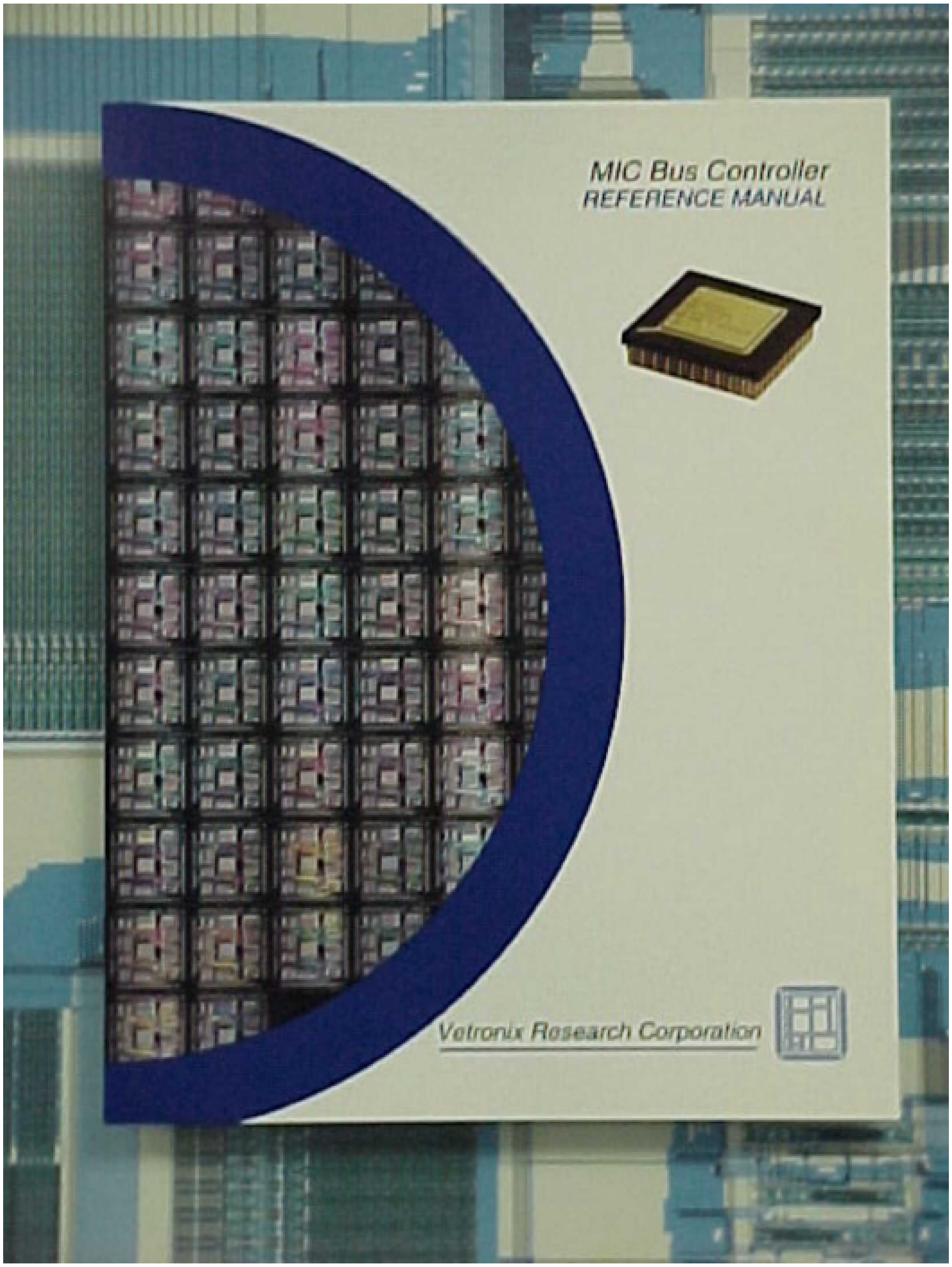

MIC Bus Controller
REFERENCE MANUAL
Vetronix Research Corporation

NOTICE

TECHNICAL SUPPORT

Technical support is available from Vetronix.
Phone number: 248-393-2749
E-mail address: vetronix@aol.com

REVISION HISTORY

Part Number	Product Version	Product Release Date	Document Version	Date	Remarks
MIC-320	V3.0	1992	Draft 1995	11/95	Document Reformatted
MIC-320	V3.0	1992	V1.0	1/97	Added Preface
MIC-320	V3.0	1992	V1.1	11/99	PDF Generation
MIC-320FC		1992	V1.2	1/03	MIC-320FC no longer available

PREFACE

This document defines the MIC Bus Architecture and the electrical interface characteristics for the MIC-320. The MIC Bus architecture is defined followed by the MIC-320 communications protocol definition. Detailed operations of each mode are described in the Modes of Operation Section. All electrical interface characteristics required to develop, use, or evaluate MIC Bus modules are contained herein.

TABLE OF CONTENTS

MIC-320

MIC BUS INTERFACE CONTROLLER

MIC FEATURES

- Single Chip Used as Serial Bus Controller or Remote Modules
- Remote Modules Operate Without a Microprocessor
- Inexpensive, High Performance Alternative to MIL-STD-1553B and ARINC 629
- Simple Command/Response Protocol
- Manchester Encoding and Decoding
- True Data Bus Redundancy
- Built-in Self-test
- Critical Logic Redundancy
- 2.0 Megabit per Second Maximum (1.33 Mb/s Military) Serial Data Bus Rate
- Message Length; 3-Bit Sync, 32-Bit Data, 1-Bit Parity
- 64 Remote Modules Addressable per Bus
- 32 Devices Directly Addressable per Module
- Interfaces to 16 or 32-Bit Microprocessors
- Programmable Control Timers
- Automatic Hand-off Between Bus Controller and Alternate Controller
- Screened to Military Standards
- Complete Hardware/Software Development Tools Available

DESCRIPTION

The MIC is the core component of a time-division multiplexed serial data bus control system. The device is a highly integrated, fault tolerant, and inexpensive component that can provide the two-way serial data interface between several remote modules in a power/data control and management system using a common single-channel transmission medium. The MIC is capable of handling processor interface, command/response serial communication, and direct input/output control functions. The implementation of built-in self-test, critical logic redundancy, and a robust protocol make the MIC ideally suited for applications requiring reliable load and data management. The MIC's highly reliable and high speed data communications enables the MIC Bus to be used as a true serial control bus that transfers control and sensory data directly between a centralized bus controller and remotely located loads and sensors.

The MIC provides an interface to and from the serial bus and a 16 or 32-bit parallel connection to either a microprocessor in the bus controller mode or input/output devices connected at one of several remote modules. The MIC Bus connection is demonstrated in the block diagram shown in

Figure 1. A redundant serial data bus transfers 32-bit Manchester encoded data between up to 64 modules at a rate of 2.0 megabit per second.

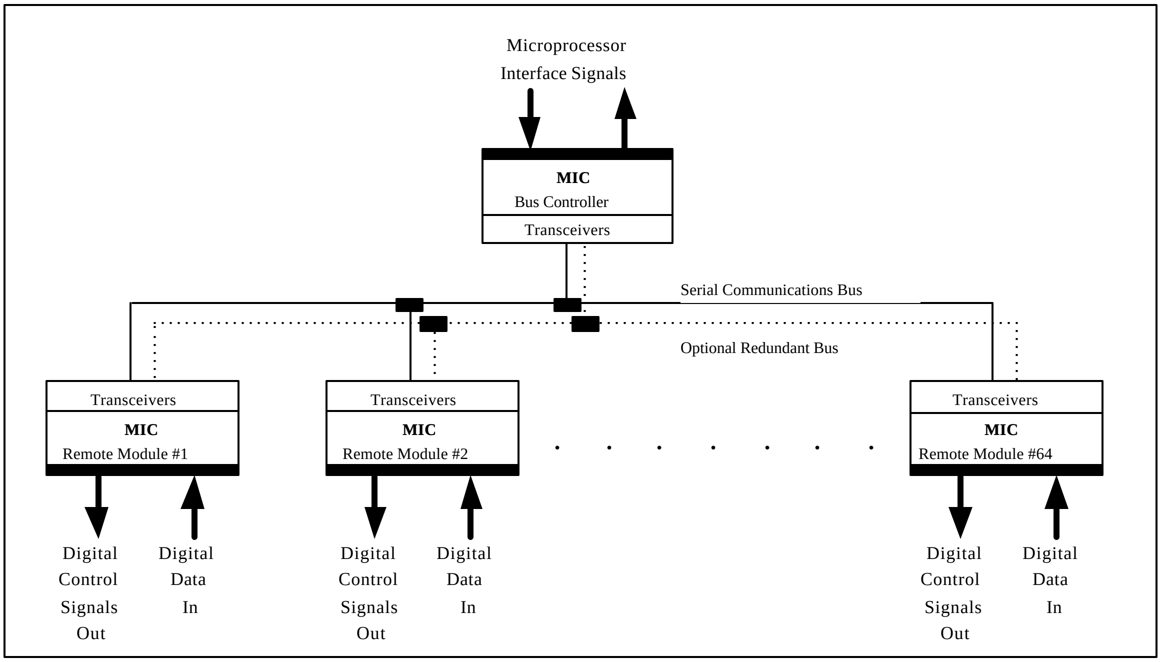


Figure 1. MIC Bus Block Diagram

The MIC is a high performance VLSI device initially developed for military applications that demand high levels of data integrity, redundant functionality, and components that can operate in harsh environments. The military version of the MIC, the MIC-320GM, is packaged in an 84-pin ceramic pin grid array and is characterized for operation from -55°C to +125°C. The MIC-320GM is processed to be fully compliant with paragraph 1.2.1 of MIL-STD-883 and has passed several stringent application and field tests including the United States Army's M1A2 main battle tank operational and nuclear effects qualifications. See Figure 2 below for a block diagram of the MIC.

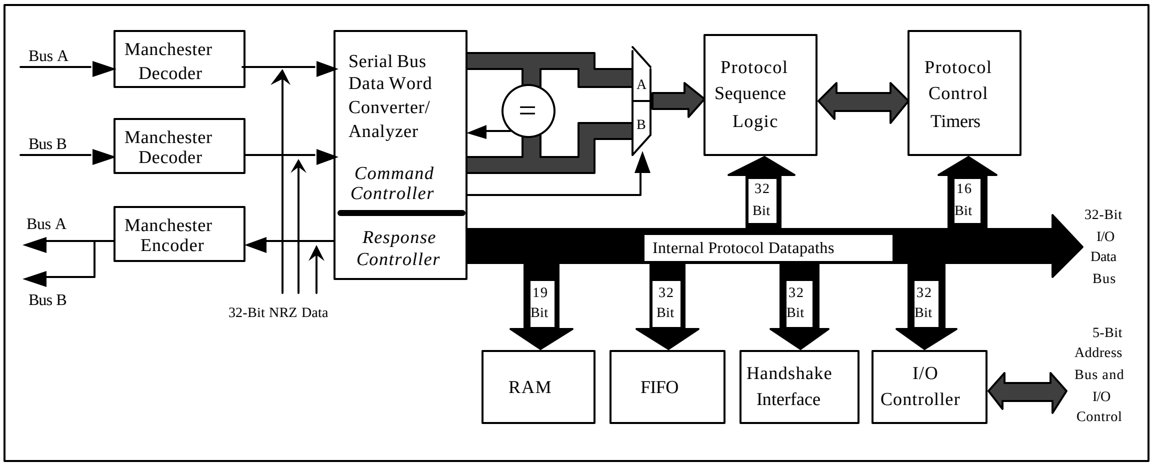


Figure 2. MIC Block Diagram

PIN DESCRIPTION

The MIC has a total of 84 pins. Each pin is shown in Figure 3 and summarized in Table 1. A more detailed description of each pin's function begins on page 6 of this document.

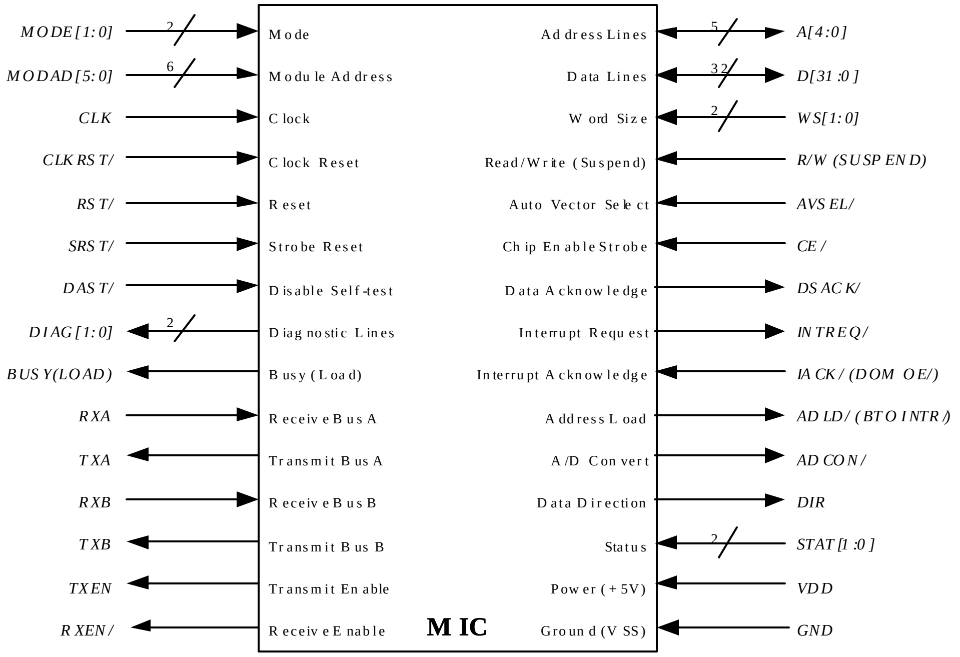


Figure 3. MIC Pin Assignment

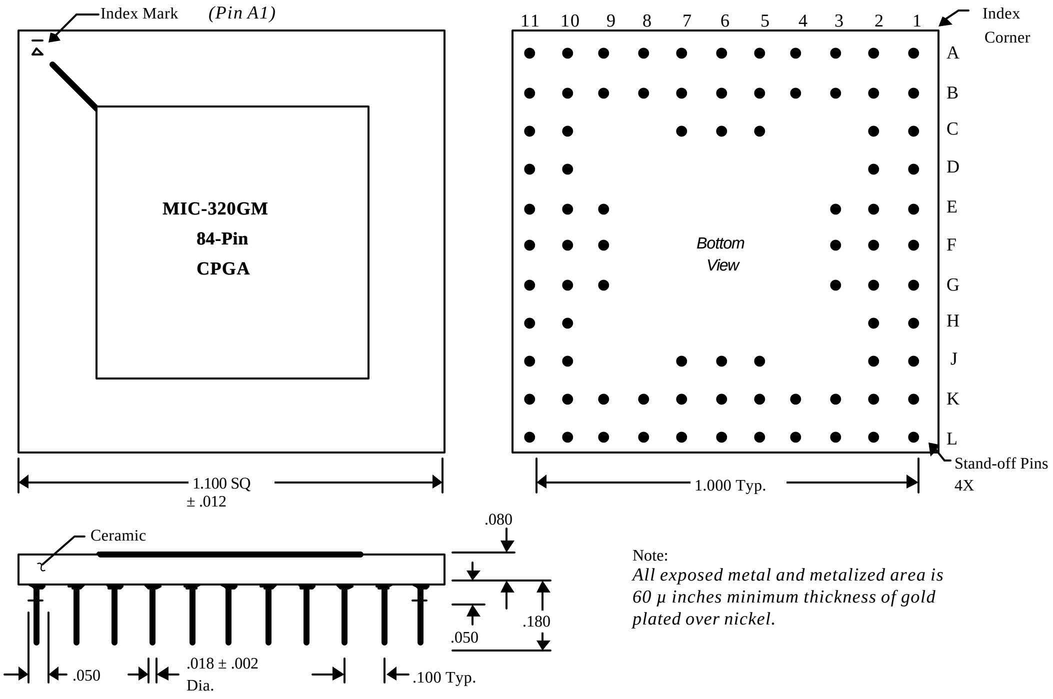
Table 1. Pin Description Summary

Name	Description
MODE[1:0]	2-bit mode select:
	MODE1 *MODE0* MODE
	0 0 Processor Interface [PIM]
	0 1 Remote Switch [RSM]
	1 0 Data Input [DIM]
	1 1 Data Output [DOM]
MODAD[5:0]	6-bit module address
CLK	MIC clock input, 16 MHz typical
CLKRST/	System clock reset, tie to *VDD*
RST/	Reset, initialize, run self-test
SRST/	Strobe reset, tie to *RST/*
DAST/	Disable self-test
DIAG[1:0]	Diagnostic status:
	DIAG1 *DIAG0* STATUS
	0 0 Self-test disabled
	0 1 Self-test failed
	1 0 Self-test passed
	1 1 Self-test running
BUSY	MIC is initializing. In PIM standby mode
(LOAD)	Output Load for *D[15:0]* [RSM/DIM]
RXA	Receive data from Bus A
TXA	Transmit data on Bus A
RXB	Receive data from Bus B
TXB	Transmit data on Bus B
TXEN	Transmit enable
RXEN/	Receive enable

Name	Description
A[4:0]	5-bit bidirectional address bus
D[31:0]	32-bit bidirectional data bus
WS[1:0]	Word size:
	WS1 *WS0* LENGTH
	0 0 Long Word
	1 0 Word
	0 1 not used
	1 1 not used
R/W	Read-write: 0=write, 1=read [PIM]
(SUSPEND)	1=Suspend input cycle [RSM, DIM, DOM]
AVSEL/	Auto vector select: 0=Generate number
CE/	Chip enable (LOW in PIM only) [PIM]
DSACK/	Data transfer/size acknowledge [PIM]
INTREQ/	Interrupt request [PIM]
IACK/	Interrupt acknowledge [PIM]
(DOM OE/)	Output enable for *D[31:16]* [DOM]
ADLD/	Address valid /load pulse
(BTO INTR/)	Bus Time-out Interrupt [PIM]
ADCON/	Analog to digital convert pulse
DIR	*D[15:0]* direction: 0=input, 1=output
	Data in Multiple Receive Buffer [PIM]
STAT[1:0]	32 2-bit status (loaded in MIC RAM)
VDD	Power (+5V)
GND	Ground (VSS)

PACKAGE DIMENSIONS AND PIN ASSIGNMENT

The MIC-320GM is packaged in a hermetically sealed 84-pin ceramic pin-grid-array. The physical dimensions of the package are shown in Figure 4 and the pin assignments are shown in Table 2 below.



Note:
All exposed metal and metalized area is 60 µ inches minimum thickness of gold plated over nickel.

Figure 4. MIC-320GM Package Dimensions

Table 2. MIC-320GM Pin Assignment

Pin Name	Pin No.	I/O	Pin Name	Pin No.	I/O	Pin Name	Pin No.	I/O	Pin Name	Pin No.	I/O
MODE0	C7	I	*RXEN/*	B1	O	*D15*	L2	I/O	*CE/*	B9	I
MODE1	B7	I	*A0*	G3	I/O	*D16*	G10	I/O	*DSACK/*	K3	O
MODAD0	A6	I	*A1*	H1	I/O	*D17*	G9	I/O	*INTREQ/*	F3	O
MODAD1	A4	I	*A2*	H2	I/O	*D18*	H11	I/O	*IACK/*	F1	I
MODAD2	B4	I	*A3*	J1	I/O	*D19*	H10	I/O	*ADLD/*	C5	O
MODAD3	A3	I	*A4*	K1	I/O	*D20*	J11	I/O	*ADCON/*	A5	O
MODAD4	A2	I	*D0*	L9	I/O	*D21*	K11	I/O	*DIR*	E3	O
MODAD5	B3	I	*D1*	K8	I/O	*D22*	J10	I/O	*STAT0*	C10	I
CLK	C6	I	*D2*	L8	I/O	*D23*	K10	I/O	*STAT1*	A11	I
CLKRST/	F2	I	*D3*	J7	I/O	*D24*	D11	I/O	*VDD*	L1	+5V
RST/	G2	I	*D4*	K7	I/O	*D25*	F11	I/O	*VDD*	K9	+5V
SRST/	G1	I	*D5*	L7	I/O	*D26*	E10	I/O	*VDD*	B10	+5V
DAST/	B2	I	*D6*	L6	I/O	*D27*	E11	I/O	*VDD*	A8	+5V
DIAG0	A10	O	*D7*	J6	I/O	*D28*	E9	I/O	*VDD*	A7	+5V
DIAG1	A9	O	*D8*	J5	I/O	*D29*	F9	I/O	*VDD*	A1	+5V
BUSY	C2	O	*D9*	L5	I/O	*D30*	F10	I/O	*GND*	J2	GND
RXA	D2	I	*D10*	K5	I/O	*D31*	G11	I/O	*GND*	K2	GND
TXA	E2	O	*D11*	K6	I/O	*WS0*	C11	I	*GND*	L10	GND
RXB	D1	I	*D12*	L4	I/O	*WS1*	B11	I	*GND*	L11	GND
TXB	E1	O	*D13*	K4	I/O	*R/W*	D10	I	*GND*	B8	GND
TXEN	C1	O	*D14*	L3	I/O	*AVSEL/*	B5	I	*GND*	B6	GND
RXB	D1	I	*D12*	L4	I/O	*WS1*	B11	I	*GND*	L11	GND

The MIC-320FC is packaged in an 84-pin plastic J-leaded chip carrier (PLCC). The physical dimensions of the package are shown in Figure 5 and pin assignments are shown in Table 3 below.

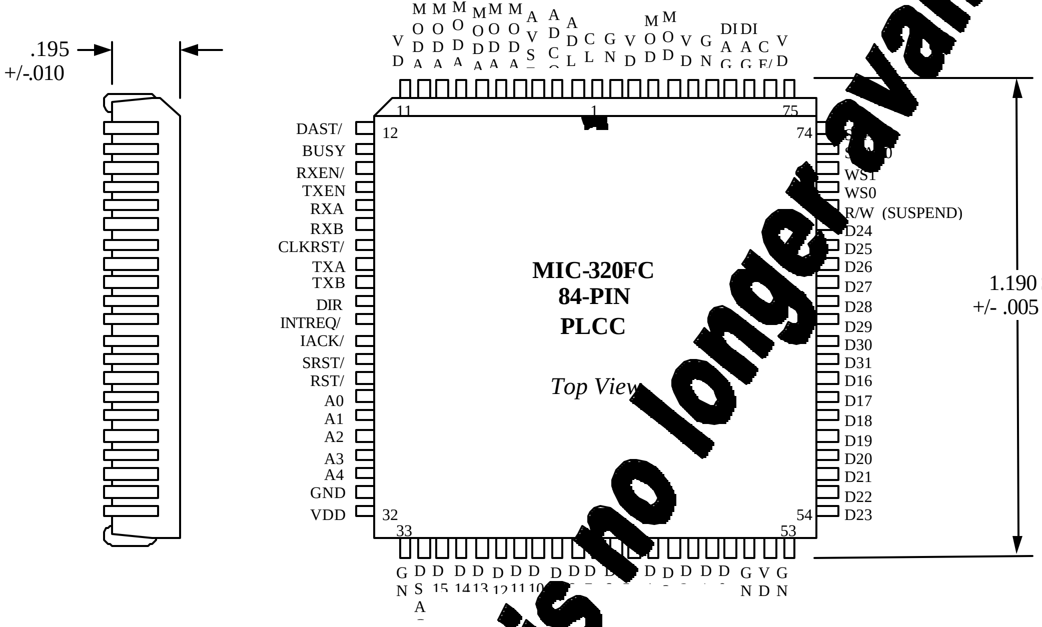


Figure 5. MIC-320FC Package Dimensions

Table 3. MIC-320FC Pin Assignment

Pin Name	Pin No.	I/O	Pin Name	Pin No.	I/O	Pin Name	Pin No.	I/O	Pin Name	Pin No.	I/O
MODE0	81	I	*RXEN/*	[illegible]	O	*D15*	35	I/O	*CE/*	76	I
MODE1	82	I	*A0*	26	I/O	*D16*	61	I/O	*DSACK/*	34	O
MODAD0	5	I	*A1*	[illegible]	I/O	*D17*	60	I/O	*INTREQ/*	22	O
MODAD1	6	I	*A2*	28	I/O	*D18*	59	I/O	*IACK/*	23	I
MODAD2	7	I	*A3*	29	I/O	*D19*	58	I/O	*ADLD/*	2	O
MODAD3	8	I	*A4*	30	I/O	*D20*	57	I/O	*ADCON/*	3	O
MODAD4	9	I	*D0*	50	I/O	*D21*	56	I/O	*DIR*	21	O
MODAD5	10	I	*D1*	49	I/O	*D22*	55	I/O	*STAT0*	73	I
CLK	1	I	*D2*	48	I/O	*D23*	54	I/O	*STAT1*	74	I
CLKRST/	18	I	*D3*	47	I/O	*D24*	69	I/O	*VDD*	11	+5V
RST/	25	I	*D4*	46	I/O	*D25*	68	I/O	*VDD*	32	+5V
SRST/	24	[illegible]	*D5*	45	I/O	*D26*	67	I/O	*VDD*	52	+5V
DAST/	12	[illegible]	*D6*	44	I/O	*D27*	66	I/O	*VDD*	75	+5V
DIAG0	77	[illegible]	*D7*	43	I/O	*D28*	65	I/O	*VDD*	80	+5V
DIAG1	78	O	*D8*	42	I/O	*D29*	64	I/O	*VDD*	83	+5V
BUSY	13	O	*D9*	41	I/O	*D30*	63	I/O	*GND*	31	GND
RXA	16	I	*D10*	40	I/O	*D31*	62	I/O	*GND*	33	GND
TXA	[illegible]	O	*D11*	39	I/O	*WS0*	71	I	*GND*	51	GND
RXB	17	I	*D12*	38	I/O	*WS1*	72	I	*GND*	53	GND
TXB	20	O	*D13*	37	I/O	*R/W*	70	I	*GND*	79	GND
TXEN	15	O	*D14*	36	I/O	*AVSEL/*	4	I	*GND*	84	GND

PIN FUNCTIONS

The MIC is designed to operate as the bus controller or in one of three remote modes of operation. Pins *MODE1* and *MODE0* are used to hard wire the following modes:

MODE1	*MODE0*	OPERATING MODE
0	0	Processor Interface Mode (PIM)
0	1	Remote Switching Mode (RSM)
1	0	Data Input Mode (DIM)
1	1	Data Output Mode (DOM)

Note: A logic HIGH (1) indicates an input pin should be driven near VDD and output pins should read near VDD. A logic LOW (0) indicates an input pin should be driven near GND and output pins should read near GND. All unused inputs must be tied to GND or VDD.

