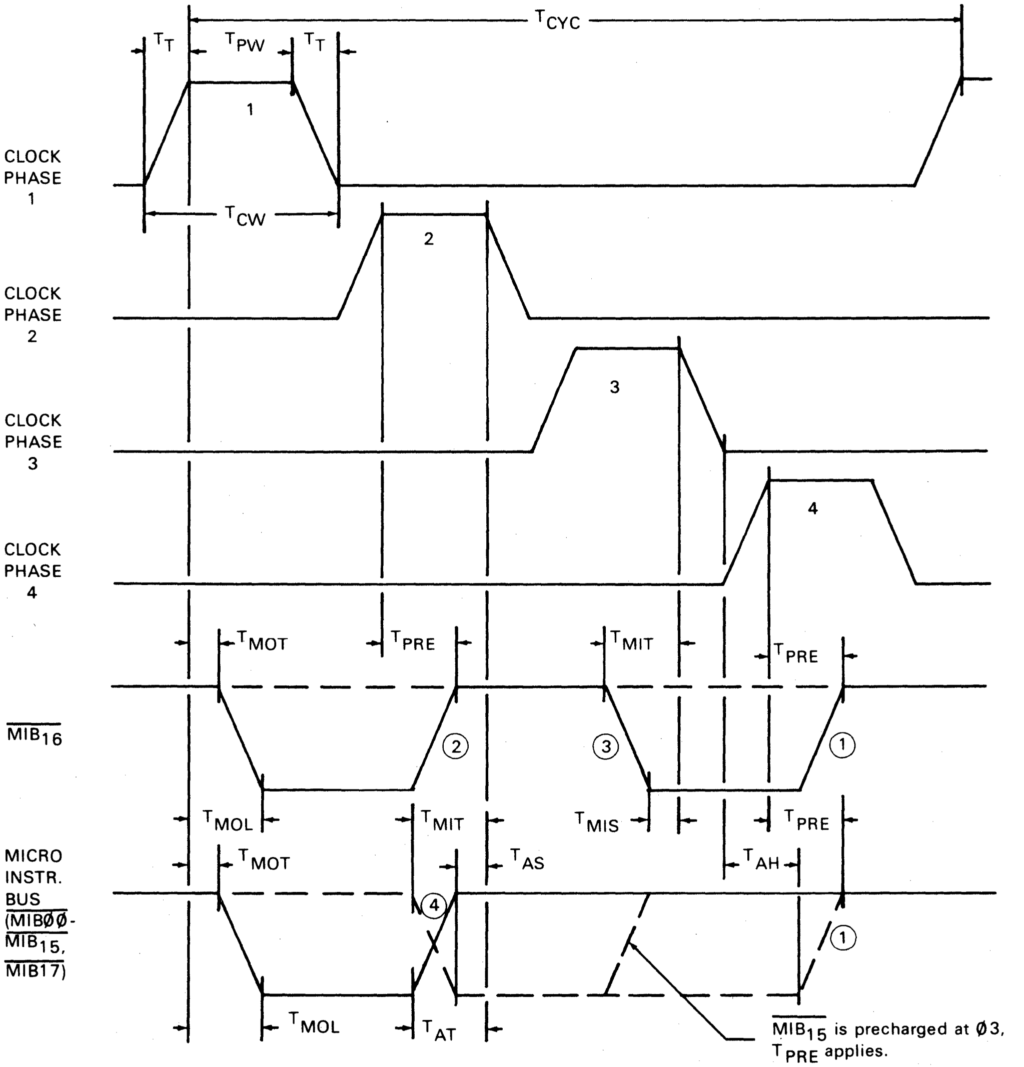
CAPACITANCE

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
$C_{\emptyset 1}$	Clock Phase 1 Capacitance		45	70	pf	See NOTE. VIN = ØV, VSS = ØV, VDD = ØV, VBB = −3.9V F = 1 MHz
$C_{\emptyset 2}$	Clock Phase 2 Capacitance		40	60	pf	
$C_{\emptyset 3}$	Clock Phase 3 Capacitance		45	70	pf	
$C_{\emptyset 4}$	Clock Phase 4 Capacitance		40	60	pf	
C_D	Data Input/Output Pin Capacitance		5.0	10.0	pf	
C_C	Clock to Clock Capacitance		4.0	7.0	pf	

NOTE: Clock Capacitances and Clock To Clock Capacitances are a function of the PTA coding.