General Pin Description

The following pin descriptions are valid for all modes of MIC operation. See the Electrical Characteristics section for additional timing and functional requirements for each pin.

MODAD[5:0] Input lines used to set MIC's unique module address. Each pin should be tied to *VDD* or *GND* as appropriate.

CLK Input line to be connected to an external clock generator or oscillator. A typical application using a 1.33 Megabit per second bus rate would use a 16 MHz clock source.

CLKRST/ Input line, active LOW, to be connected to *VDD*.

RST/ Input line, active LOW, to be connected to *SRST/* and to external circuit which can generate the appropriate reset signal as shown in Figure 14.

SRST/ Input line, active LOW, to be connected to *RST/* and to external circuit which can generate the appropriate reset signal as shown in Figure 14.

DAST/ Input line, active LOW, held LOW to disable self-test or held HIGH to allow self-test to run on power-up or reset.

DIAG[1:0] Output lines used to indicate the diagnostic results of the MIC self-test.

DIAG1	*DIAG0*	RESULT
0	0	Self-test Disabled
0	1	Self-test Failed
1	0	Self-test Passed
1	1	Self-test In Progress

RXA, RXB Input lines, serial data received from transceiver A and B.

TXA, TXB Output lines, serial data to transceiver A and B.

TXEN Output line, active HIGH, to be connected to the serial transceiver transmit enable pin.

RXEN/ Output line, active LOW, to be connected to the serial transceiver receive enable pin.

Processor Interface Mode

The following will describe the MIC pin functions for proper PIM operation. See the Electrical Characteristics section for additional timing and functional requirements for each pin.

MODE[1:0] Input lines, both are to be held LOW prior to reset.

BUSY Output line, active HIGH, indicates MIC is initializing or in stand-by mode.

A[4:0] Input lines to be connected to the external processor's address bus.

D[31:0] Bi-directional, tri-state data lines to be connected to the external processor's data bus. If interfacing to a 16-bit processor, data lines *D[15:0]* will be active and lines *D[31:16]* should be pulled-up to VDD through resistors.

WS[1:0] Input lines to be connected to the external processor or *VDD* and *GND* as appropriate to indicate the processor transfer word size.

WS1	***WS0***	**SIZE**
0	0	Long Word (32-Bit)
1	0	Word (16-Bit)
0	1	not used
1	1	not used

R/W Input line to be connected to the external processor to indicate a Read (signal is HIGH) or a Write (signal is LOW).

AVSEL/ Input line, active LOW to be connected to an external device or *VDD/GND* for desired interrupt processing. If set LOW, the MIC will place an 8-Bit vector type number on the data lines during an interrupt cycle. If HIGH, no vector number will be placed on the data bus.

CE/ Input line, active LOW, to be connected to the appropriate external logic to indicate read and write cycles.

DSACK/ Output line, active LOW, indicates that the MIC has accepted a valid Read or Write from the processor.

INTREQ/ Output line, active LOW, connected to external interrupt arbitration logic. Indicates MIC is requesting an interrupt.

IACK/ Input line, active LOW, connected to external interrupt arbitration logic. A LOW indicates the *INTREQ/* or *BTO INTR/* interrupt has been recognized.

ADLD/ Output line, active LOW, may be connected to the external interrupt
(*BTO INTR/*) arbitration logic. Indicates that a MIC bus time-out has occurred.

ADCON/ Output line, active LOW, no connection.

DIR Output line, active HIGH, indicates that data is in the multiple receive buffer or transmit buffer. *DIR* is not set HIGH when data is in the single receive buffer.

STAT[1:0] Input lines to be connected to *VDD*.

Remote Switch Mode

The following will describe the MIC pin functions for proper RSM operation. See the Electrical Characteristics section for additional timing and functional requirements for each pin.

MODE[1:0] Input lines, *MODE1* to be connected to *GND*, *MODE0* connected to *VDD* prior to reset.

BUSY Output line, active HIGH, indicates the MIC is initializing. Can be used (*LOAD*) to load *D[15:0]* data into external logic if in RSM/DIM combination mode.

A[4:0] Output lines, used to address 32 2-bit (*STAT[1:0]*) data pairs for storage in the MIC's internal memory. *A[4:0]* may be used to address 32 16-Bit data locations for storage if in the RSM/DIM.

D[31:0] Output lines, 32 individually controlled digital outputs. *D[15:0]* may be used as inputs if in RSM/DIM combination mode.

WS[1:0] Input lines to be connected to *GND*.

R/W Input line, when LOW *R/W* indicates to the MIC that the data on inputs (*SUSPEND*) *STAT[1:0]* is valid for the current address *A[4:0]*. If the *R/W* signal is brought HIGH the MIC will suspend the input cycle and wait until *R/W* (*SUSPEND*) is brought LOW before reading *STAT[1:0]* or *D[15:0]*.

AVSEL/ Input line to be connected to *VDD*.

CE/ Input line to be connected to *VDD*.

DSACK/ Output line, no connection.

INTREQ/ Output line, no connection.

IACK/ Input line to be connected to *GND*.

ADLD/ Output line, active LOW, indicates that the address on *A[4:0]* is valid.

ADCON/ Output line, active LOW, no connection in base RSM. May be used to inform external logic connected to *A[4:0]* to begin data conversion if in RSM/DIM.

DIR Output line indicates direction of *D[15:0]* data lines. 0 = input, 1= output.

STAT[1:0] Input lines can be connected to external bits corresponding to *A[4:0]*.

Data Input Mode

The following will describe the MIC pin functions for proper DIM operation. See the Electrical Characteristics section for additional timing and functional requirements for each pin.

MODE[1:0] Input lines, *MODE1* to be connected to *VDD*, *MODE0* connected to *GND* prior to reset.

BUSY Output line, active HIGH, indicates that the MIC is initializing.

A[4:0] Output lines, used to address 32 16-bit (*D[15:0]*) data words for storage in the MIC's internal memory.

D[31:0] Input lines, *D[15:0]* are used to input 16 bits of digital information that correspond to the current address *A[4:0]*. *D[31:16]* are not used and should be connected to *VDD*.

WS[1:0] Input lines to be connected to the *GND*.

R/W Input line, when LOW indicates to the MIC that the data on input lines (*SUSPEND*) *D[15:0]* is valid for the current address *A[4:0]*. If the *R/W* signal is brought HIGH the MIC will suspend the input cycle and wait until *R/W*(*SUSPEND*) is brought LOW before reading *D[15:0]*.

AVSEL/ Input line to be connected to *VDD*.

CE/ Input line to be connected to *VDD*.

DSACK/ Output line, no connection.

INTREQ/ Output line, no connection.

IACK/ Input line to be connected to *GND*.

ADLD/ Output line, active LOW, indicates that the address on *A[4:0]* is valid.

ADCON/ Output line, active LOW, to be used to inform external logic connected to *A[4:0]* to begin data conversion or present addressed data on *D[15:0]*.

DIR Output line indicates direction (0 = input) of *D[15:0]* data lines.

STAT[1:0] Input lines to be connected to *VDD*.

Data Output Mode

The following will describe the MIC pin functions for proper DOM operation. See the Electrical Characteristics section for additional timing and functional requirements for each pin.

MODE[1:0] Input lines, *MODE1* to be connected to *VDD, MODE0* to be connected to *VDD* prior to reset.

BUSY Output line, active HIGH, indicates that the MIC is initializing.

A[4:0] Output lines, used to address 32 16-bit (*D[15:0]*) data words for storage in the MIC's internal memory. Also used to indicate an external address for *D[31:16]* when *DIR* is HIGH during a data output cycle.

D[31:16] Output lines, latched data lines used to output 16 bits of digital information that corresponds to address *A[4:0]* when *DIR* is HIGH. *D[31:16]* data lines may be tri-stated using *IACK/* (*DOM OE/*).

D[15:0] Input lines, *D[15:0]* are used to input 16 bits of digital information that correspond to address A*[4:0]* when *DIR* is LOW. All unused *D[15:0]* input lines should be connected to *GND*.

WS[1:0] Input lines to be connected to the *GND*.

R/W (*SUSPEND*) Input line, when LOW indicates to the MIC that the data on input lines *D[15:0]* is valid for the current address *A[4:0]*. If the *R/W* signal is brought HIGH the MIC will suspend the input cycle and wait until *R/W* (*SUSPEND*) is brought LOW before reading *D[15:0]*.

AVSEL/ Input line to be connected to *VDD*.

CE/ Input line to be connected to *VDD*.

DSACK/ Output line, no connection.

INTREQ/ Output line, no connection.

IACK/ (*DOM OE/*) Input line, normally held LOW to enable DOM output cycles. When LOW indicates to the MIC that the external logic is ready to accept the next data word from output lines *D[31:16]*. If *IACK/* (*DOM OE/*) is brought HIGH the MIC will tri-state *D[31:16]* and wait until *IACK* (*DOM OE/*) is brought LOW before driving the next output value on *D[31:16]*.

ADLD/ Output line, active LOW, indicates that the address on *A[4:0]* is valid.

ADCON/ Output line, active LOW, to be used to inform external logic connected to *A[4:0]* to begin data conversion or present addressed data on *D[15:0]*.

DIR Output line indicates type of function. 0 = DIM, 1 = DOM.

STAT[1:0] Input lines to be connected to *VDD*.

MIC BUS ARCHITECTURE OVERVIEW

Decentralized power/data distribution systems can be developed using small remotely located modules to control electrical loads and/or to collect and distribute electrical information. Data (i.e., sensor, load, switch, or memory) flowing to and from the remote modules can be manipulated by a central management controller over a single multiplexed serial data bus. The physical construction of this bus could be two wires, a fiber optic cable, a radio link, an infra-red link, or any other serial communication medium. This interconnection reduces the problems and costs encountered in centralized distribution networks where loads and data signals are discretely wired to a central processing unit. Electrical overload protection and switching capabilities for loads can be accomplished using solid-state power controllers or electro-mechanical devices. A decentralized power/data distribution system under computer control allows autonomous system operation, fault tolerant power/data routing, improved diagnostics capabilities, and simplified process or application reconfiguration.

The MIC was developed to support such a distribution and control system. It is a highly integrated component that can provide the serial interface between a bus controller and the various remote modules in a multiplexed serial bus architecture. The term multiplexed is used here in the sense that each of the remote modules on the serial bus can be individually addressed or selected. The MIC integrates, into one component, much of the software and hardware burden that exists in command/response data buses such as MIL-STD-1553B. In addition, the MIC has a truly redundant serial data bus, allowing both buses to transmit or monitor simultaneously. The chip was designed so that remote modules on the bus would not require a microprocessor to control communications or to perform input/output functions. The resulting protocol is very simple to use and does not require system software to perform time-consuming tasks such as timer control, response verification, bus controller to alternate hand-off, or other communications overhead. This highly integrated and proven architecture simplifies system design and integration efforts, allowing more emphasis to be placed on application development.

Physical Data Bus Medium

Serial data bus transceivers are required to provide the appropriate interface between the MIC and multipoint bus transmission lines. The MIC was designed so that a variety of physical data bus techniques could be used. Most current MIC Bus applications use devices that meet the Electronic Industries Association (EIA) RS-485 electrical characteristics standard for a direct coupled linear bus configuration. These widely used, low cost bus transceivers allow redundant twisted and shielded wire pairs to be used as the MIC Bus interconnection medium. MIL-STD-1553B transceivers or other standard differential receiver/drivers could be used to provide an electrical interface to a two wire medium. The MIC can also be easily connected to almost any serial data communication link such as, fiber optic cables, radio links, infra-red links, telephone links, etc.. See the Transceiver Selection section for more information concerning transceivers.

Functional Modes Of Operation

The MIC is designed to be used in both the bus controller and the remote modules. Two of the MIC's pins are used to hard wire the configuration of the MIC to be a bus controller or a remote module. The MIC can also be used in a system where multiple bus controllers are required. In this case, one bus controller will be the master bus controller and the other bus controller(s) will be back-up or alternate bus controller(s).

The MIC operates in one of four base modes or one combination mode of operation.

Mode of Operation	Function
Processor Interface Mode (PIM)	Bus Controller or Alternate Controller
Remote Switch Mode (RSM)	Remote Module
Data Input Mode (DIM)	Remote Module
Data Output Mode (DOM)	Remote Module
Remote Switch Mode/ Data Input Mode (RSM/DIM)	Remote Module, Combination Mode

The MIC has the capability of communicating to 64 remote modules on a 20 Megabit per second serial data bus network. The various modules are typically located near groups of sensors and loads that are at a significant distance from other related sensors and loads in a control system. The remotely located modules are connected to a bus controller by a single serial communication bus. The most common serial link used in conjunction with the MIC is a simple twisted wire pair. A block diagram of the general MIC Bus architecture is shown in Figure 6. The system shown consists of a bus controller, a redundant bus medium, and 64 remote modules.

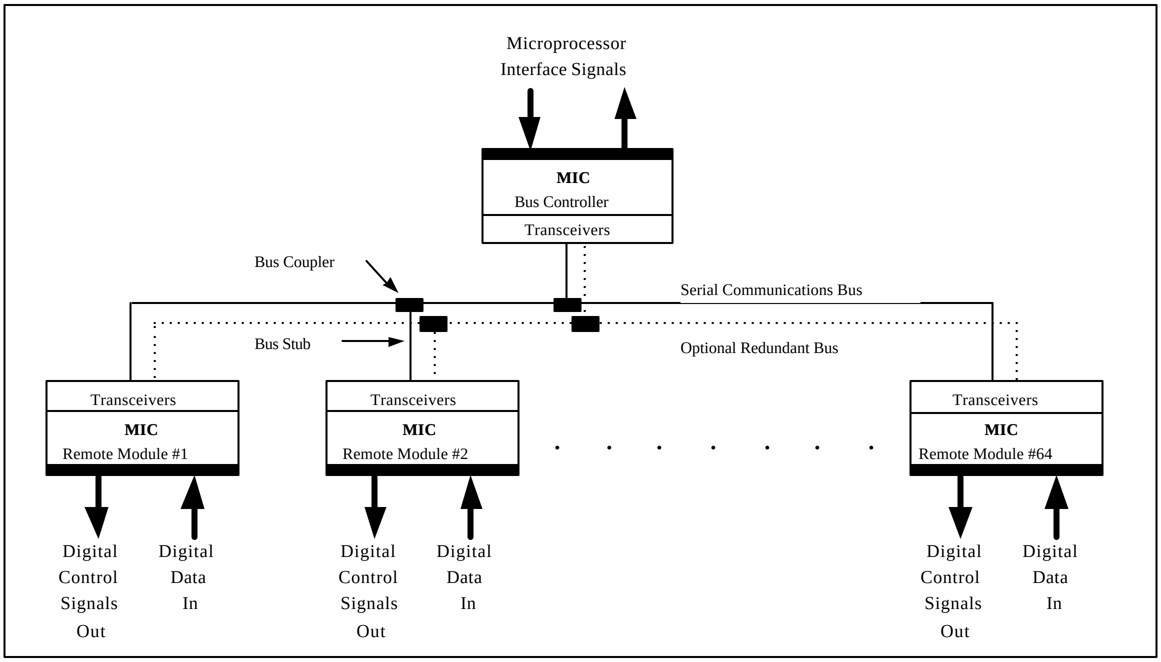


Figure 6. General MIC Bus Architecture

In the **Processor Interface Mode** the MIC is configured as the bus controller and directly connected to a 16 or 32-bit microprocessor, micro-controller or state machine via data, address, and control lines. The MIC is capable of accepting long word (32-bit) or word (16-bit) write and read data transfers to and from the processor. When in PIM, the MIC is in control of the bus and must initiate and monitor all communication. The communication includes messages to and responses from the remote modules or alternate bus controllers. The MIC's processor interface was designed to appear like a piece of memory to which the processor writes the desired command to be transmitted on the serial data bus. The MIC formats and transmits the command, monitors the bus for a response, issues an interrupt when the response has been received, and holds the response until the processor reads it. This simple interface allows software to be written for load and data management systems without the programmer having to be concerned with the details of the serial bus or remote module operation. A MIC configured in the Processor Interface Mode handles all bus system timers, processor interrupts, message response evaluation and control, bus controller to alternate controller hand-off, and other communication overhead.

The **Remote Switch Mode** is used to selectively control up to 32 digital outputs per remote module. These outputs may be used to interface with and control solid-state power switching elements or other digitally controlled electronic or electro-mechanical devices. The 5-bit output address bus is used to repetitively address 32 external 2-bit data locations for storage in the remote module MIC's internal memory. This data, typically load status, can be transmitted from the remote module to the bus controller in a response to an appropriate bus controller command.

In the **Data Input Mode** the 5-bit address bus is used to repetitively address up to 32 external 16-bit words of digital data for storage in the remote module MIC's internal memory. This data can be transmitted from the remote module to the bus controller in a response to an appropriate bus controller command. This mode may be used to directly acquire data from remote analog or digital sensors or from remotely located memory or processors. DIM is automatically combined with the Data Output Mode when a remote module is hard wired for DOM. The Data Input Mode is not typically used because of the additional features available in the DOM including complete DIM capabilities.

The **Data Output Mode** is used to transfer data from the bus controller to remotely located external logic in data blocks of up to 32 16-bit words per command. This mode will permit the MIC to be used in the implementation of output drive circuits and functions. Such circuits could be digital-to-analog converters, solid-state power controllers, or remotely located memory or processors. A remote module configured in this mode also contains the function of the Data Input Mode summarized above.

The **RSM/DIM Combination Mode** is enabled by setting a bit in a Set-up Command and sending it to a remote module configured for the RSM. A remote module in the RSM/DIM mode will be capable of performing both the RSM output and DIM input functions as described above.

Figure 7 shows data connections for each of the MIC's functional modes of operation. The system shown consists of a bus controller, an alternate bus controller, and remote modules configured in RSM, DIM, DOM, and RSM/DIM.

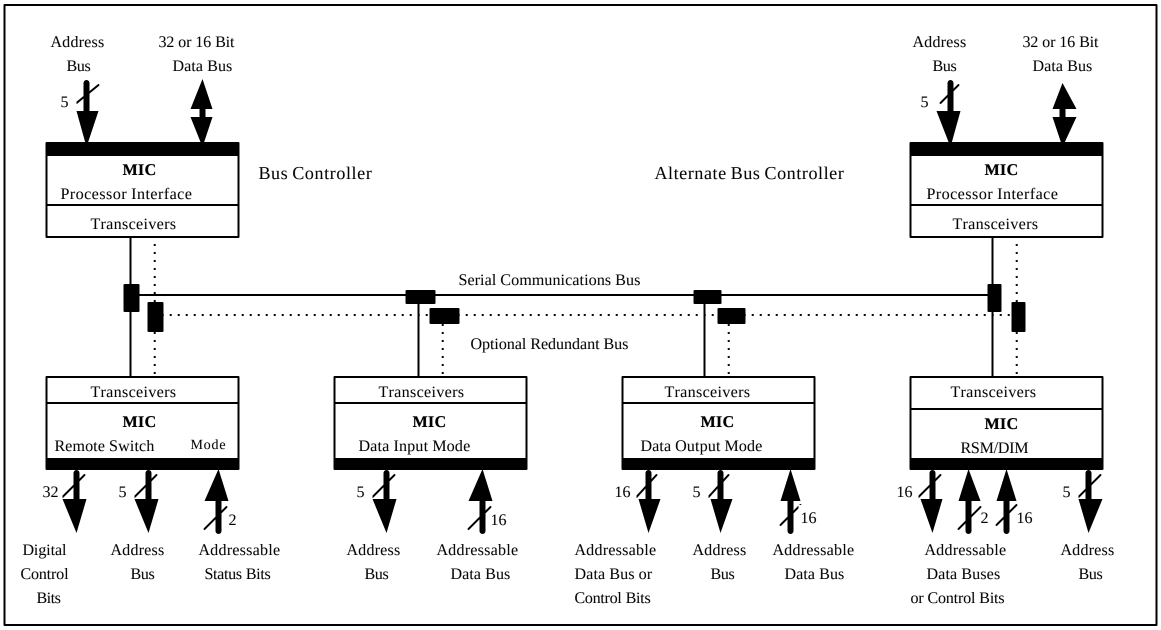


Figure 7. MIC Bus Data Connections

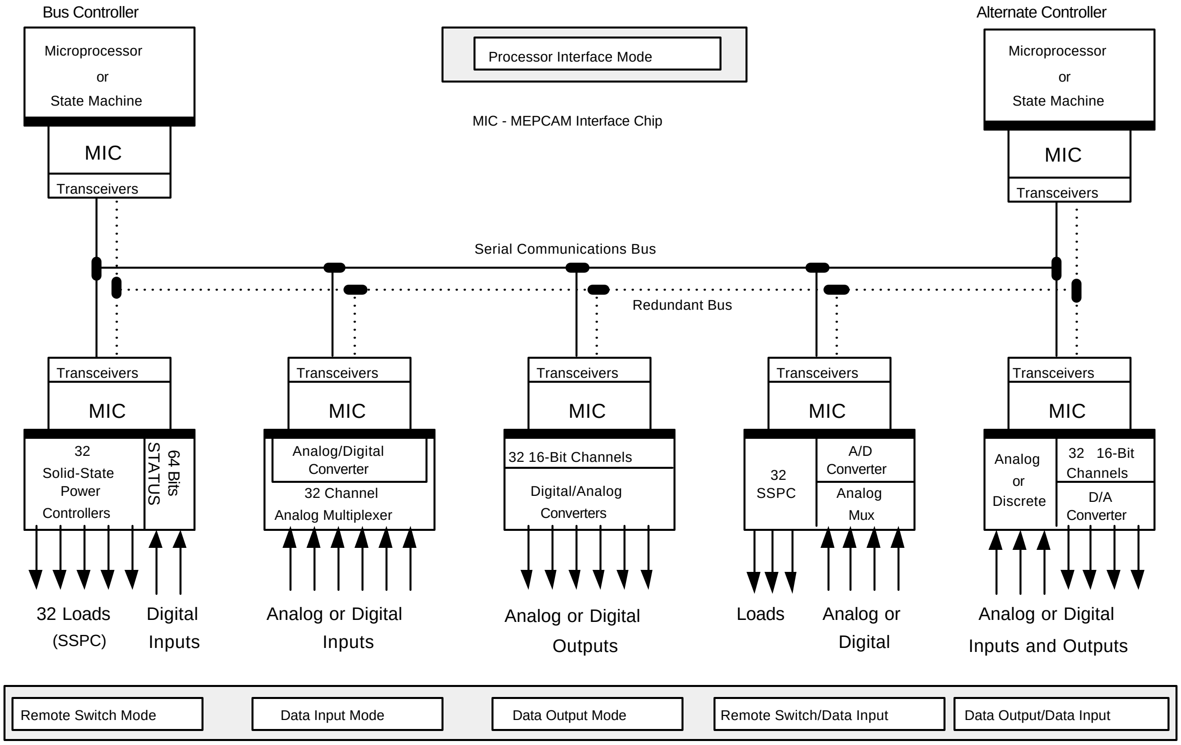


Figure 8. MIC Bus Remote Module Interface Examples

SUMMARY OF MIC PROTOCOL

The MIC implements in hardware much of the software burden that exists in command/response data buses such as MIL-STD-1553B. The MIC Bus command/response protocol was specifically developed to enable a single microprocessor and application program to control an entire power/data control and management system. Therefore, the resulting components are very simple to use and do not require protocol interpretation software or processing.

The MIC protocol is made up of nine basic commands. Eight of the nine commands require a response. The nine commands of the MIC Bus protocol are:

1. Set-up Command
2. Self-test Command
3. Peek Module Command
4. Execute Command (Including Data Words)
5. Peek Single Device Command
6. Peek Single RSM/DIM-Data Device Command
7. Peek Multiple Device Command
8. Peek Multiple RSM/DIM-Data Device Command
9. Broadcast Command (No response)

As shown in Figure 9 the bus controller initiates a communication cycle by transmitting a command across the serial data bus to a specified remote module. All remote modules on the bus will receive and evaluate the command. The remote module that is hard wired to the command's specified module address will respond to the command by immediately transmitting the appropriate response across the serial bus. The bus controller will receive the response and pass the relevant data or bus information to its microprocessor.

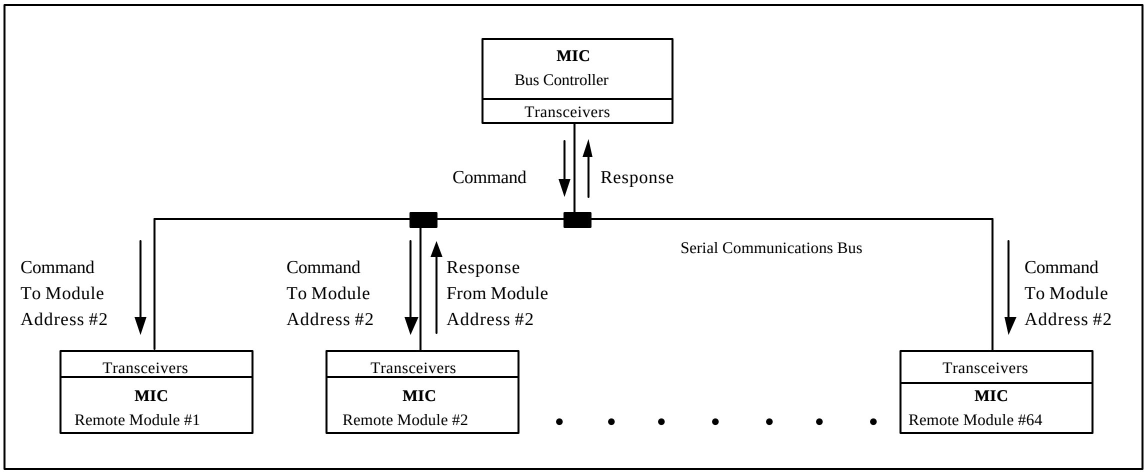


Figure 9. MIC Bus Communication Flow

All MIC Bus commands and responses contain 32-bit Manchester encoded serial data words. Each command, except the Execute Command, is a single serial data word that consists of a synchronization pulse, 32-bit data field, and a parity bit. The Execute Command contains a command word immediately followed by between one and 32 data words. The MIC serial data word format is shown in Figure 10.

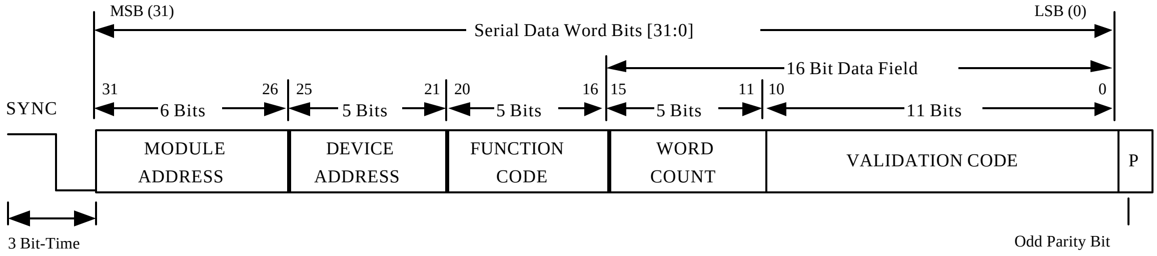


Figure 10. MIC Command/Response Format

Synchronization Pulse

The synchronization pulse is an invalid Manchester signal with a width of three bit-times. The sync pulse will be HIGH for the first one and one-half bit-times and LOW for the following one and one-half bit-times. The MIC will recognize a sync pulse on a normally held HIGH or a normally held LOW serial bus. Further details on the serial wave forms can be found in the Electrical Characteristics section of this document.

Module Address

The 6-bit Module Address field (serial data word bits 31-26) is used to address a command to one of 64 potential modules located on the MIC Bus. The Module Address is contained in every command and response in order to provide maximum data integrity and diagnostics capabilities. When a remote module receives a command from the bus controller it compares the Module Address of the command with input pins *MODAD[4:0]*. If a match occurs, and the command is valid for the particular remote module, the requested response is immediately returned to the bus controller. The remote module's address is contained in the response to allow the bus controller's MIC to validate that the response is from the proper address before passing the data to the microprocessor. A system can be made that contains 64 individually addressed remote modules on a single MIC Bus because the bus controller is not specifically addressed in a response. However, two or more MIC devices configured in the Processor Interface Mode can communicate with each other. In this situation, such as a bus controller communicating with an alternate bus controller, the destinations Module Address must be specified. The total number of remote modules that can be addressed is reduced by the number of addresses required for MIC's configured in PIM if PIM to PIM communication is used. PIM to PIM communication is discussed further in the PIM Operation section of this document.

Device Address

The 5-bit Device Address field (serial data word bits 25-21) is used to address one of 32 locations or devices at a remote module. See each mode of operation descriptions for further details.

Data

The 16-bit Data field (serial data word bits 15-0) is used to send data to or receive data from a remote module. This field is also used to provide word count and validation codes in the appropriate commands.

Word Count

The 5-bit Word Count field (serial data word bits 15-11) is used to indicate to a remote module the number of data words to expect in an Execute Command or the number of words to return in a peek multiple device response.

Validation Code

The 11-bit Validation Code field (serial data word bits 11-0) is used in Execute Commands and Self-test Commands to provide added assurance that a remote module does not alter its output control interface erroneously. The validation codes are 11-bit fixed values defined for each remote mode of operation. The codes are listed below.

VALIDATION CODE	MODE
333 Hex	Remote Switch Mode (RSM)
555 Hex	Data Input Mode (DIM)
777 Hex	Data Output Mode (DOM)
333 Hex	RSM/DIM Combination Mode

Function Code

The 5-bit Function Code field (serial data word bits 20-16) is used to indicate whether the data is a command, response, broadcast, or data word. Table 4 list each code and its corresponding function.

Table 4. Function Codes

Commands	00000	Set-up Command
	00001	Peek Multiple Devices Command
	00010	Execute Command
	00011	Peek Module Command
	00100	Peek Single Device Command
	00101	Self-test Command
	00110	Peek Multiple RSM/DIM-Data Device Command
	00111	Peek Single RSM/DIM-Data Device Command
Normal Response	01000	Set-up Normal Response
	01001	Peek Multiple Devices Normal Response
	01010	Execute Normal Response
	01011	Peek Module Normal Response
	01100	Peek Single Device Normal Response
	01101	Self-test Normal Response
	01110	Peek Multiple RSM/DIM-Data Device Normal Response
	01111	Peek Single RSM/DIM-Data Device Normal Response
Attention Response	10000	*Reserved*
	10001	Peek Multiple Devices Attention Response
	10010	Execute Attention Response
	10011	*Reserved*
	10100	Peek Single Device Attention Response
	10101	Self-test Attention Response
	10110	Peek Multiple RSM/DIM-Data Device Attention Response
	10111	Peek Single RSM/DIM-Data Device Attention Response
Broadcast/Data	11000	*Reserved*
	11001	*Reserved*
	11010	Data Word
	11011	Module Broadcast On
	11100	*Reserved*
	11101	Module Broadcast Off
	11110	*Reserved*
	11111	I'm Alive message

The **Set-up Command** is used to transfer initial system set-up data from the Processor Interface Mode (PIM) to any remote module. After a remote module has compared the command's module address field with its module address pins, the MIC will check the validity of the command, processes the command, load the set-up data, and respond with a Set-up Normal Response. This command is valid for all remote modules.

The **Self-test Command** is used to instruct a remote module to run its self-test routine. After this command is received, the module address and validation code are checked. If a match occurs, the MIC sends back a Self-test Normal Response and starts self-test. If there is a problem with the validation code or transmission, a Self-test Attention Response is sent and the self-test is not performed. This command is valid for all remote modules.

The **Peek Module Command** is used to check a remote module's configuration and diagnostic status. The remote module's internal 16-bit status register is returned in the Peek Module Normal Response. This command is valid for all remote modules.

The **Execute Command** is used to instruct a remote module to perform a specified task. It consists of a command word followed by 1 to 32 data words. The 5-bit word count field in the command word specifies the number of data words to follow. Every Execute Command must contain at least one data word. Once the Execute Command word is received, the module address and validation code is checked. If a match occurs, the data word(s) will be read and stored in the MIC's internal memory. The remote module then transmits back an Execute Normal Response and carries out the requested task. If there is a problem with the validation code or word count, an Execute Attention Response is sent and command is not performed. This command is valid for remote modules using the MIC's Remote Switch Mode (RSM), Data Output Mode (DOM), or RSM/DIM combination mode.

The **Peek Single Device Command** is used to request the status or data from specific devices assigned to a remote module. The module address and device address is sent to a remote module in the Peek Single Device Command. The device's status/data is returned in the Peek Single Device Normal Response. If the device status is not yet valid in the RSM, a Peek Single Device Attention Response will be returned. This command is valid for remote modules using the MIC's RSM, Data Input Mode (DIM), or RSM/DIM combination mode.

The **Peek Single RSM/DIM-Data Device Command** is used to request data from a single device associated with the specified remote module. This command is identical to the Peek Single Device Command except it is only used in conjunction with remote modules operating in RSM/DIM combination mode. The data returned with this command is the data from the DIM input operation of the device and not the status from the RSM input operation. The Peek Single Device Command may be used to request the status information from the RSM input operation when the combination mode is used.

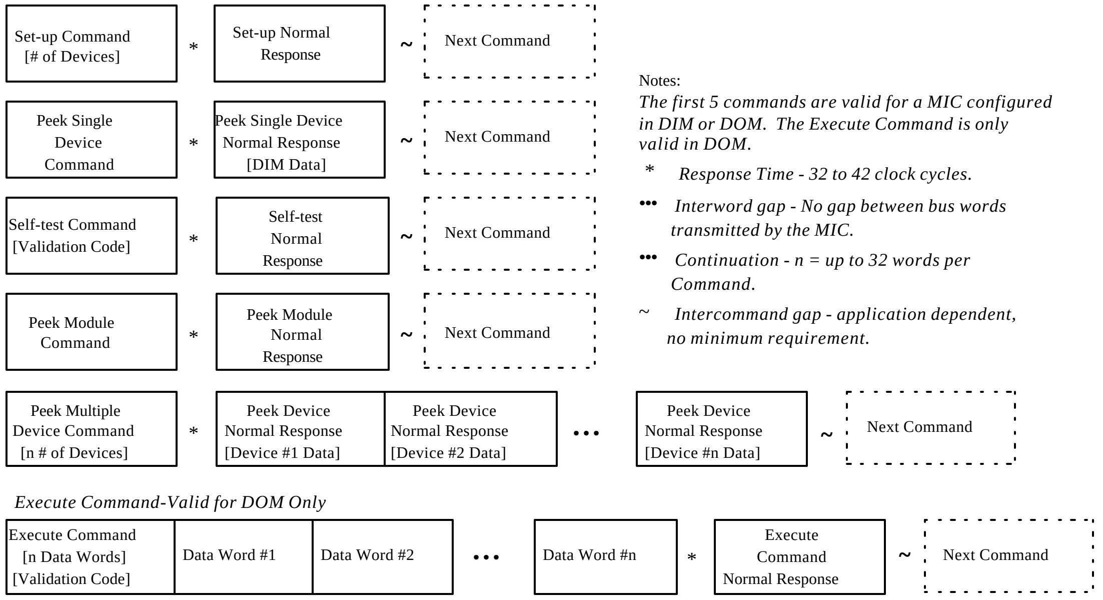
The **Peek Multiple Device Command** is used to request the status or data from multiple devices associated with the specified remote module. A Peek Multiple Device Command has the option of requesting several device responses up to the entire contents of the remote modules status memory (32 locations). The module address and number of devices to check is sent in the Peek Multiple Device Command word. Each device's status or data is returned in a Peek Multiple Device Normal Response. The multiple responses received by the bus controller are placed in the multiple receive buffer and a single interrupt to the processor is issued. If the device status is not yet valid in the RSM, a Peek Multiple Device Attention Response will be returned.

This command is valid for remote modules using the MIC's RSM, DIM, or RSM/DIM combination Mode.

The **Peek Multiple RSM/DIM-Data Device Command** is used to request data from multiple devices associated with the specified remote module. This command is identical to the Peek Multiple Device Command except that it is only used in conjunction with remote modules operating in RSM/DIM combination mode. The data returned with this command is the data from the DIM input operation of the device and not the status from the RSM input operation. The Peek Multiple Device Command may be used to request the status information from the RSM input operation when a combination mode is used.

The **Broadcast Command** is used to instruct a remote module configured in the Remote Switch Mode (RSM) to turn ON or OFF all 32 control line outputs *D[31:0]*. This command is valid only for remote modules using the MIC's Remote Switch Mode or RSM/DIM combination mode. Remote modules do not issue a response to Broadcast Commands in order to allow global functions to be used where more than one RSM's outputs are to be preset or reset to all ON or all OFF. Because no response will be transmitted, several Broadcast Commands can be sent on the bus sequentially. Broadcast Commands should only be used after carefully considering the potential effects of a missed command since no confirming response will be sent by the remote module.

Figures 11 through 13 illustrate typical command/response communication between the bus controller (PIM) and each of the remote modules or alternate bus controller.


Set-up Command [# of Devices]
*
Set-up Normal Response
~
Next Command
Peek Single Device Command
*
Peek Single Device Normal Response [DIM Data]
~
Next Command
Self-test Command [Validation Code]
*
Self-test Normal Response
~
Next Command
Peek Module Command
*
Peek Module Normal Response
~
Next Command
Peek Multiple Device Command [n # of Devices]
*
Peek Device Normal Response [Device #1 Data]
Peek Device Normal Response [Device #2 Data]
• • •
Peek Device Normal Response [Device #n Data]
~
Next Command
Execute Command-Valid for DOM Only
Execute Command [n Data Words] [Validation Code]
Data Word #1
Data Word #2
• • •
Data Word #n
*
Execute Command Normal Response
~
Next Command

Notes:
The first 5 commands are valid for a MIC configured in DIM or DOM. The Execute Command is only valid in DOM.

* *Response Time - 32 to 42 clock cycles.*

••• *Interword gap - No gap between bus words transmitted by the MIC.*

••• *Continuation - n = up to 32 words per Command.*

~ *Intercommand gap - application dependent, no minimum requirement.*

Figure 11. PIM to Data Input Mode or Data Output Mode Transfer Format

Note: *Data may be sent between MIC devices configured in PIM. No preset protocol is defined except that each command must contain the destination's Module Address.*

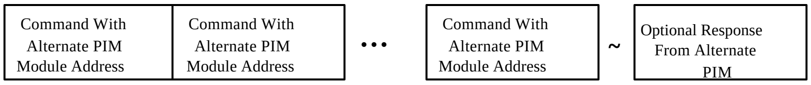
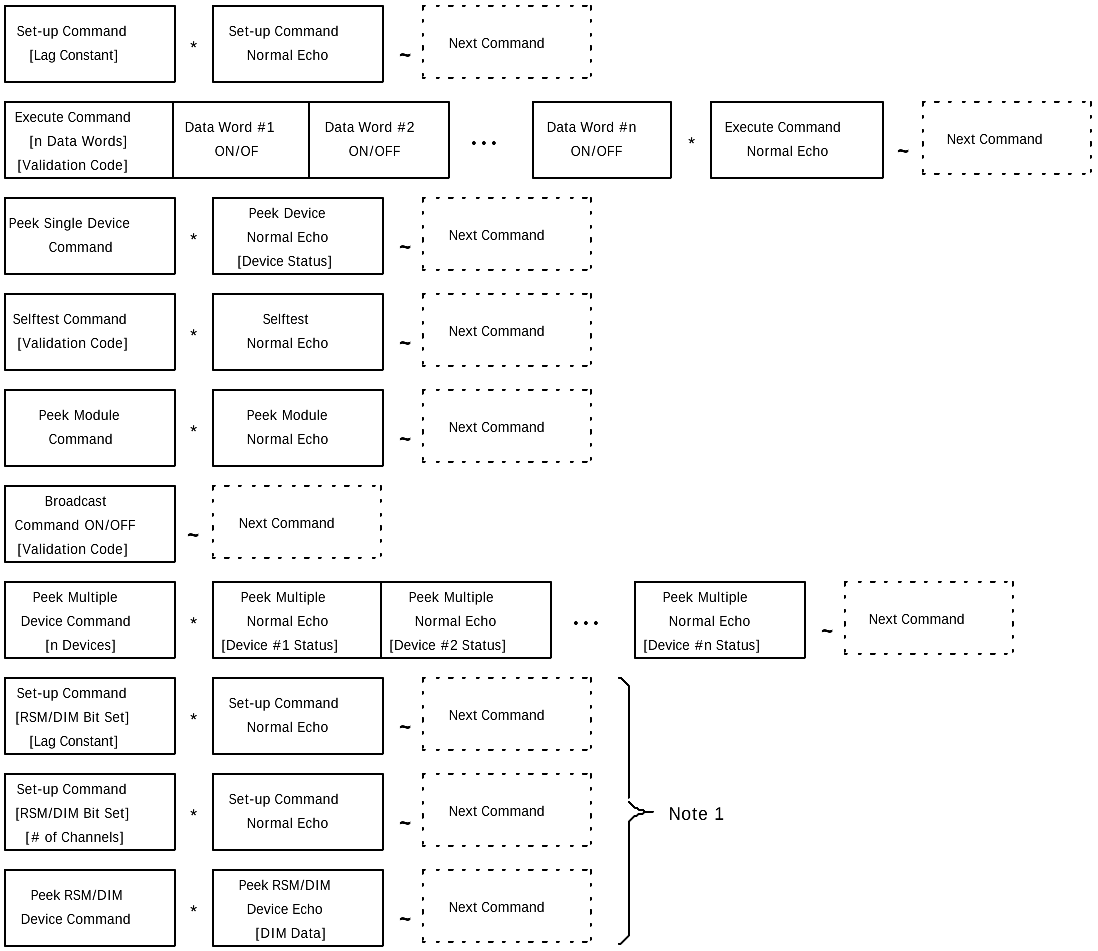

Command With Alternate PIM Module Address
Command With Alternate PIM Module Address
• • •
Command With Alternate PIM Module Address
~
Optional Response From Alternate PIM

Figure 12. PIM to Alternate PIM Transfer Format


Set-up Command [Lag Constant]
*
Set-up Command Normal Echo
~
Next Command
Execute Command [n Data Words] [Validation Code]
Data Word #1 ON/OF
Data Word #2 ON/OFF
• • •
Data Word #n ON/OFF
*
Execute Command Normal Echo
~
Next Command
Peek Single Device Command
*
Peek Device Normal Echo [Device Status]
~
Next Command
Selftest Command [Validation Code]
*
Selftest Normal Echo
~
Next Command
Peek Module Command
*
Peek Module Normal Echo
~
Next Command
Broadcast Command ON/OFF [Validation Code]
~
Next Command
Peek Multiple Device Command [n Devices]
*
Peek Multiple Normal Echo [Device #1 Status]
Peek Multiple Normal Echo [Device #2 Status]
• • •
Peek Multiple Normal Echo [Device #n Status]
~
Next Command
Set-up Command [RSM/DIM Bit Set] [Lag Constant]
*
Set-up Command Normal Echo
~
Next Command
Set-up Command [RSM/DIM Bit Set] [# of Channels]
*
Set-up Command Normal Echo
~
Next Command
Peek RSM/DIM Device Command
*
Peek RSM/DIM Device Echo [DIM Data]
~
Next Command
Note 1

Note 1: *When an RSM is set-up to perform the combination functions of RSM and DIM, load the Valid Lag Constant for the RSM status bits first, then load the number of external channels to monitor for DIM. A Peek RSM/DIM Device Command may then be used to request DIM data. All other commands including Peek Single and Peek Multiple Device Commands can still be used.*

Note 2:

* *Response Time - 32 to 42 Clock cycles. Time is measured from the end of the command parity bit to the beginning of the response SYNC.*

••• *Intermessage gap - No gap between messages transmitted by the MIC. A Remote will begin its response after a 32 Clock gap. Time is measured from the end of parity to the beginning of the next word's LOW half of the Sync pulse.*

••• *Continuation - n = up to 32 Words Per Command.*

~ *Intercommand Gap - dependent on application, no minimum requirement.*

Figure 13. PIM to Remote Switch Mode Transfer Format

CHIP OPERATION

POWER-UP / RESET / SELF-TEST

In order for the MIC to be properly initialized, *RST/* and *SRST/* must be held LOW for a minimum of 24 clock cycles after *VDD* has reached at least 4.5 volts. The reset timing requirement is shown in Figure 14. The *CLKRST/* signal is only used when synchronizing the MIC to VLSI test equipment and should be tied HIGH for normal operation. *RST/* and *SRST/* should be tied together and will be referred to as *RST/* in this section.

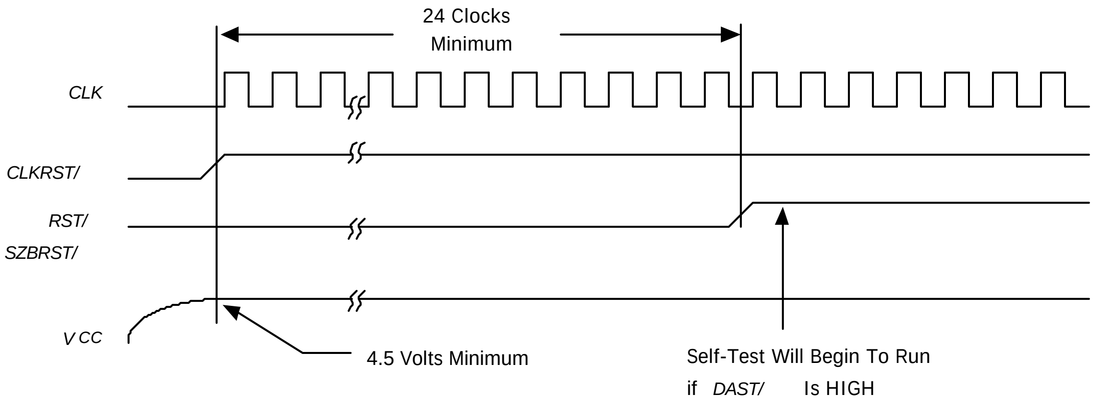


Figure 14. Reset Timing Requirements

During power-up or when the *RST/* signal is asserted LOW, all processing stops. The *BUSY* signal is set HIGH and the chip starts its internal initialization. After *RST/* is asserted HIGH, the device checks its *DAST/* pin for a signal. If *DAST/* is LOW, the device indicates that self-test is disabled on the diagnostic pins *DIAG[1:0]*, sets *BUSY* LOW, and continues normal operation. If the *DAST/* signal is HIGH, the device will begin its internal self-test routine. The self-test takes approximately 10,000 clocks to execute or for the self-test failure time-out to be reached. At the completion of the reset/self-test sequence, the device reads its 6-bit hard wired module address *MODAD[5:0]* and mode select pins *MODE[1:0]*, stores them in memory for future access, and initializes itself. The initialization routine clears all internal latches, registers, and previously loaded set-up parameters. The *BUSY* signal is returned LOW to indicate that self-test and initialization is complete. (NOTE: If in Processor Interface Mode, the *BUSY* signal will be LOW for 2 clock cycles then returned HIGH to indicate that the PIM is in the standby or alternate bus controller mode.) The diagnostic pins *DIAG[1:0]* will reflect the chip's status as shown below.

DIAG1	***DIAG0***	**RESULT**
0	0	Self-test disabled
0	1	Self-test failed (device is bad)
1	0	Self-test passed (device is good)
1	1	Self-test in progress

The MIC will then perform the operation of the selected mode. For example, if in the DIM, the MIC will begin reading and storing the data from the 32 external address locations. The DIM remote module would also be ready to monitor the data bus for appropriate commands from the bus controller.

The reset/self-test process is repeated any time a *RST/* signal occurs, a Self-test Command is received from the serial bus, or a Soft Reset is requested in the Processor Interface Mode.

MODE SELECTION

The base mode of operation is selected by setting the mode select pins as indicated below.

MODE1	***MODE0***	**MODE**
0	0	Processor Interface Mode (PIM)
0	1	Remote Switch Mode (RSM)
1	0	Data Input Mode (DIM)
1	1	Data Output Mode (DOM)

The mode select pins are read and stored in memory during initialization as described in the Power-Up/Reset/Self-test section.

Two other modes of operation are also available. They are combination modes using fundamentals from the base modes. Both of these modes are targeted for remote modules only.

DOM/DIM Combination Mode - This mode is automatically enabled in a remote module that is configured for DOM mode (*MODE[1:0]* is hard wired for DOM).

RSM/DIM Combination Mode - This mode is selected by sending a Set-up Command, with bit 15 set HIGH, to a remote module that is configured for RSM mode (*MODE[1:0]* is hard wired for RSM).

Validation codes are used in Command Words to further verify proper operation of the system. The validation codes are 11-bit fixed values defined for each mode (except PIM) and are transmitted in all command words that request a remote module to alter its outputs or to run self-test. The validation code selection should reflect the base mode of operation. The codes are listed below.

VALIDATION CODE	**MODE**
333 Hex	Remote Switch Mode (RSM)
555 Hex	Data Input Mode (DIM)
777 Hex	Data Output Mode (DOM)
777 Hex	DOM/DIM Combination Mode
333 Hex	RSM/DIM Combination Mode

MODES OF OPERATION

The following sections will provide details on the operation of each of the four base MIC modes and the two combination modes of operation. It may be helpful to review the remote modes of operation before the Processor Interface Mode to get an overall understanding of the command/response operation.