8.4 CP1311B MICROPROCESSOR MICROM CHIP

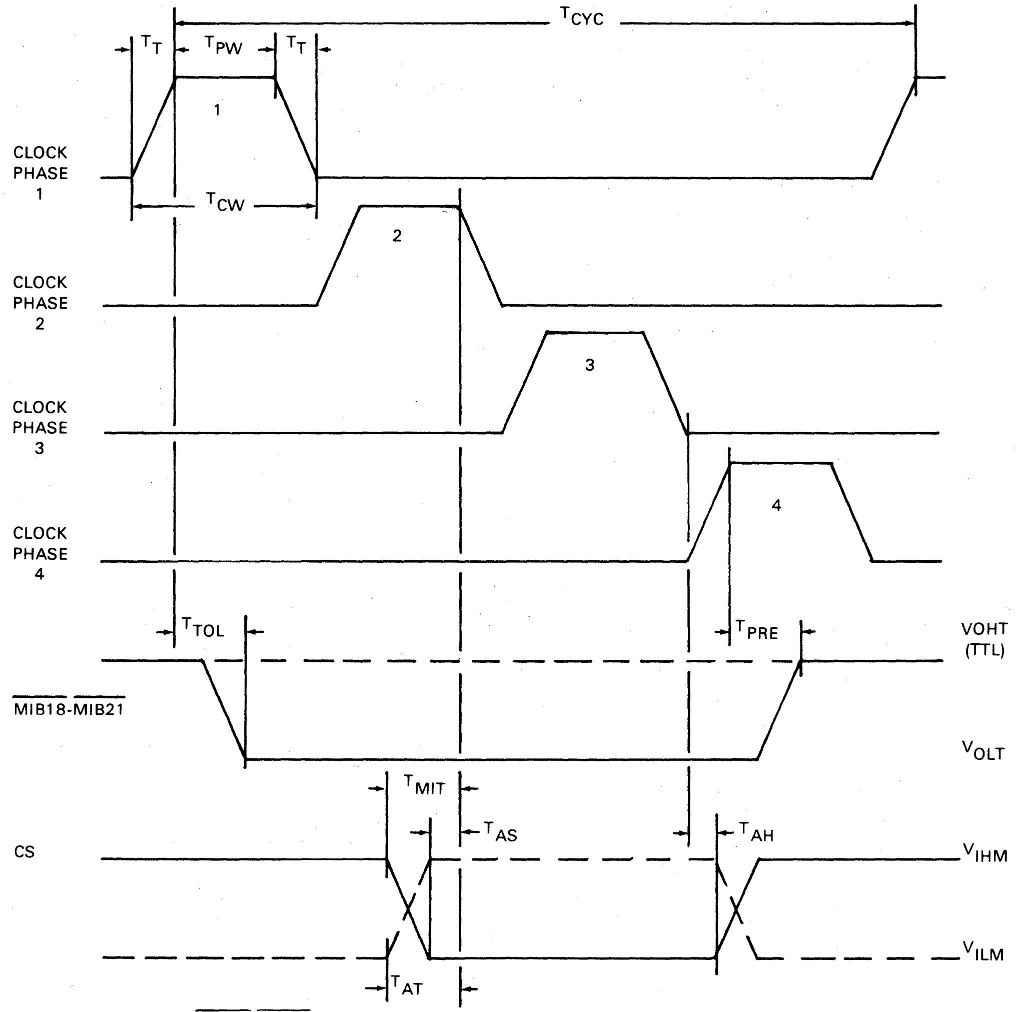
This section describes the electrical interface requirements of the CP1031B Microprocessor MICROM Chip. Adherence to nominal values will ensure reliable operation.



NOTES:

1. $\overline{\text{MIB}}$ lines except $\overline{\text{MIB15}}$ are precharged high at Phase 4.
2. $\overline{\text{MIB16}}$ is also precharged high at Phase 2.
3. $\overline{\text{MIB16}}$ is discharged to disable the $\overline{\text{MIBØØ-MIB15 and MIB18-MIB21}}$ outputs at Ø1 from discharging low.
4. Address input transistions occur only on $\overline{\text{MIBØØ}}$-$\overline{\text{MIB1Ø}}$ lines and Chip Select.
5. Switching times are measured at 10% and 90% of specified levels.

FIGURE 8-5
1631 MIB TIMING



NOTE: TTL outputs ($\overline{\text{MIB18}}$-$\overline{\text{MIB21}}$) are unconditionally driven high at Ø4 and conditionally driven low at Ø1. This conditional low will be valid until the next Ø4.

FIGURE 8-6
1631 TTL TIMING

ABSOLUTE MAXIMUM RATINGS OVER FREE AIR TEMPERATURE RANGE (Unless Otherwise Noted)*

Supply Voltage V_{DD} (See NOTE)	−0.5V to 15V
Supply Voltage V_{CC} (See NOTE)	−0.5V to 15V
Supply Voltage V_{BB} (See NOTE)	−10V tc 1.0V
All Other Pin Voltages (See NOTE)	−1.0V to 15V
Clock Voltage (See NOTE)**	−1.0V to 15V
Operating Free Air Temperature Range	0°C to 125°C
Storage Temperature Range	−55°C to 125°C

NOTE: These voltage values are with respect to V_{SS} Supply Voltage. If V_{BB} is more positive than any other voltage, then I_{BB} must be limited to 10 ma.

*Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum-rating conditions for extended periods may affect device reliability.

**The other three clock voltages must be between 0.5V and −0.6V except for switching overlaps. Not more than one clock may be high at any one time.

Applying power to the part may be any sequence of conditions that do not violate the maximum ratings specified on this sheet.

OPERATING CHARACTERISTICS

T_{CASE} = Ø°C to 70°C, V_{DD} = +12.0V ± .6V, V_{BB} = −3.9 ± .25V, V_{SS} = ØV, V_{CC} = +5V ± .25V

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	COMMENTS AND CONDITIONS
I_L	Leakage Current for any Pin Other than Clock or Power			±10	μA	V_{IN} = 5.25V/ØV
I_{BB}	V_{BB} Supply Current			−500	μA	V_{BB} = −5.25 V
I_{LC}	Clock Leakage Current			±100	μA	V_{CLOCK} = 13.7V/ØV
I_{CCAVE}	Average V_{CC} Operating Current*		10.0	20.0	mA	T_{CYC} = 300 nsec C_L = 25 pf
I_{DDAVE}	Average V_{DD} Operating Current		10.0	20.0	mA	T_{CYC} = 300 nsec C_L = 50 pf
V_{IHM}	Input High Voltage (All Inputs)	4.0		V_{CC}	V	
V_{ILM}	Input Low Voltage (All Inputs)	0.0		0.8	V	
V_{OHM}	Output High Voltage (MOS)	4.35		V_{CC}	V	I_O = −30 μA
V_{OHT}	Output High Voltage (TTL)	2.4		V_{CC}	V	I_O = −50 μA
V_{OLM}	Output Low Voltage (MOS)	V_{SS}		0.4	V	I_O = 100 μA
V_{OLT}	Output Low Voltage (TTL)	0.0		0.4	V	I_O = 1.8 ma
V_{OV}	Overlap Voltage of Any Two Adjacent Clock Phases	0.0		3.0	V	
V_{CH}	Clock High Voltage (SEE NOTE)	11.8 12.0		13.0 13.7	V	V_{DD} = 11.4 V_{DD} = 12.6
V_{CL}	Clock Low Voltage (SEE NOTE)	−0.6		0.5	V	

NOTE: Linear interpolation applies for V_{CH} when V_{DD} is between 11.4V and 12.6V. No overshoot or undershoot allowable.

* Note: The majority of this current is used to precharge the output capacitance, C_L; and therefore, is proportional to the C_L precharged by the MICROM and the frequency of discharge.

AC CHARACTERISTICS

T_{CASE} = Ø°C to 70°C, V_{DD} = +12V ± .6V, V_{BB} = −3.9V ± .25V, V_{SS} = ØV, V_{CC} = +5.ØV ± 0.25V

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
T_{PW}	Clock Width High (All Phases)	55		240	nsec	
T_{CW}	Clock Width Low (All Phases)	75		300	nsec	
T_T	Clock Transition Time (All Phases)	5			nsec	
T_{CYC}	Clock Period (All Phases)	300		1000	nsec	
T_{MOL}	Output Propagation Delay from Ø1 Clock			35	nsec	C_L = 50 pf
T_{PRE}	Time to Precharge Outputs High			55	nsec	C_L = 25 pf
T_{MIS}	Input Set-Up Time on $\overline{MIB16}$ at Phase 3	20			nsec	
T_{TOL}	TTL Out Switching Low			55	nsec	Figure 8-6
T_{AS}	Address Set-Up Time	5			nsec	
T_{AH}	Address Hold Time	Ø			nsec	
T_{MOT}	Output Transition Start Delay Time			20	nsec	C_L = 50 pf
T_{MIT}	Input Transition Start Set-Up Time	35			nsec	
T_{AT}	Address Transition Start Set-Up Time	25			nsec	

CAPACITANCE

SYMBOL	CHARACTERISTIC	MIN	TYP	MAX	UNITS	CONDITIONS
$C_{\emptyset 1}$	Clock Phase 1 Capacitance		20	50	pf	V_{IN} = ØV, V_{SS} = ØV, V_{DD} = ØV, $V_{BB} = -3.9V$ f = 1 MHz
$C_{\emptyset 2}$	Clock Phase 2 Capacitance		40	60	pf	
$C_{\emptyset 3}$	Clock Phase 3 Capacitance		20	50	pf	
$C_{\emptyset 4}$	Clock Phase 4 Capacitance		50	100	pf	
C_D	Data Input/Output Pin Capacitance		5.0	8.0	pf	
C_C	Clock to Clock Capacitance		3.0	6.0	pf	

SECTION IX

PIN ASSIGNMENTS

9.1 GENERAL

This section describes the pin assignments of the three chips that make up the MP1600 Microprocessor set.

PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL
1	Ø3	11	DALØ8	21	Ø2	31	$\overline{\text{MIBØ7}}$
2	V_{BB}	12	DALØ9	22	WAIT	32	$\overline{\text{MIBØ6}}$
3	DALØØ	13	DAL1Ø	23	$\overline{\text{MIB15}}$	33	$\overline{\text{MIBØ5}}$
4	DALØ1	14	DAL11	24	$\overline{\text{MIB14}}$	34	$\overline{\text{MIBØ4}}$
5	DALØ2	15	DAL12	25	$\overline{\text{MIB13}}$	35	$\overline{\text{MIBØ3}}$
6	DALØ3	16	DAL13	26	$\overline{\text{MIB12}}$	36	$\overline{\text{MIBØ2}}$
7	DALØ4	17	DAL14	27	$\overline{\text{MIB11}}$	37	$\overline{\text{MIBØ1}}$
8	DALØ5	18	DAL15	28	$\overline{\text{MIB1Ø}}$	38	$\overline{\text{MIBØØ}}$
9	DALØ6	19	V_{SS}	29	$\overline{\text{MIBØ9}}$	39	V_{DD}
10	DALØ7	20	Ø4	30	$\overline{\text{MIBØ8}}$	40	Ø1

FIGURE 9-1. CP1611B DATA CHIP PIN ASSIGNMENTS

PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL
1	Ø3	11	$\overline{\text{MIB16}}$	21	Ø2	31	$\overline{\text{MIBØ7}}$
2	V_{BB}	12	REPLY	22	V_{CC}	32	$\overline{\text{MIBØ6}}$
3	I3	13	WAIT	23	$\overline{\text{MIB15}}$	33	$\overline{\text{MIBØ5}}$
4	I2	14	DOUT	24	$\overline{\text{MIB14}}$	34	$\overline{\text{MIBØ4}}$
5	I1	15	WB	25	$\overline{\text{MIB13}}$	35	$\overline{\text{MIBØ3}}$
6	IØ	16	IACK	26	$\overline{\text{MIB12}}$	36	$\overline{\text{MIBØ2}}$
7	$\overline{\text{MIB17}}$	17	SYNC	27	$\overline{\text{MIB11}}$	37	$\overline{\text{MIBØ1}}$
8	BUSY	18	DIN	28	$\overline{\text{MIB1Ø}}$	38	$\overline{\text{MIBØØ}}$
9	COMPUTE	19	V_{SS}	29	$\overline{\text{MIBØ9}}$	39	V_{DD}
10	$\overline{\text{RESET}}$	20	Ø4	30	$\overline{\text{MIBØ8}}$	40	Ø1

FIGURE 9-2. CP1621B CONTROL CHIP PIN ASSIGNMENTS

PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL
1	Ø3	11	$\overline{\text{MIB16}}$	21	Ø2	31	$\overline{\text{MIBØ6}}$
2	V_{BB}	12	$\overline{\text{MIB17}}$	22	V_{CC}	32	$\overline{\text{MIBØ5}}$
3	NC	13	$\overline{\text{MIB18}}$	23	CHIP SELECT	33	$\overline{\text{MIBØ4}}$
4	NC	14	$\overline{\text{MIB19}}$	24	NC	34	$\overline{\text{MIBØ3}}$
5	NC	15	$\overline{\text{MIB2Ø}}$	25	NC	35	$\overline{\text{MIBØ2}}$
6	NC	16	$\overline{\text{MIB21}}$	26	$\overline{\text{MIB11}}$	36	NC
7	$\overline{\text{MIB15}}$	17	NC	27	$\overline{\text{MIB1Ø}}$	37	$\overline{\text{MIBØ1}}$
8	$\overline{\text{MIB14}}$	18	NC	28	$\overline{\text{MIBØ9}}$	38	$\overline{\text{MIBØØ}}$
9	$\overline{\text{MIB13}}$	19	V_{SS}	29	$\overline{\text{MIBØ8}}$	39	V_{DD}
10	$\overline{\text{MIB12}}$	20	Ø4	30	$\overline{\text{MIBØ7}}$	40	Ø1

FIGURE 9-3. CP1631B MICROM CHIP PIN ASSIGNMENTS

Western Digital

WESTERN DIGITAL CORPORATION
3128 Redhill Ave. • Newport Beach, CA 92663
(714) 557-3550 • TWX 910 595-1139

Printed U.S.A. 1M 10/77