Processor Interface Mode (PIM) Operation

In this mode, the MIC is used to interface the serial data bus to a processor or state machine via data, address, and control lines. The MIC is capable of accepting long word (32-bit) or word (16-bit) write and read data transfers to and from the processor. The MIC's processor interface was designed to appear like a memory device to which the processor writes the desired command to be transmitted on the serial data bus. The MIC transmits the command, monitors the bus for a response, issues an interrupt when the response has been received, and holds the response until the processor reads it. This simple interface allows software to be written for load and data management systems without the programmer having to be concerned with the details of the serial bus or remote module operation. PIM write and read timing is shown in Figures 15 and 16. Detailed timing diagrams are shown in the Electrical Characteristics section of this document.

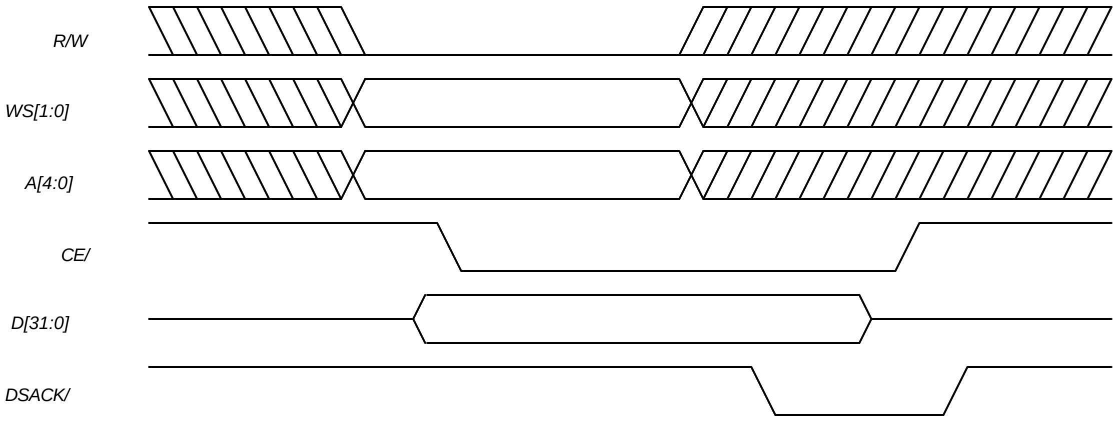


Figure 15. Write-To-MIC Wave form

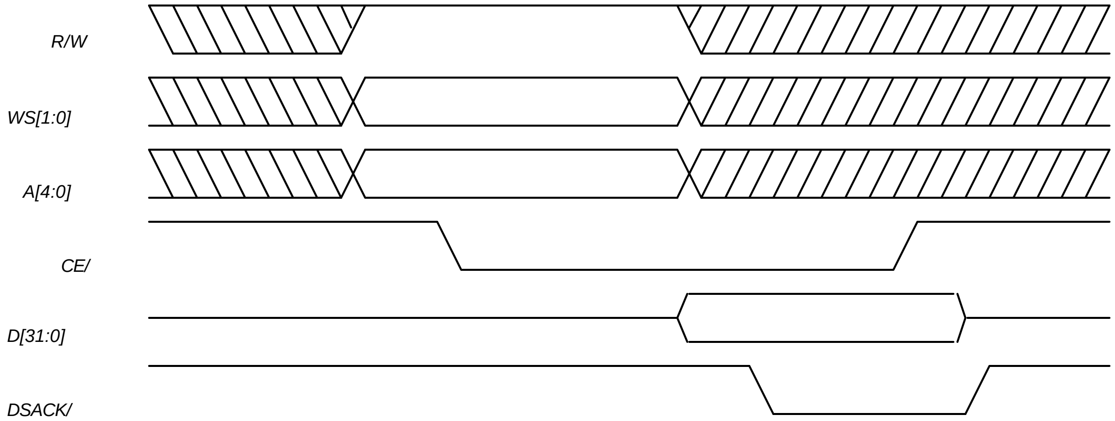


Figure 16. Read-From-MIC Wave form

MIC Control And Status Registers Addressing. The MIC contains 10 control and status registers when operating in the PIM. The MIC address location for each register is shown in Table 5. The processor used must be capable of accessing the MIC's registers using long word (32-bit) or word (16-bit) reads and writes. The size pins *WS[1:0]* inform the MIC of the length of the desired data transfer cycle. *WS[1:0]* is held to 00 during a long word transfer cycle. *WS[1:0]* is held to 10 during a word transfer cycle. The MIC is always capable of long word transfers to or from the selected processor or interface logic. If a word transfer format is selected, two sequential word transfers will be required to complete a read or write to a MIC register that is longer than 16 bits. The first word (16-bit) transfer would use the target register's base address and the second word transfer would use the register's base address plus one. For example, if the processor writes to the MIC's Transmit Buffer using word transfers, it must first write the most significant 16 bits of data to address 01110 then the least significant 16 bits of data to address 01111. If a word transfer is used to write-to or read-from a register of 16 bits or less, a single word transfer with the register's base address plus one is all that is necessary.

TABLE 5. PIM Registers

MIC Address *A[4:0]*	**Register Type**	**Size**	**Active I/O Lines**	**Register Description**
00000	Write Only	16-Bit	*D[15:0]*	Bus Time-out Counter*
00010	Write Only	16-Bit	*D[15:0]*	No Command Time-out Counter*
00100	Write Only	10-Bit	*D]9:0]*	No Response Time-out Counter*
00110	Write Only	10-Bit	*D[9:0]*	Interrupt Ack. Time-out Counter
01000	Write Only	16-Bit	*D[15:0]*	Set-up Register
01010	Read Only	23-Bit	*D[22:0]*	Status Register
01100	Write Only	8-Bit	*D[7:0]*	Base Vector Number
01100	Write Only	6-Bit	*D[13:8]*	Module Address Register
01110	Write Only	32-Bit	*D[31:0]*	Transmit Buffer
10000	Read Only	32-Bit	*D[31:0]*	Receive Buffer/Multiple Rx Buffer

* $\text{Time} = \frac{\text{(Decimal Value of Register) X 240}}{\text{External Clock (Frequency)}}$

Note: 1X master BC < alternate BC < 2X master BC

Time-out Counter registers. The MIC contains two 16-bit and two 10-bit time-out Counter registers. These registers are filled with zeros during Power-up/Reset/Self-test. The processor must initialize all four registers with non-zero values before the registers are activated. Each register will decrement, when appropriate, one bit for every 240 external clocks.

Bus Time-out Counter (BTC). The MIC will begin to decrement the BTC after all time-out counters have been initialized. The BTC re-initializes to the originally loaded value when a valid sync is detected, the MIC transmits on the serial data bus, a new BTC is loaded, the BTC times-out, or when the Set-up Register bits disable the BTC. If the BTC decrements to zero (times-out), the *BUSY* signal, *ADLD/* signal, and bit 1 of the Status Register are set LOW to indicate that the PIM is the master bus controller. Also, after the BTC times-out, the MIC will begin to repetitively transmit an "I'm Alive" message over the serial data bus until a request is made by the processor to stop or to transmit a bus command. If the No Command or Interrupt Acknowledge time-out counters time-out, the original BTC will be doubled then re-loaded into the BTC. Because the BTC may be doubled by the MIC, the largest value that can be loaded into the BTC register by the processor is 7FFF Hex.

No Command Time-out Counter (NCTC). The MIC will begin to decrement the NCTC after all time-out counters have been initialized. The NCTC re-initializes to the originally loaded value when a valid read or write transfer is performed by the processor or when the counter times-out. If this counter times-out and the time-out counters are not disabled by the Set-up Register bit 10, the *BUSY* signal and bit 1 (Standby) of the Status Register will be set HIGH to indicate that the PIM is now in an alternate bus controller mode. An NCTC time-out will also cause the original BTC value to be doubled.

No Response Time-out Counter (NRTC). The MIC will begin to decrement the NRTC after the last bit of the last serial command has been transmitted unless bit 4 of the Set-up Register is set HIGH. The NRTC re-initializes to the originally loaded value when; the MIC receives a complete serial bus message with valid parity, the Set-up Register is written to, or when the NRTC times-out while the time-out counters are not disabled. If this counter times-out and the time-out counters are not disabled by the Set-up Register bit 10, a no response interrupt to the processor will be generated. The interrupt type will be set in the Status Register on bits 11 and 10.

Interrupt Acknowledge Time-out Counter (IATC). The IATC will begin to decrement after the MIC generates an interrupt request (*INTREQ/* = LOW) to the processor. The IATC counter re-initializes to the originally loaded value when the processor acknowledges the interrupt request (*IACK/* = LOW) or when the IATC times-out while the time-out counters are not disabled. If this counter times out and the time-out counters are not disabled by the Set-up Register bit 10, the *BUSY* signal and bit 1 (Standby) of the Status Register will be set HIGH to indicate that the PIM is now in an alternate bus controller mode. An IATC time-out will also cause the original BTC value to be doubled and the *INTREQ/* to be set HIGH.

Set-up Register. The MIC contains a 17-bit Set-up Register that provides the processor with a means to set-up and initiate communications or diagnostics. The Set-up Register bits are shown in Table 6. Detailed bit descriptions are given on the following page.

TABLE 6. Set-up Register

Register Bit	Bit Description
16	Reset Standby (Set *BUSY* Signal and Status Standby Bit LOW
15	Set in Standby Mode (Reset BTC and Set Standby Bit HIGH, indicates alternate PIM)
14	Select Module Address Register (Use Register for Module Address)
13	Receive Multiple Words with Valid Module Address
12	Disable Receiving “I’m Alive” Messages
11	Receive Buffer Control (Increment on Odd Address Reads)
10	Disable Time-outs (Disable Time-outs Counters)
9	Interrupt on Module Address Only (Interrupt Only on Words with Valid Module Address)
8	Interrupt on Responses Only (Int on Response or Words with Valid Module Valid Address)
7	Receive Multiple Responses (Expect to Receive Multiple Responses)
6	Clear Buffers (Clear Data from Multiple Receive and Transmit Buffers)
5	Monitor Mode (Disable “I’m Alive” Message Transmission)
4	Broadcast Transmit (Do Not Expect a Response)
3	Soft Reset (Reset and Run Self-test)
2	Disable Bus B (Disable Transmission on Bus B)
1	Disable Bus A (Disable Transmission on Bus A)
0	Transmit (Transmit Data Located in Transmit Buffer)

Set-up Register Bits. The Set-up Register bits are initialized to LOW during Power-up/Reset/Self-test. All bits are inactive when LOW and active when set HIGH. Set-up Register Bits 0-4, 6, 7, 15 and 16 are automatically reset LOW after their function is performed. Bits 5, 8, 9-13, and 14 are only reset LOW when written to or when a reset occurs. All 17 bit locations must be set to the appropriate value each time the Set-up Register is written to. All bit locations that are not to be activated should be filled with zeros, including bits 16-31 if written to.

Bit 0 - Transmit: When set HIGH, the data located in Transmit Buffer is transmitted.

Bit 1 - Disable Bus A: When set HIGH, transmission on serial Bus A is disabled.

Bit 2 - Disable Bus B: When set HIGH, transmission on serial Bus B is disabled.

Bit 3 - Soft Reset: When set HIGH, the MIC performs the Reset/Self-test sequence. All register/counter values and set-up bits are re-initialized to LOW.

Bit 4 - Broadcast Transmit: When set HIGH, the MIC will not expect a response after a command is transmitted. The No Response Time-out Counter will be disabled.

Bit 5 - Monitor Mode: When set HIGH, the MIC will not be capable of transmitting "I'm Alive." *BUSY* and bit 1 of the Status Register will be set HIGH.

Bit 6 - Clear Buffers: When set HIGH, the Transmit and Multiple Receive buffers will be cleared of data.

Bit 7 - Receive Multiple Responses: When set HIGH, multiple responses will be placed in the Multiple Receive Buffer. After verifying the correct number of valid responses has been received, a single interrupt will be issued to the processor.

Bit 8 - Interrupt on Responses Only: When set HIGH, the MIC stores and issues interrupts only for responses or messages received with a valid Module Address.

Bit 9 - Interrupt on Valid MODAD Only: When set HIGH, the MIC stores and issues interrupts only for bus words received with a valid Module Address.

Bit 10 - Disable Time-outs: When set HIGH, all time-out counters are disabled and can not be initialized. *BUSY* and bit 1 (Standby) of the Status Register will be set HIGH and the MIC will not be capable of transmitting "I'm Alive."

Bit 11 - Receive Buffer Control: When reading multiple words from the Multiple Receive Buffer using 16-bit word transfers, this bit can be used to select which read address increments the 32-bit buffer.

Bit 11 = LOW (Default)			
	A[4:0]	Buffer	*D[15:0]*
Read #1	10000	Increments	MSW#1
Read #2	10001		LSW#1
Read #3	10000	Increments	MSW#2
Read #4	10001		LSW#2

Note: 10000 could be read successively to extract only the MSWs of the Receive Buffer.

Bit 11 = HIGH			
	A[4:0]	Buffer	*D[15:0]*
Read #1	10001	Increments	LSW#1
Read #2	10000		MSW#1
Read #3	10001	Increments	LSW#2
Read #4	10000		MSW#2

Note: 10001 could be read successively to extract only the LSWs of the Receive Buffer.

Bit 12 - Disable "I'm Alive" Interrupts: When set HIGH, the MIC will not store or issue interrupts when "I'm Alive" messages are received.

Bit 13 - Receive Multiple Mod Address: When set HIGH, all bus words with a valid Module Address will be place in the Multiple Receive Buffer. When an interword gap of >36 clocks is detected, the MIC will issue a single interrupt to the processor.

Bit 14 - Select Module Address: When LOW, the active Module Address of the PIM is determined by the 6-bit hard wired *MODAD[5:0]* pins. When HIGH, the active Module Address of the PIM is determined by the 6-bit value of the Module Address Register.

Bit 15 - Set In Standby Mode: When set HIGH, the BTC will be reset and the *BUSY* signal and bit 1 of the Status Register will be set HIGH to indicate the PIM is an alternate bus controller mode.

Status Register. The MIC has an internal 23-bit Status Register that can be accessed by the processor. The base 16 bits of this register, bits 0 to 15, are also available in each of the remote modes of operation. Bits 16 through 22 are only available in the PIM.

TABLE 7. Status Register

Register Bit	Name	Bit Description
22	Data In Buffer	There is Data in the Multiple Receive Buffer
21	Module Address5	PIM Module Address Bit 5
20	Module Address4	PIM Module Address Bit 4
19	Module Address3	PIM Module Address Bit 3
18	Module Address2	PIM Module Address Bit 2
17	Module Address1	PIM Module Address Bit 1
16	Module Address0	PIM Module Address Bit 0
15	Mode1	Mode Select Bit
14	Mode0	Mode Select Bit
13	Diag1	Diagnostic Result
12	Diag0	Diagnostic Result
11	Vector1	Interrupt Type
10	Vector0	Interrupt Type
9	Bad Par A	Parity Error on Bus A
8	Bad Par B	Parity Error on Bus B
7	Bad Man A	Invalid Manchester Data on Bus A
6	Bad Man B	Invalid Manchester Data on Bus B
5	Word Count Error	Data Word Count Error
4	Code Error	Validation Code Error
3	Block Error	Serial Bus A Not Equal to Bus B
2	Peek Busy	Loads Are Not Yet Stable
1	Standby	Standby - Alternate Bus Controller
0	Timer Started	Time-Out Counters Are Initialized

Status Register Bits. The Status Register bits are initialized during Power-up/Reset. A HIGH indicates the bit is set.

Bit 0 - Timer Started: When HIGH, indicates that all Time-out Counters have been initialized. The Timer Started bit is valid in PIM only and will be LOW in all other modes.

Bit 1 - Standby: When HIGH, indicates PIM is in standby mode (alternate bus controller), Monitor mode, or indicates the Time-outs are disabled by the Set-up Register. When LOW, indicates that the PIM is the bus controller. The Standby bit is valid in PIM only and will be HIGH in all other modes.

Bit 2 - Peek Busy: When HIGH, indicates that the Valid Lag Constant has not time-out since the last output cycle in the Remote Switch Mode. Peek Busy is reset LOW when the Valid Lag Constant times-out. The Peek Busy bit is valid in RSM only and will be LOW in all other modes.

Bit 3 - Block Error: When HIGH, indicates that a serial bus word with valid parity was received on Bus A and Bus B but the 32 bits of data did not compare. Block Error is reset LOW when a valid message, excluding a Peek Module Command, is received with the correct Module Address or when any bus word with an incorrect Module Address is received by a remote module. Block Error is reset to LOW in PIM when any valid command or response, as defined by the Set-up Register, is received. If Set-up Register bit 13 is set HIGH, indicating multiple bus words are expected in the PIM, the Block Error bit will only be reset when the PIM transmits a command. The Block Error bit is valid in all modes.

Bit 4 - Code Error

When HIGH, indicates that a command was received at a remote module with an incorrect validation code. Code Error is reset LOW when a valid message, excluding a Peek Module Command, is received with the correct Module Address or when any message with an incorrect Module Address is received by the remote module. The Code Error bit is valid in remote modes only and will be LOW in PIM.

Bit 5 - Word Error:

When HIGH, indicates that an Execute Command (RSM, DOM) or a Peek Multiple response (PIM) has been received with an incorrect number of words. Word Count Error is reset LOW when a valid message, excluding a Peek Module Command, is received with the correct Module Address or when any message with an incorrect Module Address is received by a remote module. Word Count Error is reset to LOW in PIM when any valid command or response, as defined by the Set-up Register, is received. The Word Count Error bit is valid in all modes except DIM. In DIM the Word Count Error bit will be LOW.

Bit 6 - Bad Man B:

When HIGH, indicates that invalid Manchester data was detected on Bus B. Bad Man B is reset LOW when a valid message, excluding a Peek Module Command, is received with the correct Module Address or when any bus word with an incorrect Module Address is received by a remote module. Bad Man B is reset LOW in PIM when any valid command, as defined by the Set-up Register, is received. If Set-up Register bit 13 is set HIGH, indicating multiple responses are expected in he PIM, the Bad Man B bit will not be reset. This bit is valid in all modes.

Bit 7 - Bad Man A:

When HIGH, indicates that invalid Manchester data was detected on Bus A. Bad Man A is reset as described for Bad Man B. This bit is valid in all modes.

Bit 8 - Bad Par B:

When HIGH, indicates that an invalid parity bit was detected on Bus B. Bad Par B is reset as described for Bad Man B. This bit is valid in all modes.

Bit 9 - Bad Par A

When HIGH, indicates that an invalid parity bit was detected on Bus A. Bad Par A is reset as described for Bad Man B. This bit is valid in all modes.

Bit 10 - Vector0:
Bit 11 - Vector1:

The Vector bits are used to indicate which type of response was last received in the PIM. The Vector bits are valid in PIM only and are defined below.

VECTOR1	*VECTOR0*	**Response Type**
0	0	Normal Response
0	1	Attention Response
1	0	No Response
1	1	Bad Response

Bit 12 - Diag0:
Bit 13 - Diag1:

The Diag bits are used to indicate the diagnostic status of the MIC. The Diag bits are valid in all modes and are defined below.

DIAG1	*DIAG0*	**Diagnostic Status**
0	0	Self-test is Disabled
0	1	Self-test Failed
1	0	Self-test Passed
1	1	Self-test in Progress

Bit 14 - Mode0:
Bit 15 - Mode1:

The Mode bits are used to indicate the base mode of operation of the MIC. Mode bits are valid in all modes and are defined below.

MODE1	*MODE0*	**Mode Selected**
0	0	PIM
0	1	RSM
1	0	DIM
1	1	DOM

Bit 16-21 - Mod Addr:

Indicates the active Module Address of the PIM . Mod Addr bits are valid in PIM only.

Bit 22 - Data In Buffer: When HIGH, indicates that data is located in the Multiple Receive or Transmit Buffer. The Data In Buffer bit is valid in PIM only.

Base Interrupt Vector Number Register. An 8-bit interrupt vector number register is provided for processors with the capability of utilizing vector number interrupt processing. This value will be loaded into the register by the processor during PIM initialization. The least significant 2 bits of this number may be modified before the vector number is placed on the least significant byte of the data bus during an interrupt cycle. The four types of MIC interrupt numbers are shown below. Because the MIC has four interrupt vector numbers the maximum value of the base vector loaded in the register is 11111100 (FC Hex). This interrupt method can be disabled, by setting *AVSEL/* HIGH, for processors that cannot take advantage of vector number interrupt processing or systems where auto vectoring is preferred.

VECTOR1	*VECTOR0*	**Interrupt Type**
0	0	Normal response received
0	1	Attention response received
1	0	No response received
1	1	Bad response received

Module Address Register. A 6-bit module address register is provided to allow the processor to program the module address of the MIC. The MIC uses the value of this register for its Module Address instead of the hard wired *MODAD[5:0]* pins when bit 14 of the Set-up Register is set HIGH. This register shares the same address location within the MIC as the Base Interrupt Vector Number register. If used, both the Module Address and the Interrupt Vector Number must be loaded during the same write-to-MIC cycle.

Transmit Buffer. A 32-bit wide by 33 word deep write only FIFO is provided internal to the MIC to buffer the bus commands that will be sent on the serial data bus. When requested through the Set-up Register, the data located in this buffer will be transmitted on a first data-word-in, first data-word-out order. Each data word will be formatted into 36-bit Manchester messages and transmitted over the serial data bus. Bit 22 of the Status Register can be read to determine if there is data in the Transmit Buffer. The Transmit Buffer may be cleared by setting bit 6 in the Set-up Register.

Receive Buffer. A 32-bit wide by 33 word deep read only FIFO is provided internal to the MIC to store incoming serial bus messages on a first data-word-in, first data-word-out order. A single response from a remote module is stored in a latch while multiple responses are stored in the FIFO. The received messages are available to the processor via register location 10 Hex. FIFO arbitration is transparent to the system. The Receive Buffer may be cleared by setting bit 6 in the Set-up Register.

PIM operational sequences. The following paragraphs describe some of the MIC's basic sequences of operation. The sequences are not intended to provide examples of every condition of operation. The Set-up Register bits can be used in a variety of combinations to disable some features or enhance others.

Time-out counter loading sequence. Time-out Counter counters can be loaded with non-zero values any time after Power-up/Reset/Self-test has been completed. The Timer Started bit (Status Register bit 0) will be set HIGH upon completion of the time-out counter loading sequence. If used, the Base Interrupt Vector Number and Module Address Register should be loaded at this time.

Manchester encoded bus monitor sequence. The MIC continuously monitors the serial data input buses for valid Manchester encoded messages until requested to transmit. The MIC will also accept valid read and write commands from the processor. The No Command Time-out Counter is re-initialized after receiving any valid read or write from the processor.

BTC counter time-out sequence. If no bus activity is detected and the BTC times-out, the following sequence of operations occur:

1) Status Register Bit 1 (Standby), *BUSY*, and *ADLD/* signals are set LOW to indicate that the MIC is in the master bus controller mode.

2) The original BTC value is reloaded into the BTC.

3) The MIC begins to repetitively transmit "I'm Alive" messages which contain a sync pulse, 32 Manchester encoded "1s" and a parity bit indicating odd parity. There is no inter message gap between "I'm Alive" messages.

4) The MIC monitors the *IACK/* pin for a response to the *ADLD/* signal. See the Electrical Characteristics section for detailed timing.

No Command time-out sequence. The MIC performs the following sequence of operations when the NCTC times-out.

1) Status Register Bit 1 (Standby) and the *BUSY* signal are set HIGH to indicate that the MIC is in the alternate bus controller mode.

2) The original BTC value is doubled and reloaded into the BTC. This could allow an alternate bus controller to take control of the bus.

3) The "I'm Alive" message is immediately terminated if active.

4) The MIC will continue the Bus Monitor Sequence.

Receive Cycle Overview. The PIM receives serial Manchester encoded data from remote modules and other bus controllers via the *RXA* and *RXB* pins simultaneously. The MIC checks the data, selects a good message, and stores it in its Receive Buffer (10H). After receiving the appropriate number of messages, the MIC sets the *INTREQ/* signal LOW to indicate a message has been received and waits for the processor to respond with an interrupt acknowledge.

After receipt of an interrupt acknowledge from the processor (processor sets *IACK/* LOW), the MIC places the Auto Vector Level (Base Interrupt Vector Number + interrupt type) on *D[7:0]* and sets the *DSACK/* line LOW. The response interrupt types are Normal response=0, Attention response=1, No response=2, and Bad response=3.

If the processor fails to respond to an interrupt, an Interrupt Acknowledge Time-out will occur (programmable via address O6H). The MIC responds by setting its *INTREQ/* LOW, doubling its BTC, and returning to monitor bus activity. This allows an alternate bus controller, if used, to take over. Any errors in the receive cycle causes the MIC to set the appropriate bits in the Status Register. This completes the bus controller Receive Cycle. The MIC then goes back and waits for a command from the processor.

For systems where autovectoring is preferred, a second interrupt mode is available. In this mode *AVSEL/* is held HIGH and the interrupt level is stored in the MIC Status Register. After the MIC issues an interrupt, the processor may read the location of the interrupt level from the MIC's Status Register (0AH); bits 10 and 11. The interrupt level is not placed on the data bus.

MIC/Processor interrupt timing and MIC/Transceiver interface timing are shown in the Electrical Characteristics section of this document.

Receive sequence. The MIC receives and decodes messages from both buses by performing the following sequence. The MIC is in receive mode until it begins to transmit "I'm Alive" or the processor initiates a transmission by setting bit 0 of the Set-up Register.

The MIC:

1) Monitors both buses for valid sync signal.

2) Decodes next 32 data bits and the parity bit.

3) Calculates parity for the 32 data bits.

4) Compares calculated parity with received parity (parity check).

5) If messages have been received on both buses, compares both 32-bit data words to check if they are identical.

a. If both are identical, selects one to be placed in the Receive Buffer.

b. If the data words are different but the parity check passed:

- Sets Block Error bit HIGH (bit 3 of Status Register).
- If it is not the last expected message of a multiple response it decrements the response counter and returns to the start of receive sequence.
- If it is the last expected message enters interrupt sequence.

c. If the parity check passes on one bus but not on the other:

- Selects the data word with the correct parity and places it in the Receiver Buffer.
- Sets corresponding parity error bit (8 or 9 of Status Register).
- If it is not the last expected message of a multiple response it decrements the response counter and returns to the receive sequence.
- If it is the last expected message enters interrupt sequence.

d. If both data words fail parity check:
- Sets Status Register bits 8 and 9 HIGH.
- Returns to start of receive sequence.

6) If no message has been received from the other bus within 6 clocks (375ns @ 16 MHz):

a. If parity check has passed.
- Places the data into the Receive Buffer.

- If it is not the last expected message of a multiple response it decrements the response counter and returns to the receive sequence.
- If it is the last expected message enters interrupt sequence.

b. If parity check failed:
- Sets corresponding parity error bit (8 or 9 of Status Register).
- Returns to start of receive sequence.

7) If Manchester error has been detected in a received message:

- Sets Manchester error bit (6 or 7 of Status Register).
- If message on other Bus is valid, places the data into Receive Buffer
- If other message received is invalid, sets corresponding Status Register bits.
- If it is not the last expected message of a multiple response it decrements the response counter and returns to the receive sequence.
- If it is the last expected message enters interrupt sequence.

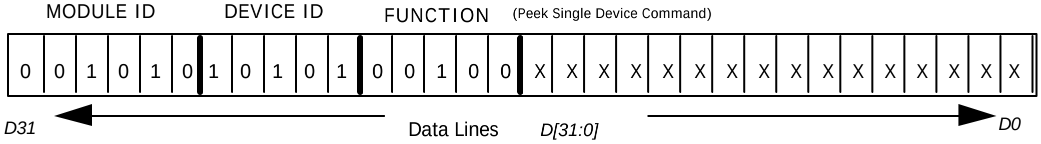
Monitor Mode. The MIC has the ability to enter a monitor mode in which it monitors the serial bus and places all valid messages in its Receive Buffer. This mode can be enabled by setting bit 5 of the Set-up Register. Set-up Register bits 8 and 9 can be used to selectively monitor only messages with valid Module Address and/or messages which are valid responses. Set-up Register bit 12 can be used to disable the MIC from storing "I'm Alive" messages. In the Monitor mode, the MIC disables its own ability to generate an "I'm Alive" signal on the serial bus. The MIC also does not attempt to identify which particular type of message has been received and therefore considers all messages Normal Responses.

The MIC:
1) Receives all messages as described above in the receive sequence.
2) Enters interrupt sequence.
3) The processor may request the data via the Receive Buffer.

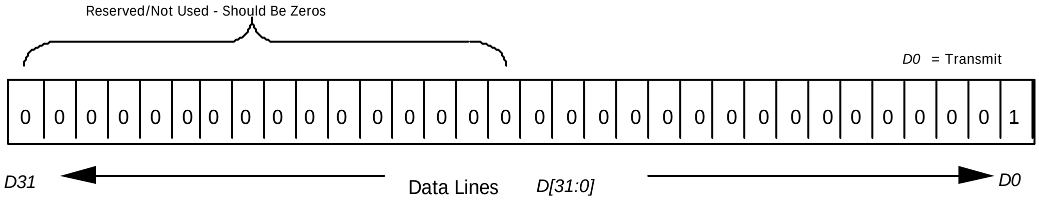
Transmit Cycle Overview. The messages to be transmitted follow the MIC command/response and data formats as specified in Figure 10. To transmit a message, the processor writes a maximum of 33 data words (one command word and up to 32 data words) into the MIC's Transmit Buffer (0EH) and then writes a bit configuration to the Set-up Register (08H). See Table 6 for Set-up Register bit descriptions.

For example, to command a remote module located at module address 0AH and configured in Data Input Mode to respond back with the data acquired from its device address 15H, the following sequence would be performed.

STEP 1) The processor would write the following data to address 0EH (Transmit Buffer) of the MIC.

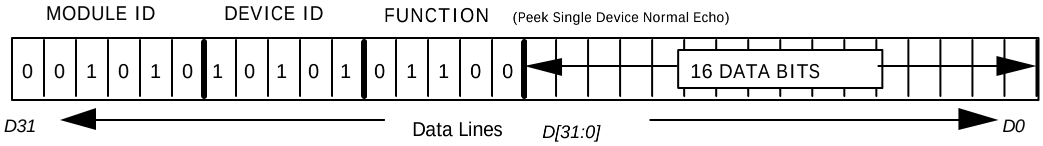

MODULE ID
DEVICE ID
FUNCTION
(Peek Single Device Command)
0 0 1 0 1 0 | 1 0 1 0 1 | 0 0 1 0 0 | X X X X X X X X X X X X X X X X
D31
Data Lines
D[31:0]
D0

STEP 2) The processor would then write to address 08H (Set-up Register) of the MIC. Set-up Register bit 0 (data line *D0*) set HIGH tells the MIC to transmit the data located in the Transmit Buffer.



The MIC would encode and send the message on Bus A and Bus B (approximately 432 clocks to transmit) and then monitor Bus A and Bus B for a response. In normal operation, the processor should not write to the MIC again until it receives an interrupt. The remote module would receive the message then pull the requested data from its internal memory and begin to respond (within 42 clocks) with a normal response on both Bus A and Bus B. The PIM would receive the response (after 432 clocks) and indicate it has a response by bringing *INTREQ/* LOW as described in the receive and interrupt sequences. The response can then be read from the MIC.

STEP 3) The processor could read the response from address 10H (Receive Buffer) of the MIC.



If an execute command is used, the processor would repeat STEP 1 until all words have been loaded; then perform STEP 2. There is no gap between the command word and data words. If a remote module does not receive the proper number of data words or sees a message gap greater than 32 clocks, it will respond with an Attention Response.

The maximum amount of time the bus controller will wait before it considers that no response is going to be returned is programmable via the NRTC. If a MIC in any mode transmits on the serial bus for a time greater than 16,000 clocks, an internal CHATTER TIMER will automatically disable the transmit enable *TXEN* signal. This transmit enable signal is permanently disabled until a reset is issued.

Transmit command sequence. The MIC may perform one of three transmit command sequences after the processor has written data to the Transmit Buffer. The three types of transmit commands are Broadcast, Regular, and Loopback. A transmit command is initiated by a write from the processor to the Set-up register with the appropriate bits set.

The Set-up Register's transmit bit, bit 0, is set for either transmit command. A Transmit command is not valid during Power-up/Reset/Self-test and is dependent on the state of the time-out counters if time-outs are used.

For example, if a transmit command occurs when the time-outs are disabled or not initiated, the MIC will switch from receive mode to transmit mode in nine clock cycles as shown in Figure 17. However, if the MIC is transmitting "I'm Alive" messages, the MIC will not begin the requested transmit command for seventeen clock cycles as shown in Figure 18. The additional eight clocks will allow the remote modules to detect the termination of the current "I'm Alive" message and to prepare to receive the forthcoming message. The remote modules will set the appropriate error bits in their Status Registers (typically Manchester errors) depending on when the "I'm Alive" message was terminated. See the Electrical Characteristics section for further Transmit and Receive timing.

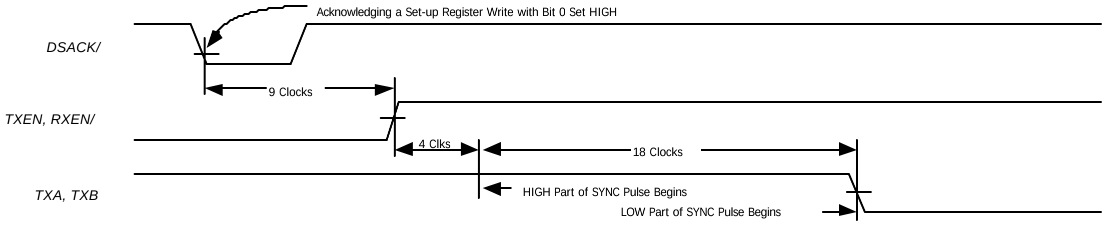

Acknowledging a Set-up Register Write with Bit 0 Set HIGH
DSACK/
9 Clocks
TXEN, RXEN/
4 Clks
18 Clocks
TXA, TXB
HIGH Part of SYNC Pulse Begins
LOW Part of SYNC Pulse Begins

Figure 17. Transmit Timing When "I'm Alive" is not active.

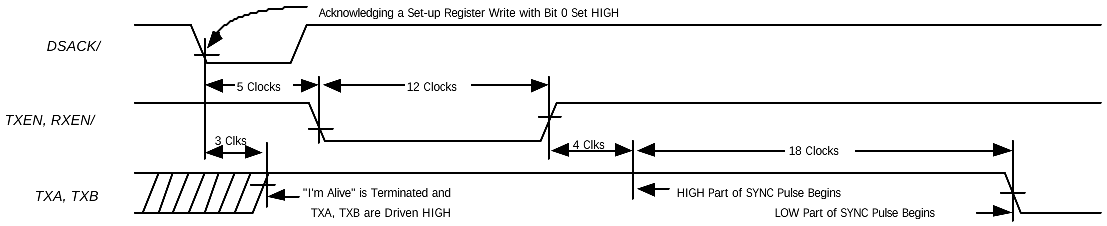

Acknowledging a Set-up Register Write with Bit 0 Set HIGH
DSACK/
5 Clocks
12 Clocks
TXEN, RXEN/
3 Clks
4 Clks
18 Clocks
TXA, TXB
"I'm Alive" is Terminated and
TXA, TXB are Driven HIGH
HIGH Part of SYNC Pulse Begins
LOW Part of SYNC Pulse Begins

Figure 18. Transmit Timing When "I'm Alive" is active.

Broadcast sequence. The MIC will begin transmitting the broadcast message on the serial bus after the processor writes the bit pattern 10xx1 to Set-up Register bits 4 through 0 and *DSACK/* is set LOW. Set-up Register bits 1 and 2 are used to disable either Bus A or Bus B when set. When performing the Broadcast sequence, the MIC:

1) Fetches the message from Transmit Buffer.
2) Adds sync and parity to format the message.
3) Manchester encodes and transmits the message.
4) Starts/continues the bus monitor sequence.

Regular sequence. The MIC will begin transmitting a message on the serial bus after the processor writes the bit pattern 00xx1 to Set-up Register bits 4 through 0 and *DSACK/* is set LOW. Set-up Register bits 1 and 2 are used to disable either Bus A or Bus B when set. When performing the Regular sequence the MIC:

1) Fetches the message from Transmit Buffer.
2) Adds sync and parity to format message.
3) Manchester encodes and transmits the message.
4) Continues sequence 1 through 3 until the buffer is empty. The inter-message gap time shall be zero.
5) Decrements the No Response Time-out Counter if its enabled.
6) Monitors the bus until either a valid message(s) is received or a NRTC time-out occurs.

a. If a valid message(s) is received, returns to receive sequence and enters the interrupt sequence.

Note: Multiple messages may be received depending on the command transmitted and the Set-up Register configuration.

b. If a NRTC time-out occurs, enters the interrupt sequence.

Loopback sequence. During this special transmit cycle the MIC receives its own transmission (*RXA* = *TXA*; *RXB* = *TXB*). The PIM processor may transmit one data word to itself by writing a single word to the Transmit Buffer then setting bits 0 through 2 of the Set-up Register (08H). This data will not be transmitted on the serial bus because both busses are disabled from transmitting. The received message is saved in the Receive Buffer (10H). The MIC issues an interrupt to inform the processor it can read the buffer and check the data integrity. The data loaded in the Transmit Buffer can be combined with Set-up bits to test various functions of the MIC receive sequence. This diagnostic feature checks the processor interface and the MIC's internal circuitry (PCB connections, sync generation/detection, Manchester encoder/decoder, parallel to serial logic, serial to parallel logic). When performing a Loopback the MIC:

1) Fetches the single message from Transmit Buffer.
2) Adds sync and parity to format message.
3) Manchester encodes the message.
4) Transmits the data word to itself. (Without enabling the serial transceivers)
   Both buses perform the following sequences in parallel:
   a. Transmits the data word.
   b. Performs a receive sequence.

Interrupt sequence. The MIC determines the interrupt type after decoding the received message. The four types of interrupts are normal response, attention response, no response, and bad response.

Normal response sequence. The MIC performs the normal response sequence when bits 16 thru 20 (See Table 4 for Function Codes.) of the received message indicate a normal response for the command that was transmitted. A normal response is also assumed if no message was transmitted from the PIM prior to receiving a message (Monitor mode or master/alternate communications). Once the message is validated, the MIC:

1) Sets the interrupt request *INTREQ/* line LOW.
2) Decrements the Interrupt Acknowledge Time-out Counter IATC.
3) Sets Status Register bits 10 and 11 LOW.

4) Monitors the interrupt acknowledge *IACK/* line until it is set LOW by the processor or an IATC time-out occurs.
5) If *IACK/* goes LOW, sets *INTREQ/* HIGH and reads the *AVSEL/* line.

a. If *AVSEL/* is HIGH, resumes the bus monitor sequence.
b. If *AVSEL/* is LOW, places the contents of the base vector interrupt register on the data lines *D[7:0]* and sets *DSACK/* LOW.
   - Monitors *IACK/* until it is set HIGH by the processor.
   - If *IACK/* goes HIGH, tri-states *D[7:0]* and sets *DSACK/* HIGH.
   - Resumes the bus monitor sequence.

6) If IATC time-out occurs, sets *INTREQ/* HIGH and executes the No Command time-out sequence.

Attention Response Sequence. The MIC performs the attention response sequence when bits 16 thru 20 (Function Code) of the received message indicate an attention response for the command that was transmitted. The attention response sequence is the same as the normal response sequence except for the following:

1) Status Register bit 11 is reset LOW and bit 10 is set HIGH.
2) The base vector value (fetched from the base vector interrupt register) is increment by one before being placed on the data lines *D[7:0]*.

No Response Sequence. The MIC performs the no response sequence when a NRTC time-out occurs. The no response sequence is the same as the normal response sequence except for the following:

1) Status Register bit 10 is reset LOW and bit 11 is set HIGH.
2) The base vector value (fetched from the base vector interrupt register) is increment by two before being placed on the data lines *D[7:0]*.

Bad Response Sequence. The MIC performs the bad response sequence when a block compare error occurs or a message is received in the buffer, but the message is not a valid response to the message which was transmitted. The bad response sequence is the same as the normal response sequence except for the following:

1) Status Register bits 10 and 11 are set HIGH.
2) The base vector value (fetched from the base vector interrupt register) is increment by three before being placed on the data lines *D[7:0]*.

Master/Alternate. In some applications it may be desired to have multiple bus controllers to provide redundancy, processor sharing, or a reconfiguration capability. The MIC was designed to be capable of automatically passing the bus mastership between two or more bus controller nodes. A typical application may have a master bus controller and an alternate bus controller.

Master/Alternate Communication. The master bus controller and the alternate bus controller may communicate with each other by transmitting messages with the receiver's

Module Address in the most significant 6 bits. The other 26 bits of the message may contain any information. The transmitting controller may set the broadcast bit (Set-up Register bit 4) to indicate no response is expected. The master/alternate communication sequence is a follows:

Master/Alternate (transmitting device):

1) Fetches message from Transmit Buffer.
2) Adds sync and parity to format the message.
3) Manchester encodes and transmits the message.
4) Repeats 1 - 3 until the Transmit Buffer is empty.
5) Begins the bus monitoring sequence.

Alternate/Master (receiving device):

1) Performs the regular receive sequence.
2) The processor may access the message by reading the Receive Buffer.

Master/Alternate Hand-off. During the initialization routine, the processor loads the Bus Time-out Counter (BTC), the No Command Time-out Counter (NCTC), the No Response Time-out Counter (NRTC), and the Interrupt Acknowledge Time-out Counter (IATC) into the MIC's internal memory. The BTC value is then used to determine which device (master PIM or alternate PIM) will initiate serial bus activity. The master bus controller's BTC is should be less than the alternate bus controller's BTC at power-up to insure that the master PIM always gets control of the bus. When a Bus Time-out occurs, the MIC repeatedly sends an "I'm Alive" message (Sync, FFFFFFFFH, Parity) until a command telling it to stop is received from the processor or the No Command Time-out occurs. The alternate bus controller detects the serial bus activity, resets its Bus Time-out Counter and continues to monitor bus activity. This is done by the MIC's internal hardware. This operation allows the master bus controller to keep control of the serial bus.

Under normal operation, the processor (software schedule) should issue at least one read or write command before a No Command Time-out can occur. When the MIC receives a processor command, the No Command Time-out is reset and the command is executed by the MIC's internal hardware. If for any reason the processor cannot communicate with the MIC, a No Command Time-out will occur and cause the master bus controller to double its BTC. The MIC then monitors bus activity and waits for another Bus Time-out. The alternate bus controller, will take control of the bus since its MIC's BTC is less than the master bus controller's doubled BTC. (The alternate will time-out before the master bus controller and will begin sending "I'm Alive").

Remote Switch Mode (RSM) Operation

In Remote Switch Mode, the 32 data I/O pins *D[31:0]* (Control Bits) are used as discrete output control signals to turn ON/OFF 32 different devices. The address I/O pins *A[4:0]* are used as outputs to address up to 32 2-bit status inputs *STAT[1:0]* which are loaded into the MIC's internal memory. *ADLD/* is used to indicate that the address bus is valid as shown in Figure 19. If *R/W* is asserted HIGH the MIC will hold at the current address and wait for *R/W* to go LOW before loading the *STAT[1:0]* to memory. This signal allows external control of the automatic status gathering feature of the MIC. For example, *R/W* could be used if the status pins for a certain address do not become valid for a period of time longer than the MIC's cycle time. Detailed timing diagrams can be found in the Electrical Characteristics section. Interface examples of the Remote Switch Mode are shown in Applications section of this document. The serial communications capability of the RSM is performed automatically. Communications details are given in the PIM Operation section.

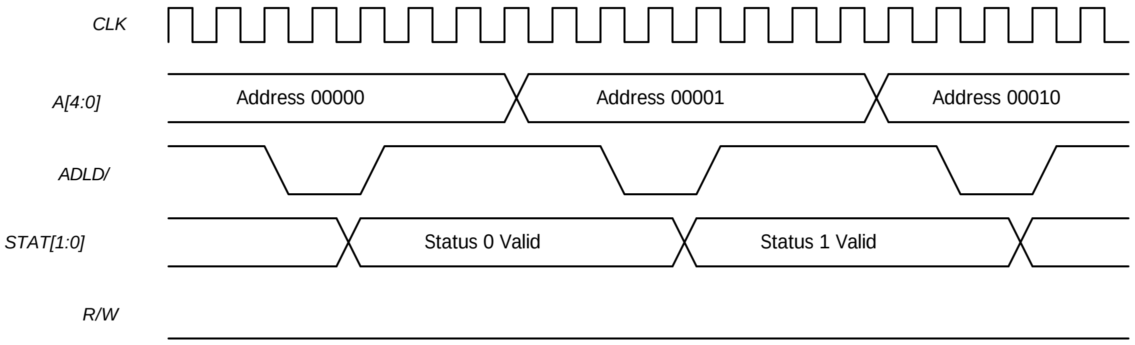


Figure 19. RSM Status Cycling

Single Device Commands: The following illustrates single device commands and responses between the bus controller and an RSM.

Bus Controller: May issue a **Set-up Command**.

MODULE ID	X	X	X	X	X	0	0	0	0	0	0	0	X	X	X	X	VALID LAG CONSTANT

10 - Bit Value

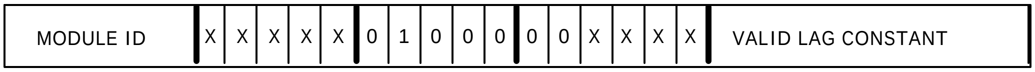
000H = 0 Clocks

001H = 240 Clocks

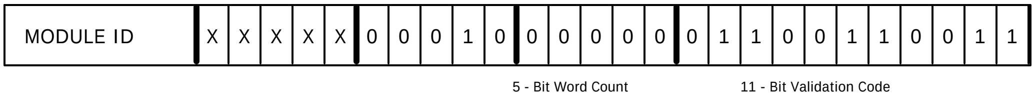
3FFH = 245,520 Clocks

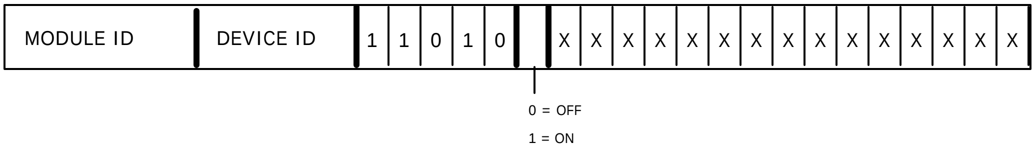
The Set-up Command contains a valid data time lag constant. This time constant can be used to indicate the amount of time that a load connect to *D[31:0]* needs to stabilize (worst case) after being turned ON. If a Peek Single Device command occurs, the return status will reflect a Peek Device busy until this time constant times out. If used, this features prohibits the bus controller from receiving device status information before the device's status is valid. This feature is disabled upon power-up or reset and can be disabled during operation by writing a valid lag constant of 000H to the RSM.

RSM: If no error is detected, an RSM will send back a Set-up Normal Response.

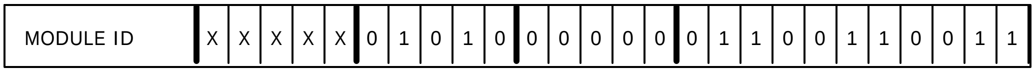


Bus Controller: May issue an **Execute Command** telling the remote module to turn ON or OFF a specific device. The command word will be followed by a 32-bit data word.



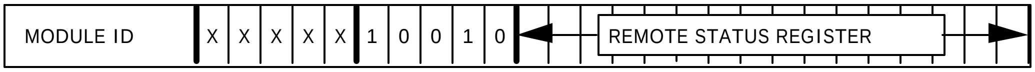


RSM**:** If no error is detected, an RSM will transmit an Execute Normal Response.



After beginning the Execute Normal Response, the MIC begins executing the command. The Data I/O pins *D[31:0]* will be set to the appropriate ON/OFF bit pattern as determined by the DEVICE Address and the ON/OFF bit in the data word. Note that in single device operation, only one Data output line changes, the other 31 Data output lines will stay as previously defined. The RSM will continuously read all status bits. The 2-bit status inputs *STAT[1:0]* are saved in memory and are made available to the bus controller when Peek Single/Multiple Device command is issued.

RSM: If the received validation code does not match with the fixed value 333H or the number of data word received is not equal to the word count, the RSM will send an Execute Attention Response. All Attention Responses contain the remote module's internal status register. See Table 5 for Status Register bit descriptions.



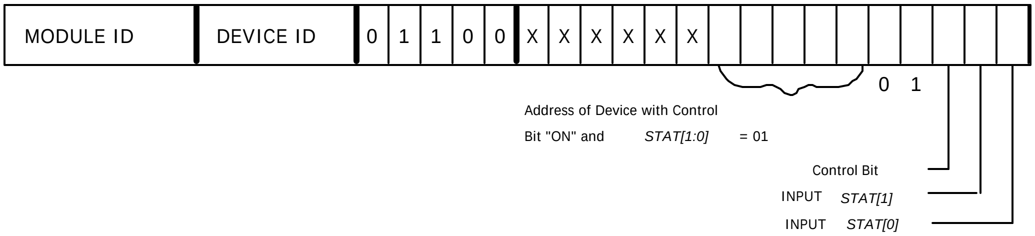
No execution will be performed.

If the command message is not received correctly but the data words are, the entire message will be ignored and no response will be sent back. The block compare check error bit (bit 3) is set HIGH in the Status Register when the block compare check fails. No response will be sent back on a block error.

Bus Controller: May issue a **Peek Single Device Command** to check the status inputs *STAT[1:0]*.

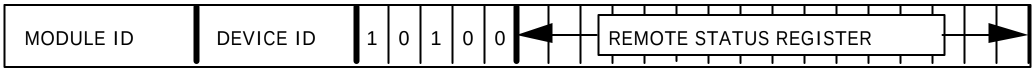
MODULE ID	DEVICE ID	0	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

RSM: Returns a Peek Single Device Normal Response.



The Peek Single Device Normal Response returns the *STAT[1:0]* associated with the requested Device Address. The RSM has built in logic that will allow it to identify any Device Address whose Control Bit is ON and *STAT[1:0]* = 01. When an response's bits 4 and 3 = 01 the bit locations 9 thru 5 contain the address of a device within that module meeting this condition. Bit locations 9-5 will correspond to address bits 4-0. This feature can be useful if, for example, the RSM is interfaced to solid state power controllers and this condition represents a "circuit breaker trip." This priority information is available to the bus controller software in any Peek Device Normal Response from that specific module address.

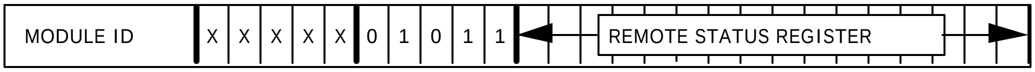
RSM: Returns a Peek Single Device Attention Response if the valid data time lag constant has not timed out.



Bus Controller: May also issue a **Peek Module Command** to check the MIC's internal status.

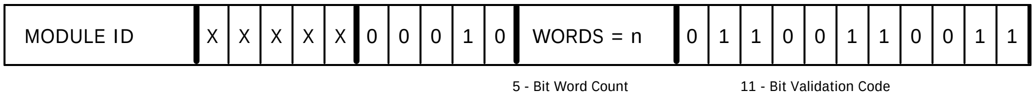
MODULE ID	X	X	X	X	X	0	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

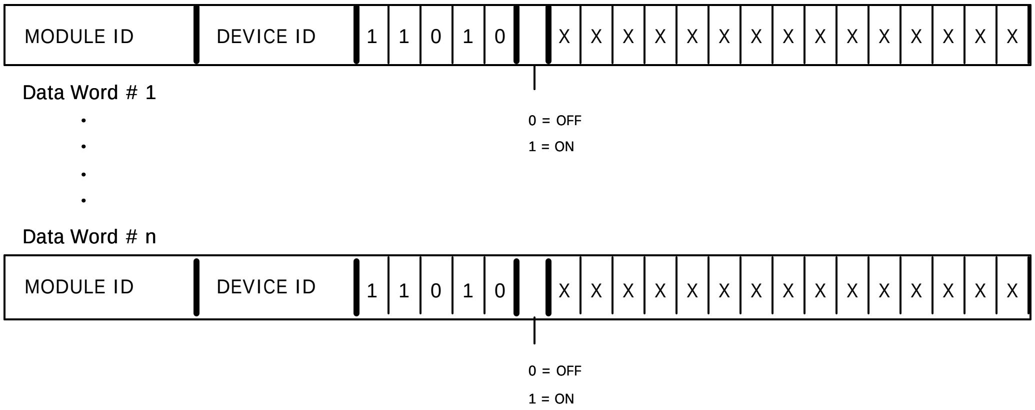
RSM: Returns a Peek Module Normal Response.



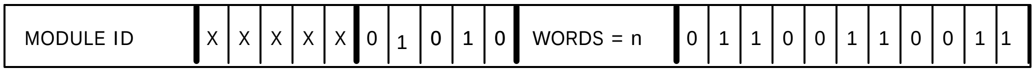
Multiple Device Commands: The bus controller can issue multiple device commands to turn ON/OFF more than one data line (Control Bit) at a time. The following diagrams illustrate multiple device commands and responses between the bus controller and an RSM.

Bus Controller: May issue an **Execute Command**. The command word will be followed by (n) 32-bit Data Words which will tell the remotely located MIC to turn ON or OFF (n) control lines respectively. A word count field of 00000 indicates 1 data word, a field of 00001 indicates 2 data words will be sent.



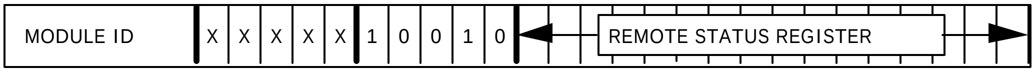


RSM: If no error is detected, an RSM will transmit an Execute Normal Response.



After beginning the Execute Normal Response, the MIC will then execute one data output cycle to set or reset the new data line (Control Bit) values.

RSM: If the received validation code does not match with the fixed value 333H or the number of data word received is not equal to the word count, the RSM will send an Execute Attention Response.



No execution will be performed.

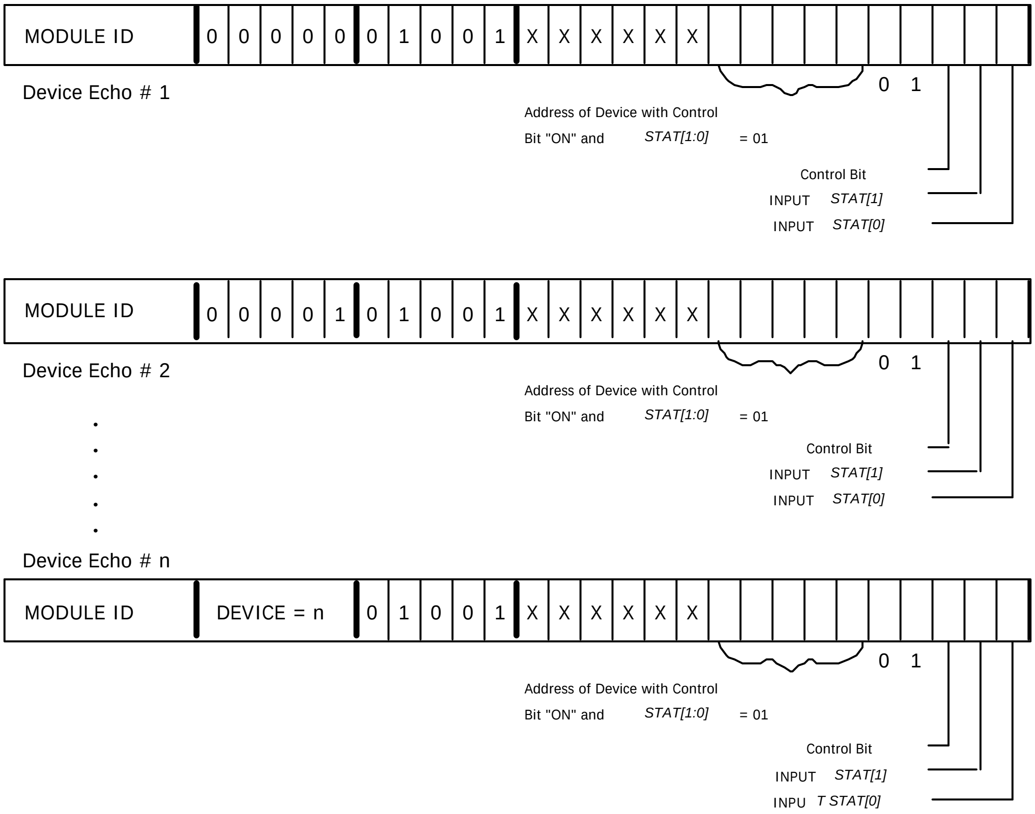
A **Peek Multiple Device Command** can request a block of data from a single remote module. The number of devices (n) to check is sent in the command word. The remote module will return "n" responses, starting at device 1, to the bus controller. In RSM mode, the remote module will send back the status inputs *STAT[1:0]* from device 1 to device (n). The bus controller will place all of the valid responses in the Receive Buffer and then will generate a single interrupt request *INTREQ/*. When the bus controller sets the transmit bit (bit 0 of the Set-up Register), it must also set the Receive Multiple Responses bit (bit 7 of the Set-up Register) to indicate that the bus controller should expect multiple responses.

Bus Controller: Issues a **Peek Multiple Device Command** for information from (n) devices.

Note: The "DEVICES" field is 5 bits; 00000 = 1 device, 00001 = 2 devices, ... and, 11111 = 32 devices.

MODULE ID	X	X	X	X	X	0	0	0	0	1	DEVICES= n	X	X	X	X	X	X	X	X	X	X	X

RSM: Responds with the status input information for (n) devices starting at device 1.



Device Echo # 1

MODULE ID	0	0	0	0	0	0	1	0	0	1	X	X	X	X	X	X									

Device Echo # 2

MODULE ID	0	0	0	0	1	0	1	0	0	1	X	X	X	X	X	X									

.
.
.
.
.

Device Echo # n

MODULE ID	DEVICE = n	0	1	0	0	1	X	X	X	X	X	X										

A **Module Broadcast Command** may be used to either turn ON or OFF all devices on a specific remote module operating in RSM or RSM/DIM Combination Mode. The following illustration shows a module broadcast command. No response is expected from the remote module with this type of command.

Bus Controller: Sends a **Module Broadcast "ON" Command**.

MODULE ID	X	X	X	X	X	1	1	0	1	1	X	X	X	X	X	0	1	1	0	0	1	1	0	0	1	1

11 - Bit Validation Code

Note: no response will be issued by the remote module.

In broadcast operation, the MIC outputs either 00000000H to turn OFF all 32 data lines or FFFFFFFFH to turn ON all 32 data lines. Only one data output cycle is performed.

A **Self-Test Command** can be issued by the bus controller to instruct any remote module to a run self-test. The following example illustrates the Self-test Command and its responses between the bus controller and a remote module.

Bus Controller: Sends a **Self-test Command** to a remote module.

MODULE ID	X	X	X	X	X	0	0	1	0	1	X	X	X	X	X	0	1	1	0	0	1	1	0	0	1	1

11 - Bit Validation Code

RSM: Sends a Self-test Normal Response to acknowledge the reception of the Self-test Command.

MODULE ID	X	X	X	X	X	0	1	1	0	1	X	X	X	X	X	0	1	1	0	0	1	1	0	0	1	1

The remote module will start its self-test if no error has occurred. The data line outputs (Control Bits) remain unchanged. If the received validation code does not match with the proper fixed value, the remote module will send a Self-test Attention Response and ignore the command.

MODULE ID	X	X	X	X	X	1	0	1	0	1	◄ REMOTE STATUS REGISTER ►															

Bus Controller: May issue a **Peek Module Command** after the self-test command in order to obtain the results of the self-test and status of the remotely located MIC.

MODULE ID	X	X	X	X	X	0	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Remote Module: Will return a Peek Module Normal Response with the remotely located MIC's internal status.

MODULE ID	X	X	X	X	X	0	1	0	1	1	◄ REMOTE STATUS REGISTER ►															

Data Input Mode (DIM) Operation

In Data Input Mode, the MIC interfaces the MIC Bus to any device providing a digital format. The 5-bit address bus is used to repetitively address up to 32 external devices of 16-bit digital data words for storage in the MIC's internal memory. This data can be transmitted to the bus controller from the DIM in the appropriate responses. This mode may be used to allow the MIC to acquire data from remote analog or digital sensors or from remotely located memory. Figure 20 shows DIM address and data cycling wave forms. Detailed timing diagrams can be found in the Electrical Characteristics section. Interface examples of the Data Input Mode are shown in Applications section of this document. The serial communications capability of the DIM is performed automatically. Communications details are given in the PIM Operation section.

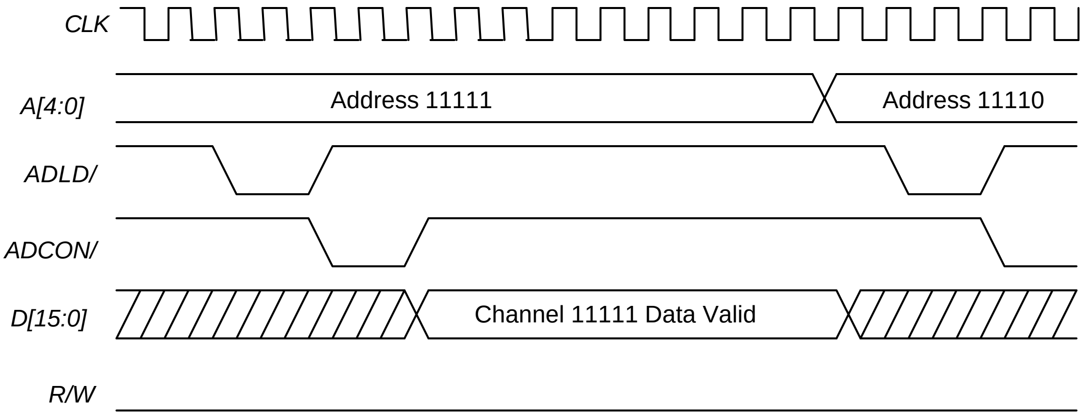


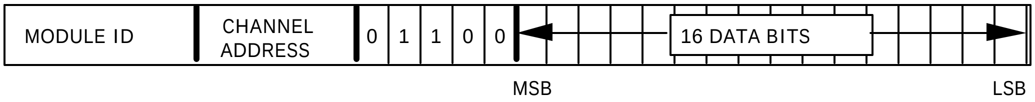
Figure 20. DIM Data Cycling

Address I/O pins *A[4:0]* are used as outputs to address up to 32 devices. The 16 data I/O pins *D[15:0]* are used as inputs to read the 16-bit digital data per device. The *ADLD/* signal is an output used to indicate the address signals are valid. The *ADCON/* signal is used to initiate a conversion or request data from an external device. If *R/W* is asserted HIGH the MIC will hold at the current address and wait for *R/W* to go LOW before loading the *D[15:0]* to memory. This signal allows external control of the automatic data gathering feature of the MIC. For example, *R/W* could be used if the data for a certain address does not become valid for a period of time longer than the MIC's cycle time. When using an analog-to-digital converter, *R/W* is set HIGH during conversion and is asserted LOW to indicate when the conversion is complete. The digital data is then read from the data bus and placed in a specific memory location (location reflects address of specified device). The address decrements and the cycle repeats. The MIC begins with the highest address (1FH), decrements to the lowest (00H) and then begins at the highest address again. The bus controller may access the data via a Peek Device commands.

Bus Controller: May issue a **Peek Single Device Command**.

MODULE ID	CHANNEL ADDRESS	0	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

DIM: Returns a Peek Single Device Normal Response.



The bus controller can also issue a **Set-up Command** to specify the number of device(s) to monitor. The 5 least significant bits can be used to short cycle the number of devices sampled. The bus controller can insert a number less than 32 to utilize this feature. The MIC will default to 32 if no Set-up Command is issued.

Bus Controller: May issue a **Set-up Command**.

Note: The "CHANNEL" field is 5 bits; 00000 = 1 device, 00001 = 2 devices, ...and, 11111 = 32 devices.

MODULE ID	X	X	X	X	X	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	CHANNEL= n

DIM: If no error is detected, the DIM will send back a Set-up Normal Response.

MODULE ID	X	X	X	X	X	0	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	CHANNEL= n

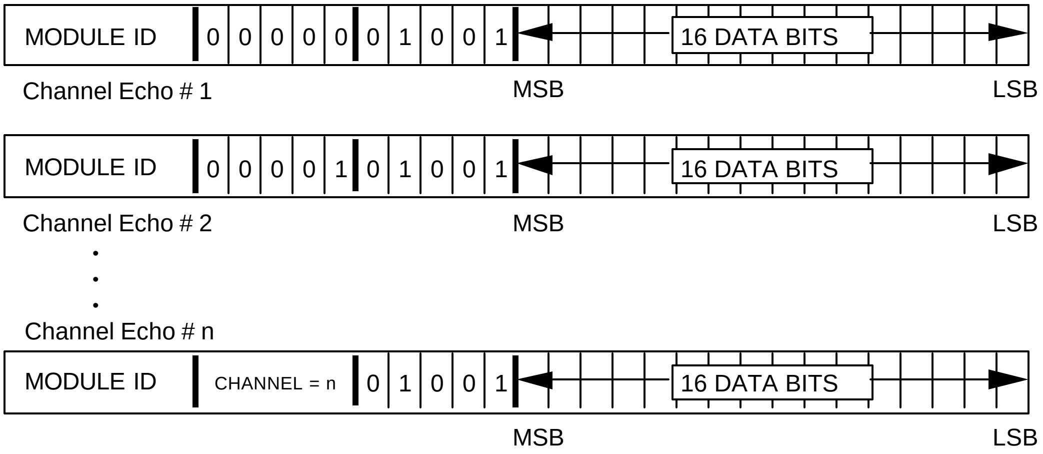
The bus controller may issue a **Peek Multiple Device Command** to request a block of data from a single remote module. The number of devices (n) to check is sent in the command word. The remote module will return (n) responses, starting at device 1, to the bus controller. Each successive message will be transmitted without any message gap. In DIM mode, the remote module will send back the data line inputs *D[15:0]* from device 1 to device (n). The bus controller will place all of the valid responses in the Receive Buffer and then will generate a single interrupt request *INTREQ/*. When the bus controller sets the transmit bit (bit 0 of the Set-up Register), it must also set the Receive Multiple Responses bit (bit 7 of the Set-up Register) to indicate that the bus controller should expect multiple responses.

Bus Controller: May issues a **Peek Multiple Device Command** for information from (n) devices.

Note: The "DEVICES" field is 5 bits; 00000 = 1 devices, 00001 = 2 devices, ...and, 11111 = 32 devices.

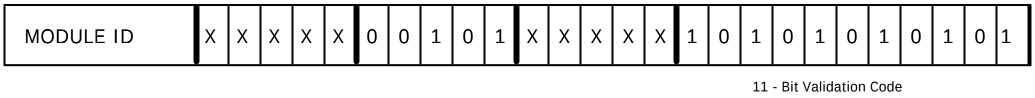
MODULE ID	X	X	X	X	X	0	0	0	0	1	CHANNELS= n	X	X	X	X	X	X	X	X	X	X	X

DIM: Responds with Peek Device Normal Responses for (n) devices starting at device 1.

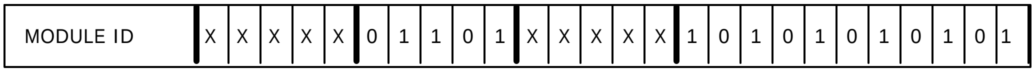


A **Self-Test Command** can be issued by the bus controller to instruct any remote module to a run self-test. The following example illustrates the self-test command and its responses between the bus controller and a remote module.

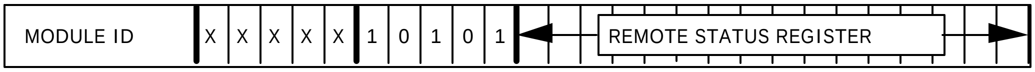
Bus Controller: May issue a **Self-test Command** to a remote module.



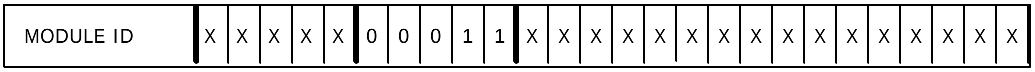
DIM: Sends a Self-test Normal Response.



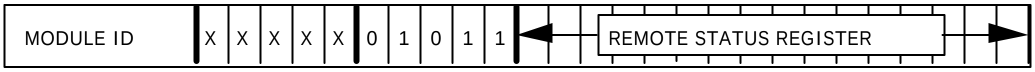
The remote module will start its self-test if no error has occurred. If the received validation code does not match with the proper fixed value, the remote module will send a Self-test Attention Response and ignore the command.



Bus Controller: May issue a **Peek Module Command** after the Self-test Command in order to obtain the results of the self-test and status of the remotely located MIC.



DIM: Will return a Peek Module Normal Response with the remotely located MIC's internal status.



Data Output Mode (DOM)/DIM Operation

The Data Output Mode is automatically combined with Data Input Mode for remotely located MIC's hard wired for the Data Output base operational mode. In this DOM/DIM combination mode, the DIM operation interfaces the MIC Bus to any device driving a digital output (A/D converters for example). Address I/O pins *A[4:0]* are used as outputs to address up to 32 devices. The 16 data I/O pins *D[15:0]* are used as inputs to read the 16-bit digital data per device. The *ADLD/* signal is an output used to indicate the address signals are valid. The *ADCON/* signal is used to initiate conversion. *R/W* is used as an input signal to suspend the DIM cycle until the data on *D[15:0]* is valid. The DIM operation functions as described in the Data Input Mode Operation section of this document.

The Data Output Mode has priority over the Data Input Mode. When a DOM command is received by the MIC, it stops the DIM operation, asserts *DIR* HIGH and begins to perform the requested DOM operation. See timing in Figure 21. The output cycle begins after the *ADCON/* signal is asserted in the current DIM cycle. In essence, DOM performs a cycle sharing operation because is uses the converting time between an *ADCON/* (DIM convert signal) and *R/W* (DIM - ADC conversion signal) to output the DOM information.

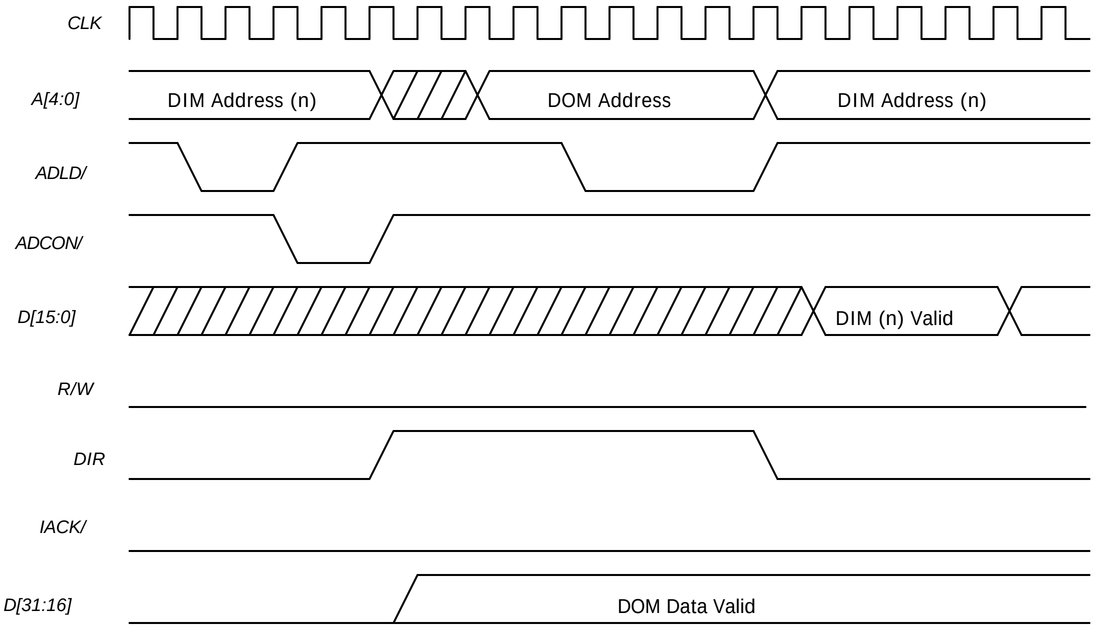
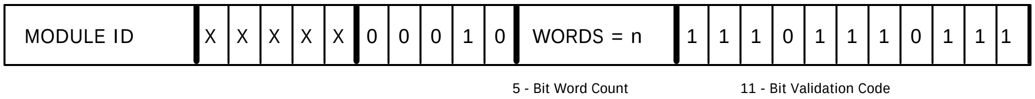


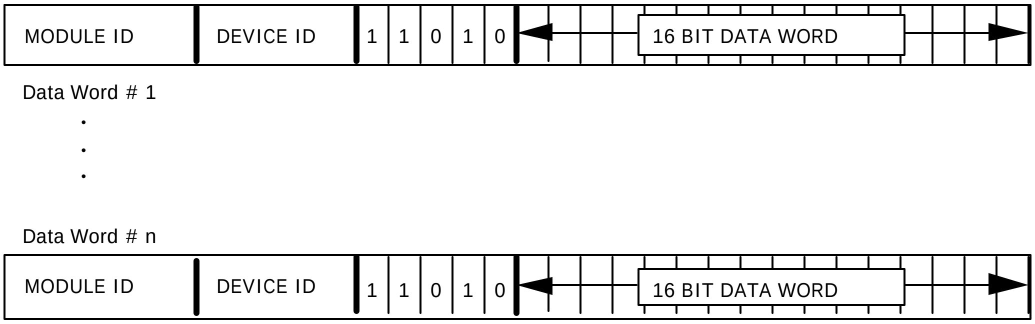
Figure 21. DOM/DIM Operation

The DOM base operational mode interfaces the MIC Bus to devices that accept digital information such as; D/A Converters, microprocessors, memory devices and various discrete components. The 16 data I/O pins *D[31:15]* are used as outputs. Address I/O pins *A[4:0]* are used as outputs to address up to 32 locations. In addition, *ADLD/* is used to indicate the address signals are valid. The *IACK/* signal is used as an input that indicates the external device (location) is ready to accept the next data word. This signal may the held LOW to disable the handshaking feature. If *IACK/* is held LOW, multiple data words are placed consecutively on the data bus in the order they were received. Detailed timing diagrams can be found in the Electrical Characteristics section. Interface examples of the Data Output Mode are shown in Applications section of this document. The serial communications capability of the DOM is performed automatically. Communications details are given in the PIM Operation section.

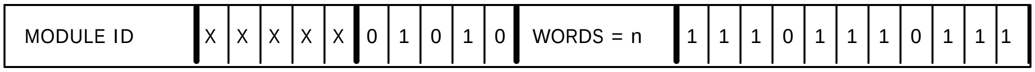
The following diagrams will illustrate Data Output Mode operation between the bus controller and a remotely located MIC in Data Output Mode. All DIM commands are valid in the DOM and can be used as previously described.

Bus Controller: May issue an **Execute Command**. The command word will be followed by (n) 32-bit Data Words.

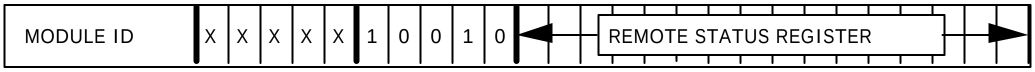




DOM: If no error is detected, the DOM will transmit an Execute Normal Response and then proceed to writing data to the device(s) specified by the bus controller.



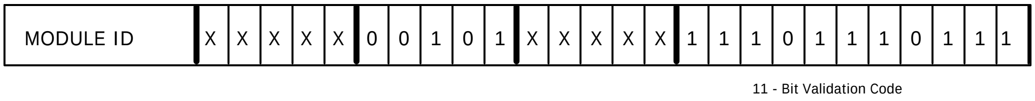
DOM: If the received validation code does not match with the fixed value 777H, or the number of Data Words received is not equal to the word count, the DOM will send an Execute Attention Response.



No execution will be performed.

A **Self-Test Command** can be issued by the bus controller to instruct any remote module to a run self-test.

Bus Controller: Sends a **Self-test Command** to a remote module.



DOM: Sends a Self-test Normal Response.

MODULE ID	X	X	X	X	X	0	1	1	0	1	X	X	X	X	X	1	1	1	0	1	1	1	0	1	1	1

The remote module will start its self-test if no error has occurred. If the received validation code does not match with the proper fixed value, the remote module will send a Self-test Attention Response and ignore the command.

MODULE ID	X	X	X	X	X	1	0	1	0	1	← REMOTE STATUS REGISTER →

Bus Controller: May issue a **Peek Module Command** after the Self-test Command in order to obtain the results of the self-test and status of the remotely located MIC.

MODULE ID	X	X	X	X	X	0	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

DOM: Will return a Peek Module Normal Response with the remotely located MIC's internal status.

MODULE ID	X	X	X	X	X	0	1	0	1	1	← REMOTE STATUS REGISTER →

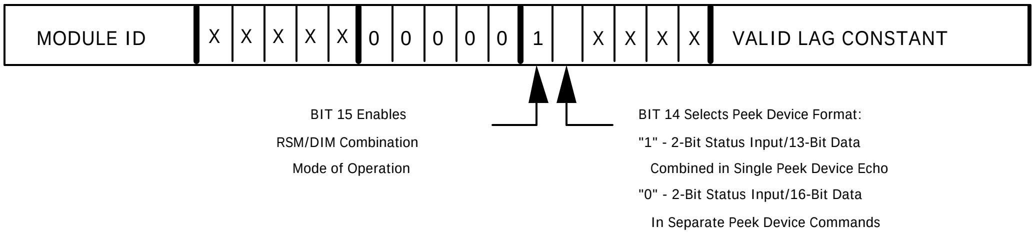
RSM/DIM Combination Mode Operation

A remote module configured in RSM/DIM mode will be capable of performing both the remote switching output and data input functions as previously described. The RSM/DIM Combination Mode is enabled by setting bit 15 in a Set-up Command and sending the command word to a remote module that is hard wired for RSM mode. This mode is useful for modules which must control data as well as acquire data. One example would be if a remote module was required to transmit memory data back to the bus controller. The RSM control line outputs with *A[4:0]* could be used to directly address a very large amount of memory in blocks of 32 16-bit words.

The remote module in RSM/DIM will cycle through all 32 2-Bit *STAT[1:0]* locations then perform a DIM *D[15:0]* cycle to the highest device address. The MIC will then cycle through the RSM status again then decrement the DIM address and perform one DIM cycle. This cycling between the RSM status and DIM data is repeated continually. The *DIR* signal is used to indicate to external logic which cycle is being performed.

The valid lag constant (for RSM mode) is sent in the first Set-up Command word to an RSM when setting RSM/DIM combination mode (Bit 15 HIGH). All following Set-up Commands will contain the number of devices to monitor for the DIM operation.

Bus Controller: Issues a **Set-up Command** with the valid data time lag constant.



MODULE ID	X	X	X	X	X	0	0	0	0	0	1		X	X	X	X	VALID LAG CONSTANT

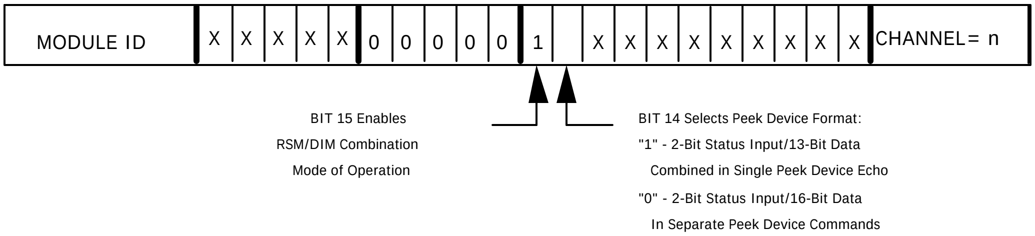
BIT 15 Enables
RSM/DIM Combination
Mode of Operation

BIT 14 Selects Peek Device Format:
"1" - 2-Bit Status Input/13-Bit Data
Combined in Single Peek Device Echo
"0" - 2-Bit Status Input/16-Bit Data
In Separate Peek Device Commands

RSM/DIM: Will respond with a Set-up Normal Response.

MODULE ID	X	X	X	X	X	0	1	0	0	0	1		X	X	X	X	VALID LAG CONSTANT

Bus Controller: The number of devices to monitor for DIM mode may be configured by sending a second **Set-up Command** (with bit 15 set).



MODULE ID	X	X	X	X	X	0	0	0	0	0	1		X	X	X	X	X	X	X	X	X	CHANNEL= n

BIT 15 Enables
RSM/DIM Combination
Mode of Operation

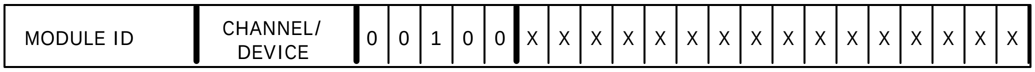
BIT 14 Selects Peek Device Format:
"1" - 2-Bit Status Input/13-Bit Data
Combined in Single Peek Device Echo
"0" - 2-Bit Status Input/16-Bit Data
In Separate Peek Device Commands

RSM/DIM: Will respond with a Set-up Normal Response.

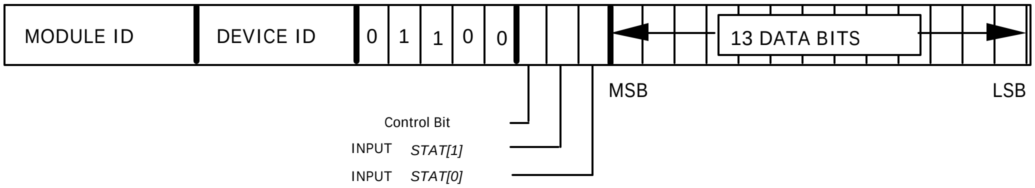
MODULE ID	X	X	X	X	X	0	1	0	0	0	1		X	X	X	X	X	X	X	X	X	CHANNEL= n

In both cases, bit 14 of the Set-up Command indicates the format of a Peek Device Normal Response. Bit 15 must be set in all subsequent Set-up Commands in order to remain in RSM/DIM Combination Mode.

Bus Controller: May issue a **Peek Single Device Command**.

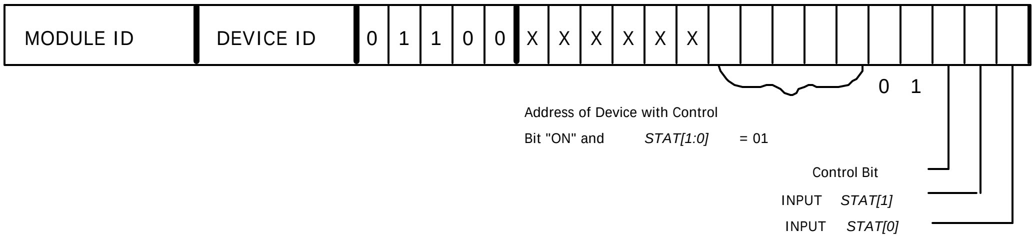


RSM/DIM: If the last Set-up Command had BIT 14 set, then the RSM/DIM will respond with a Peek Single Device Normal Response with RSM data (2-BIT status input and a reserved bit) and DIM data (Least Significant 13 bits-*D[12:0]*).

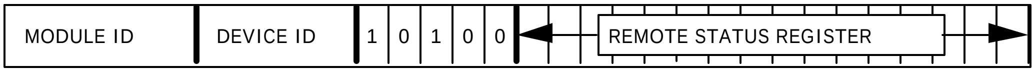


OR

RSM/DIM: If the last Set-up Command had bit 14 reset, then RSM/DIM will respond with a Peek Single Device Normal Response with RSM data only.



RSM/DIM: If the valid data time lag constant has not timed out, the RSM/DIM returns a peek single device attention response.



The RSM/DIM combination mode requires a special command to acquire the data from the DIM operation of the combined mode if the combined data format is not being used (bit 14 of the Set-up Command is reset). This command is used to send back the 16-bit input data words located in the MIC's internal memory. The DIM operation in combined mode works the same as in non-combined mode except a Peek Single RSM/DIM-Data Device Command and Peek Multiple RSM/DIM-Data Device Command are used instead of Peek Single Device Command and Peek Multiple Device Command for regular DIM non-combined operation. The functions of the commands are the same; however, the function code field in the command word is different. All of the RSM commands remain the same during combination mode when bit 14 of the Set-up Command is reset.

Bus Controller: May issue a **Peek Single RSM/DIM-Data Device Command** if the requested data is more than 13 bits.

MODULE ID	CHANNEL ADDRESS	0	0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

RSM/DIM: Will respond with a Peek Single RSM/DIM-Data Device Normal Response which contains all 16-bits of data for the requested device.

MODULE ID	CHANNEL ADDRESS	0	1	1	1	1	◄ 16 DATA BITS ►
							MSB … LSB

Bus Controller: May issue a **Peek Multiple RSM/DIM-Data Device Command** to receive data from (n) devices.

MODULE ID	X	X	X	X	X	0	0	1	1	0	CHANNELS = n	X	X	X	X	X	X	X	X	X	X	X

RSM/DIM: Will respond with multiple responses starting at device 1 and ending with device (n). Each message will contain up to 16 bits of device data. This command operates the same as the Peek Multiple Device Command which is described in the DIM operation section.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +7.0	V
Input Voltage	V_{in}	-0.3 to +7.0	V
Operating Temperature Range	T_A	-55 to +125	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

Note: *Stresses listed under "Absolute Maximum Ratings" may be applied to the MIC without causing permanent damage. However, exposure to absolute maximum ratings for extended periods may affect the MIC's reliability. All unused inputs must be connected to either V_{DD} or Ground. All unused outputs should be left open.*

Recommended Operating Conditions

Parameter	Symbol	Min	Max	Unit
Supply Voltage	V_{DD}	4.5	5.5	V
Input High Voltage	V_{IH}	2.0	--	V
Input Low Voltage	V_{IL}	--	0.8	V
Output High Current (source)	I_{OH}	--	-1.0	mA
Output Low Current (sink)	I_{OL}	--	6.0	mA
Operating Temperature Range	T_A	-55	+125	°C

Clock Pin Parameter	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	3.9	--	V
Input Low Voltage	V_{IL}	--	0.6	V
Input Capacitance	C_{IN}	--	15.0	pf

Note: *Decoupling capacitance is required to adequately control the inductive effects caused by high speed and high current output switching. It is recommended that at least two 0.2 microfarad capacitors be connected from V_{DD} to Ground as close to the MIC as possible.*

DC and Operating Characteristics

(At Recommended Operating Conditions)

Parameter	Condition	Symbol	Min	Max	Unit
Output High Voltage	V_{DD}=Min, I_{OH}=Max	V_{OH}	2.4	--	V
Output Low Voltage	V_{DD}=Min, I_{OL}=Max	V_{OL}	--	0.4	V
Input Leakage Current	GND V_{IN} V_{DD}	I_{IL}	-10.0	10.0	µA
Output Leakage Current (high Z)	GND V_{OUT} V_{DD}	I_{OZ}	-10.0	10.0	µA
Supply Current*	V_{DD}=Max	I_{DD}	--	150.0	mA
Power Dissipation*	V_{DD}=Max	P_D	--	1.0	W
Input Capacitance		C_{IN}	--	6.0	pf
Output Capacitance		C_{OUT}	--	7.0	pf
Bidirectional Capacitance		$C_{I/O}$	--	7.0	pf

* *Supply current and power dissipation include Quiescent and Dynamic current with a maximum external clock frequency and 50 pf loads.*

Thermal Considerations

Characteristic	Symbol	Value	Rating
Thermal Resistance - Ceramic PGA Junction to Ambient	$ØJ_A$	40	^{o}C/W

A critical component of a semiconductor's reliability is the chip-junction temperature. The chip-junction temperature, T_J , can be computed from

$$T_J = T_A + (P_D \, Ø_{JA})$$

where

T_A = ambient temperature in °C

P_D = power dissipation in Watts

$Ø_{JA}$ = thermal resistance, junction to ambient, in °C/W.

Microelectronic packages are characterized by their resistance to thermal transfer. The total thermal resistance ($Ø_{JA}$) of a package can be divided into two components. The internal thermal resistance ($Ø_{JC}$) between the semiconductor junction and the package (case) external surface, and the external thermal resistance ($Ø_{CA}$) from the package (case) to the outside ambient. The components are related by the equation:

$$Ø_{JA} = Ø_{JC} + Ø_{CA}$$

$Ø_{JC}$ is a fixed value due to the package thermal conductivity and the die attachment material and processing. However, $ØJ_{CA}$ and T_A are dependent on the application and the thermal management technique used. It must be kept in mind that local printed-circuit board temperatures may be elevated due to the heating effect of other components and poor natural-convection-cooling conditions. Forced convection or thermal conduction techniques can be employed where appropriate.

AC Characteristics

Most current applications, including the U.S. Army's M1A2 battle tank, are clocking the MIC at 16 MHz to obtain a 1.33 Megabit per second data bus rate. The current implementation of the MIC has a 24 MHz maximum external frequency.

Serial Data Rate Calculation

The MIC was designed so that the serial data bus rate would equal the external clock frequency divided by twelve. For example, if a 1.33 Megabit per second serial data rate is desired, the external clock frequency can be calculated as follows:

$$\textbf{Data Rate} = \frac{\textbf{External Clock (MHz)}}{\textbf{12 (Clocks/bit)}} \quad = \quad \textbf{1.33 Megabit/Sec} = \frac{\textbf{16 MHz}}{\textbf{12}}$$

Because the external clock can be any value, depending on the application, some of the following AC parameters are related to number of clocks instead of specific time.

AC Characteristics - Clock Input and Reset Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
	Frequency of Operation	*	24.0	MHz
	Duty Cycle at 1.5V Level	45	55	%
1	Cycle Time (Clock)	41	*	ns
2, 3	Clock Pulse Width	19	*	ns
4, 5	Rise and Fall Times	—	10	ns
6	Reset Delay Time (Power-Up)	24	—	Clocks
7	Reset Pulse Width	4	—	Clocks

* *The MIC is a fully static component and* ***does not*** *require a minimum clock frequency to refresh internal logic.*

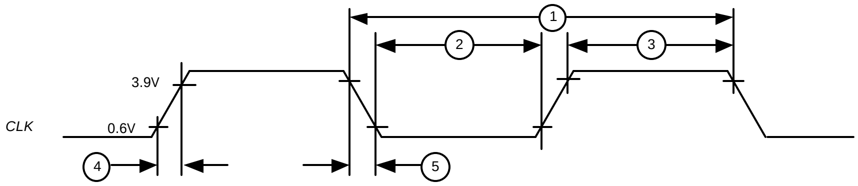


Figure 22. Clock Input Timing

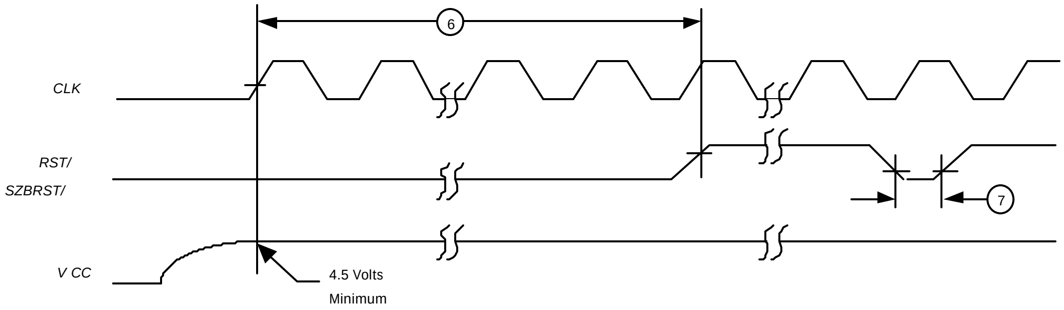


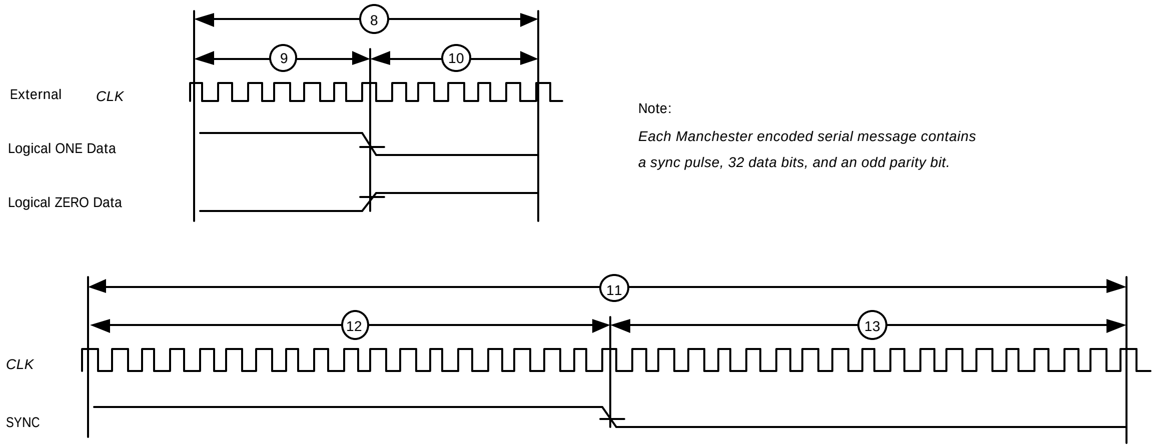
Figure 23. Reset Timing

AC Characteristics - Serial Transmit/Receive Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
8 TX	Bit-Time, TX-Transmit (clocks per data bit)	12	12	Clocks
8 RX	Bit-Time, RX-Receive	10.5	13.5	Clocks
9, 10 TX	Data Bit HIGH/LOW Time, Transmit	6	6	Clocks
9, 10 RX	Data Bit HIGH/LOW Time, Receive	4.5	7.5	Clocks
11 TX	Sync Pulse Width, Transmit	36	36	Clocks
11 RX	Sync Pulse Width, Receive	34.5	37.5	Clocks
12, 13 TX	Sync Pulse HIGH/LOW Time, Transmit	18	18	Clocks
12, 13 RX	Sync Pulse HIGH/LOW Time, Receive	16.5	19.5	Clocks
14	Transmit Enable Active to Sync Pulse Start	4	--	Clocks
15	Parity Bit Complete to Transmit Enable Inactive	0	--	ns
16	Receive Enable Active to Sync Pulse Start	0	--	ns
17	Parity Bit Complete to Receive Enable Inactive	40*	--	Clocks

**Value shown for Number 17 is valid for remote modes only. All modes hold RXEN/ active except when running self-test or transmitting. Processor interface mode is a bus controller and can transmit at any time.*



Note:

Each Manchester encoded serial message contains a sync pulse, 32 data bits, and an odd parity bit.

Figure 24. Serial Data Characters

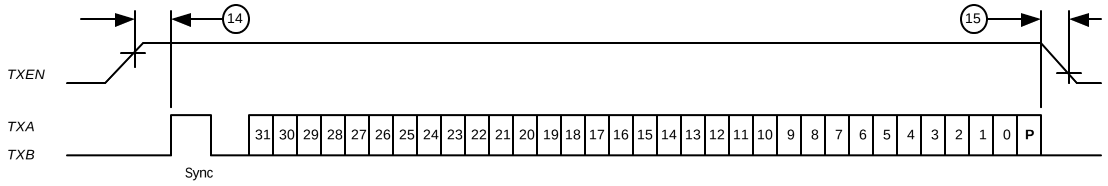

14
15
TXEN
TXA
TXB
Sync
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 P

Figure 25. Transmit Timing

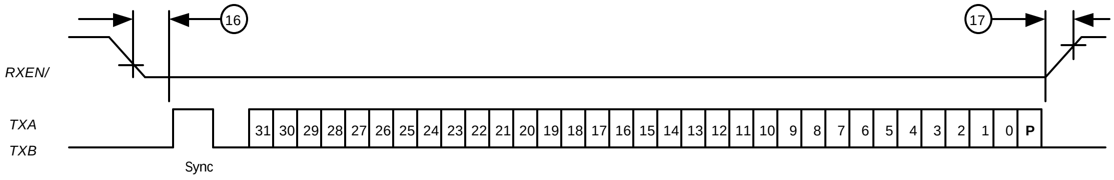

16
17
RXEN/
TXA
TXB
Sync
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 P

Figure 26. Receive Timing

AC Characteristics - Processor Write/Read Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
18	*R/W, WS[1:0], A[4:0]* Valid to *CE/* Falling (Set-up time)	5	--	ns
19	*CE/* LOW to *R/W, WS[1:0], A[4:0]* Invalid (Hold time)	20	--	ns
20	*D[31:0]* Valid to *CE/* Falling (Set-up time)	0	--	ns
21	*DSACK/* LOW to *D[31:0]* Invalid (Hold time)	20	--	ns
22	*DSACK/* LOW to *CE/* HIGH	0	--	ns
23	*CE/* HIGH to *DSACK/* HIGH, *D[31:0]* High Impedance	--	0.5	Clocks
24	*CE/* LOW to *D[31:0]* Valid	--	4	Clocks
25	*D[31:0]*Valid to *DSACK/* LOW (Set-up time)	0	--	ns

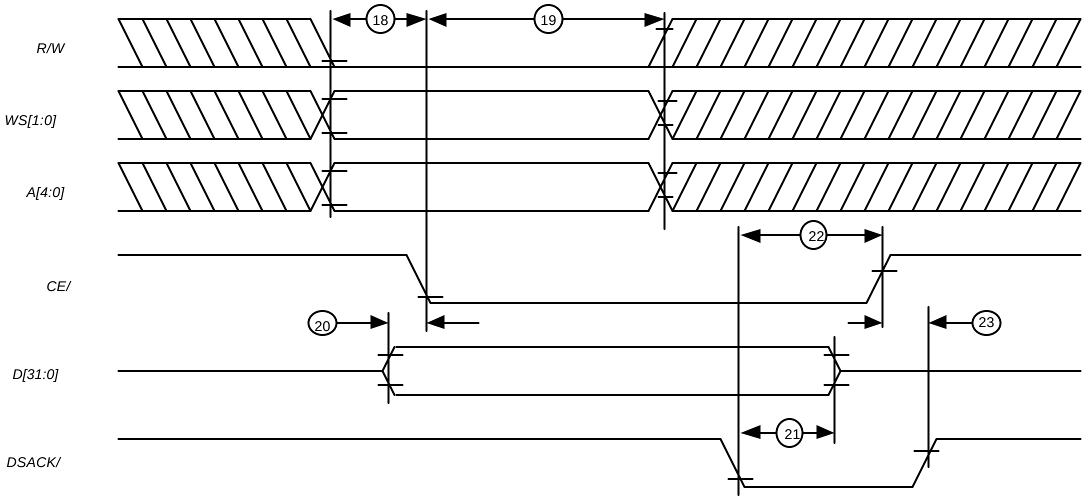


Figure 27. Write-To-MIC Timing

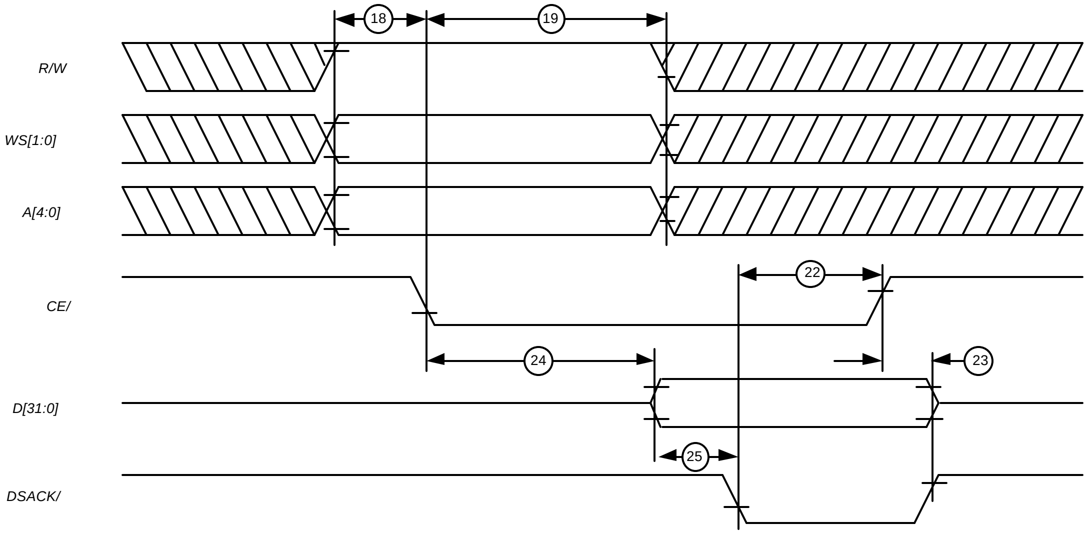


Figure 28. Read-From-MIC Timing

AC Characteristics - Interrupt Acknowledge Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
26	Interrupt Request LOW to Interrupt Acknowledge LOW	0	245K*	Clocks
27	*IACK/* LOW to Interrupt Vector Number *D[7:0]* Valid	--	3	Clocks
28	*D[7:0]* Valid to *DSACK/* LOW (Set-up time)	0	--	ns
29	*IACK/* HIGH to *D[7:0]* High Impedance, *DSACK/* HIGH	--	0.5	Clocks
30	*IACK/* LOW to *INTREQ/* or *ADLD/* HIGH	--	5	Clocks

**Value shown for Number 26 is programmable via the Interrupt Acknowledge Time Out Constant Register.*

Note: *The MIC will place an interrupt vector number on the least significant byte of the data bus when signal AVSEL/ is tied LOW.*

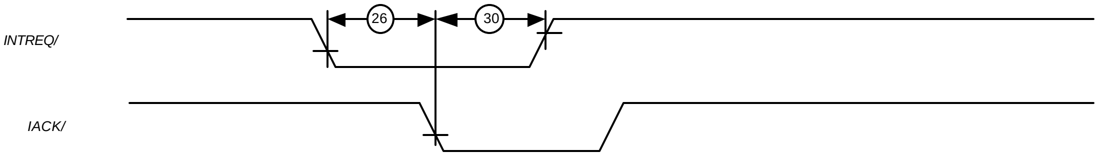


Figure 29. Interrupt Cycle Timing

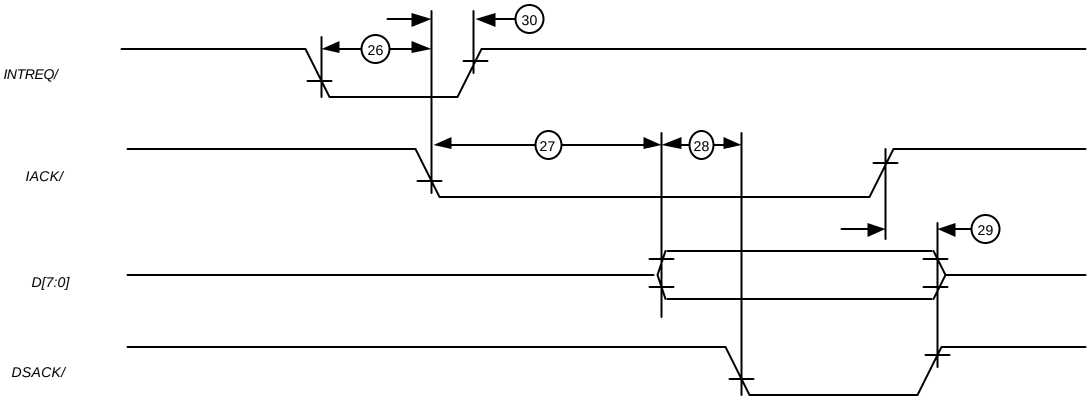


Figure 30. Interrupt Cycle Timing - MIC Generated Vector Number

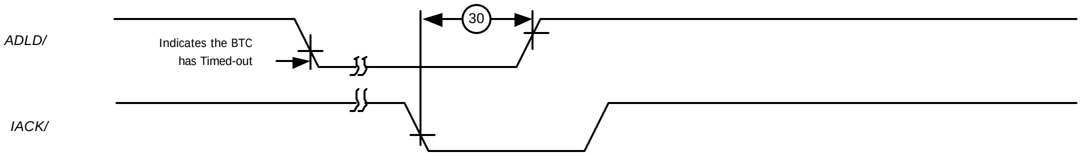


Figure 31. Bus Time-out Indication Timing

AC Characteristics - Remote Switch Mode Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
31	Address Valid to Address Load LOW	2	--	Clocks
32	*ADLD/* LOW to Status (*STAT[1:0]*) Valid	--	2	Clocks
33	*ADLD/* LOW Pulse Width	2	--	Clocks
34	*ADLD/* LOW to *ADLD/* LOW Cycle Time	--	8	Clocks
35	*ADLD/* LOW to *R/W* HIGH (To Suspend Address Cycle)*	--	1.5	Clocks
36	*STAT[1:0]* Valid to *R/W* LOW	0.5	--	Clocks
37	*R/W* LOW to Next Address Valid (2 Rising Clock Min.)	2	--	Clocks

**The MIC signal R/W can be used to suspend the address cycle. If R/W is brought HIGH during RSM operation, the current address will remain valid and the MIC will not store the corresponding STAT data until R/W is brought back LOW. Normal 8-clock address cycles are repeated while R/W is held LOW.*

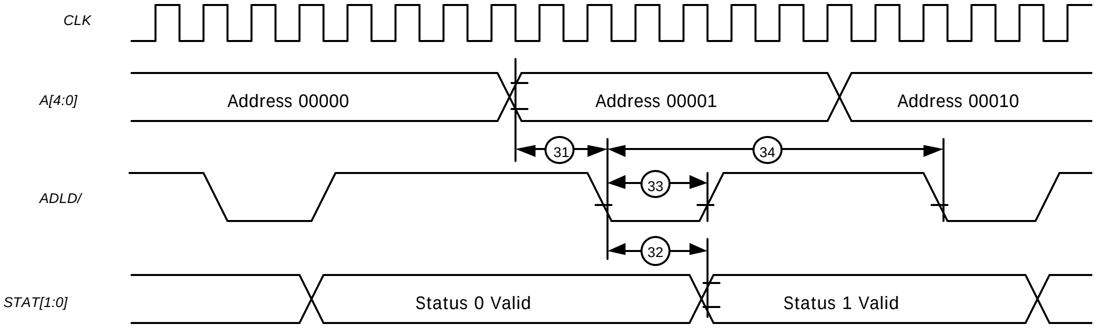


Figure 32. Remote Switch Mode Timing

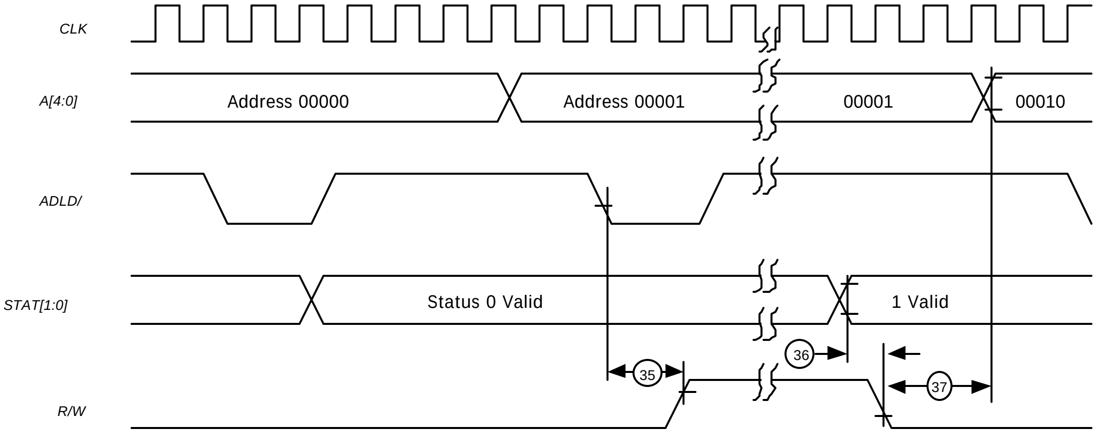


Figure 33. Remote Switch Mode Timing - R/W Control

AC Characteristics - Data Input Mode Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
38	Address (n) Valid to Address Load LOW	2	--	Clocks
39	*ADLD/* LOW to *ADLD/* HIGH and *ADCON/* LOW	--	2	Clocks
40	*ADCON/* LOW Pulse Width	--	2	Clocks
41	*ADLD/* LOW to Device (n) Data Valid (Sample Time)	--	8	Clocks
42	*ADLD/* LOW to Device (n) Data Invalid (Sample Hold)	10	--	Clocks
43	*ADLD/* LOW to *ADLD/* LOW (Addressing Cycle Time)	--	14	Clocks
44	*ADLD/* LOW to *R/W* HIGH (To Suspend Address Cycle) *	--	6	Clocks
45	*ADCON/* LOW to *R/W* HIGH (To Suspend Address Cycle) *	--	4	Clocks
46	(n) Data Valid to *R/W* LOW (Set-up Time)	0	--	ns
47	*R/W* LOW to (n) Data Invalid (Hold Time)	4	--	Clocks
48	*R/W* LOW to Address (n-1) Valid	--	9.5	Clocks

** The MIC signal R/W can be used to suspend the address cycle. If R/W is brought HIGH during DIM operation, the current address will remain valid and the MIC will not store the corresponding D[15:0] data until R/W is brought back LOW. Normal 14-clock address cycles are repeated while R/W is held LOW.*

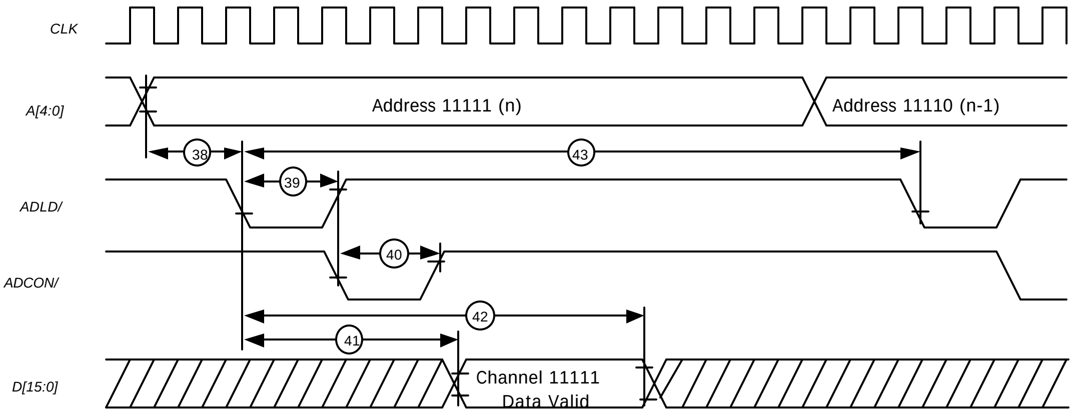


Figure 34. Data Input Mode Timing

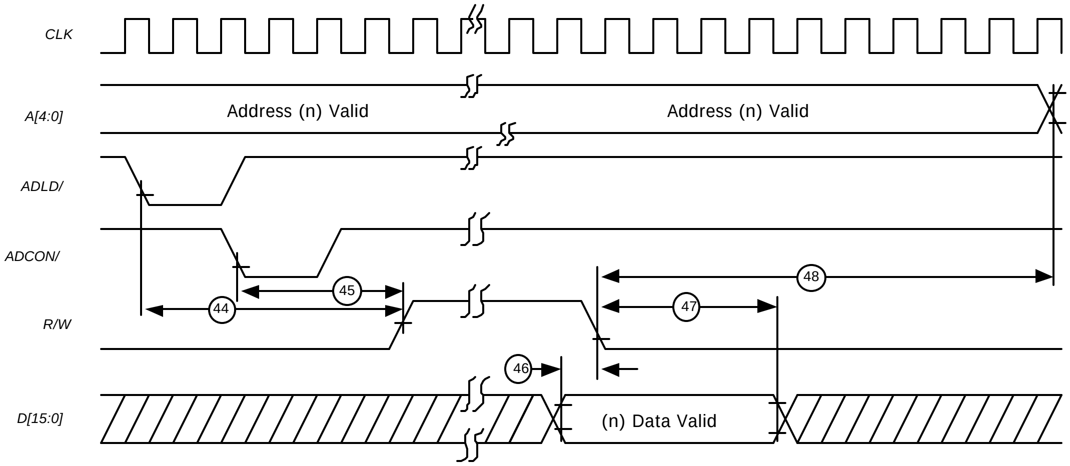


Figure 35. Data Input Mode Timing - R/W Control

AC Characteristics - Data Output Mode/Data Input Mode Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
49	Direction HIGH to First DOM Address Valid	2	--	Clocks
50	First DOM Data Word *D[31:16]* Valid to *A[4:0]* Valid	--	1.5	Clocks
51	*A[4:0]* Valid to Address Load LOW	--	2	Clocks
52	*ADLD/* LOW Pulse Width	--	4	Clocks
53	*ADLD/* LOW to *ADLD/* LOW (DOM Output Cycle Time)	8	--	Clocks
54	*DIR* LOW to *A[4:0]* Invalid and *ADLD/* HIGH	0	--	ns
55	*DIR* LOW to DIM (n) Data Valid	--	1.5	Clocks
56	*DIR* LOW to DIM (n) Data Invalid	4	--	Clocks
57	*IACK/* LOW to DOM (a) Data Valid	--	0.5	Clocks
58	*IACK/* LOW to DOM (a) Data Still Valid	--	4	Clocks
59	*IACK/* HIGH to DOM Data Invalid	0	1	Clocks
60	Last *IACK/* LOW to *DIR* LOW	--	4.5	Clocks

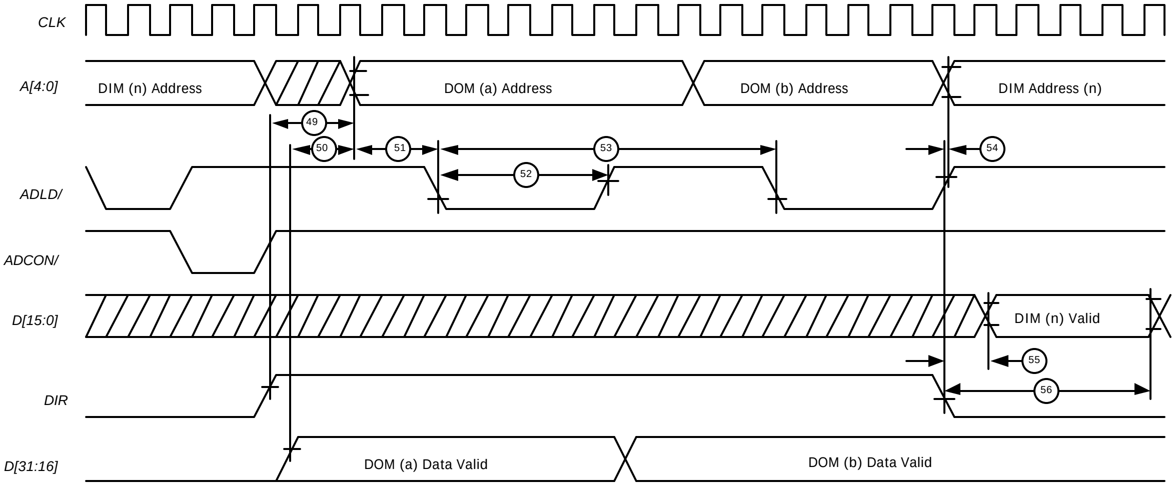


Figure 36. Data Output Mode Timing

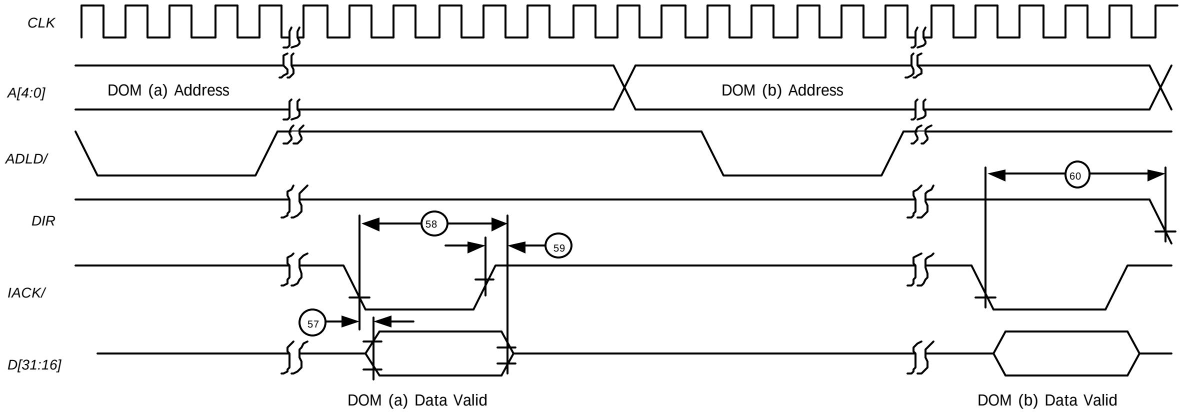


Figure 37. Data Output Mode Timing - IACK Control

AC Characteristics - Remote Switch Mode/Data Input Mode Timing

(At Recommended Operating Conditions)

Number	Characteristic	Min	Max	Unit
61	Last RSM Address Load LOW to DIM Address Valid	--	10	Clocks
62	DIM Address Valid to RSM Address 00H Valid *	--	18	Clocks
63	RSM to DIM Address Switch-over Time (*A[4:0]* = 00H)	--	4	Clocks
64	*D[15:0]* High Impedance to *DIR* LOW (Switch to RSM/DIM)	1	--	Clocks
65	*DIR* HIGH to *D[15:0]* RSM Data Valid (RSM Output)	2	2.5	Clocks
66	RSM Output Valid to *BUSY* HIGH (Data Set-up Time)	1	--	Clocks
67	*BUSY* HIGH Pulse Width	--	2	Clocks
68	*BUSY* LOW to *D[15:0]* High Impedance (Hold Time)	2	3	Clocks
69	*D[15:0]* High Impedance to *DIR* LOW	1	2	Clocks
70	*DIR* HIGH to *DIR* LOW (RSM Output Cycle)	--	10	Clocks
71	Last RSM Address Load HIGH to Beginning of Input Cycle	--	6	Clocks
72	** Output Cycle Overlap With Input Cycle - *Example*	2	--	Clocks
73	** Delay In DIM *ADLD/* Due to Overlap - *Example*	2	--	Clocks

**If the DIM address is not 00 Hex, the value for number 62 would be 14 clocks instead of 18 clock cycles. The value for number 62 may also be different if R/W is used to hold a cycle.*

**External latching circuitry may be required if all 32 output bits are used. Data lines *D[15:0]* are used both as an input bus and an output bus in RSM/DIM. An output cycle is indicated to external logic by the *DIR* signal being held HIGH. Output cycles may overlap with input cycles because serial bus messages are received asynchronously. If an output cycle overlaps with a Status pair input cycle, no delay is created and the input cycling continues. However, if an output occurs after the beginning of a DIM cycle but before the *ADCON/* signal is asserted, a proportional delay (equal to Output passed dashed line in Figure 39) will be present in the DIM cycle.

Note: RSM/DIM Combination Mode is set after receiving the appropriate Set-up Command word. Once in RSM/DIM, the MIC alternates between storing the 32 addressable *STAT[1:0]* bit pairs and the 32 addressable 16-bit words from *D[15:0].* The MIC Cycles through all 32 STAT pairs incriminating the address bus then performs one DIM 16-bit input cycle after which the DIM address register decrements once. This processes is repeated. All control signals are available as described in the RSM and DIM timing diagrams. It takes 256 clock cycles to complete STAT pair addressing and storage and a minimum of 14 clocks per DIM input.

AC Characteristics - RSM/DIM Timing (Continued)

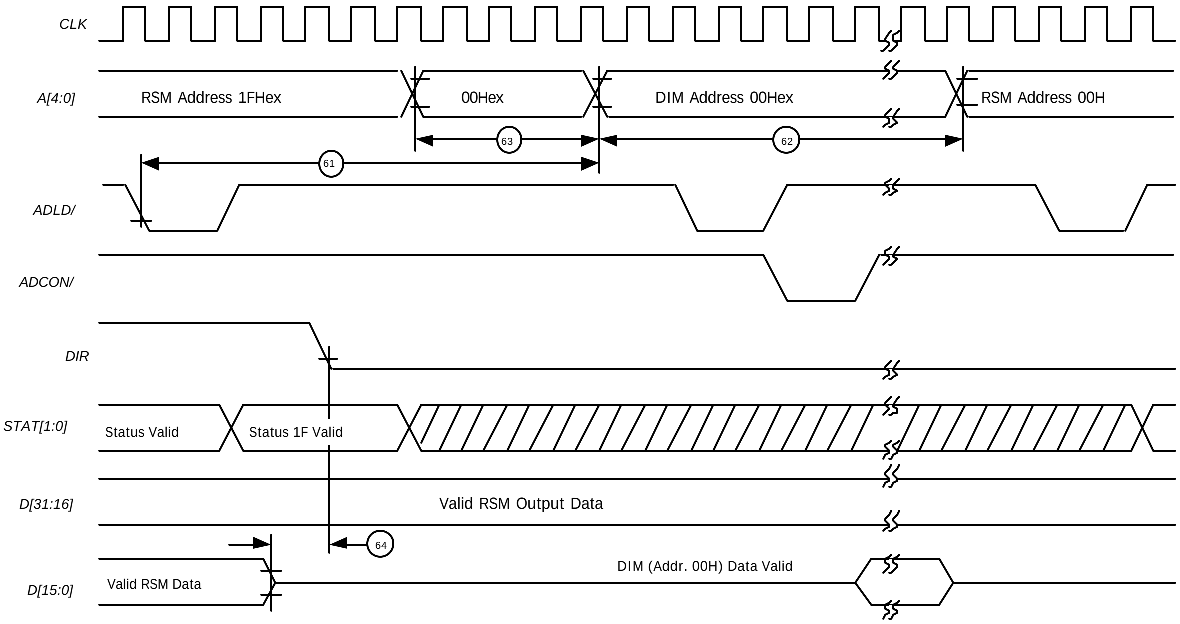


Figure 38. Remote Switch Mode/Data Input Mode Timing

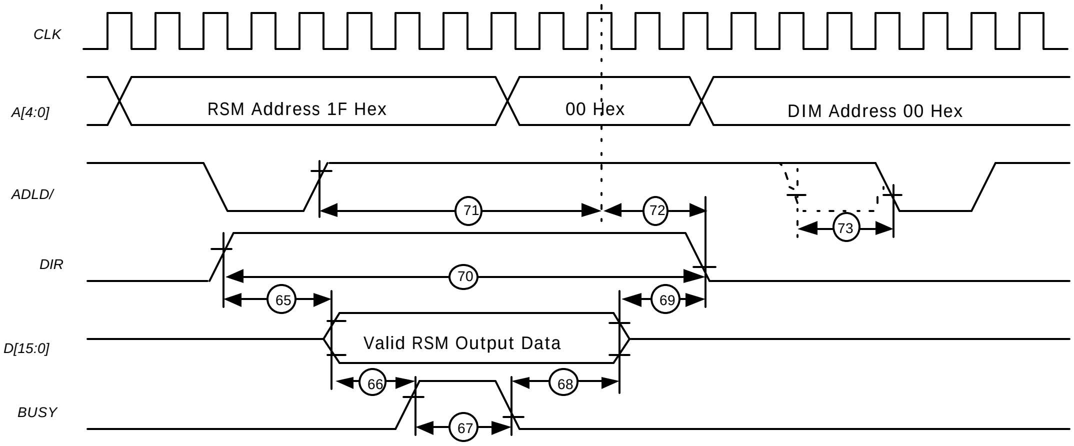


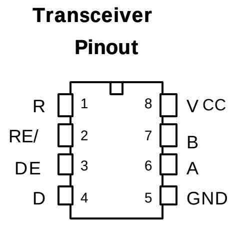
Figure 39. RSM/DIM Timing - Output Cycle

Transceiver Selection

Serial data bus transceivers are required to provide the appropriate interface between the MIC and multipoint bus transmission lines. The MIC was designed so that a variety of physical data bus techniques could be used. Most current MIC Bus applications use devices that meet the Electronic Industries Association (EIA) RS-485 electrical characteristics standard for a direct coupled linear bus configuration. These widely used, low cost bus transceivers allow redundant twisted and shielded wire pairs to be used as the MIC Bus interconnection medium. MIL-STD-1553B transceivers or other standard differential receiver/drivers could be used to provide an electrical interface to a two wire medium. The MIC can also be easily connected to almost any serial data communication link such as, fiber optic cables, radio links, infra-red links, telephone links, etc.. The RS-485 transceiver is summarized in this section because of its popularity in current MIC applications.

RS-485 Transceivers

The MIC Bus implementation used in most current applications require devices that meet the Electronic Industries Association (EIA) RS-485 electrical characteristics for a direct coupled linear bus configuration. This low cost bus architecture allows redundant twisted and shielded wire pairs to be used. Figure 40 summarizes standard RS-485 device functions while Figure 41 shows the signal interface between the MIC and transceivers. Individual vendor data sheets should be referred to for details concerning transceiver characteristics and interface requirements. Texas Instruments, National Semiconductor, and Linear Technology as well as several other vendors each make transceivers that meet EIA RS-485 standards.



Transmit Function

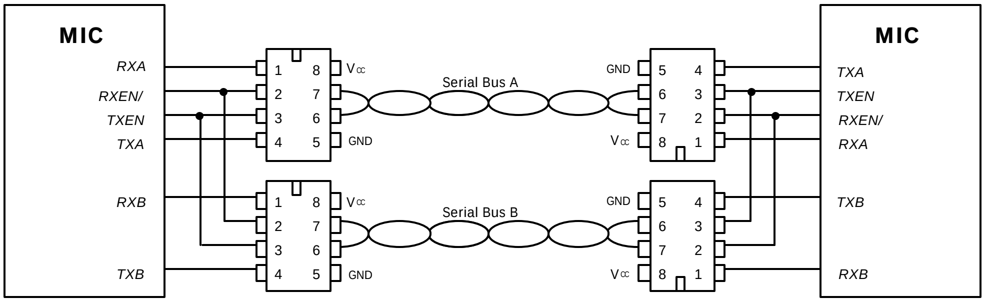
INPUTS			OUTPUTS	
RE/	DE	D	A	B
X	1	1	1	0
X	1	0	0	1
X	0	X	Z	Z

Receive Function

INPUTS			OUTPUT
RE/	DE	A-B	R
0	0	$V_D > 0.2$ V	1
0	0	V_DŠ -0.2 V	0
0	0	$-0.2 < V_D < 0.2$	?
1	0	X	Z

X=Irrelevant , Z= High Impedance
?= Indeterminate (No Differential)
V_D=Differential Input Voltage

Figure 40. RS-485 Transceiver Information

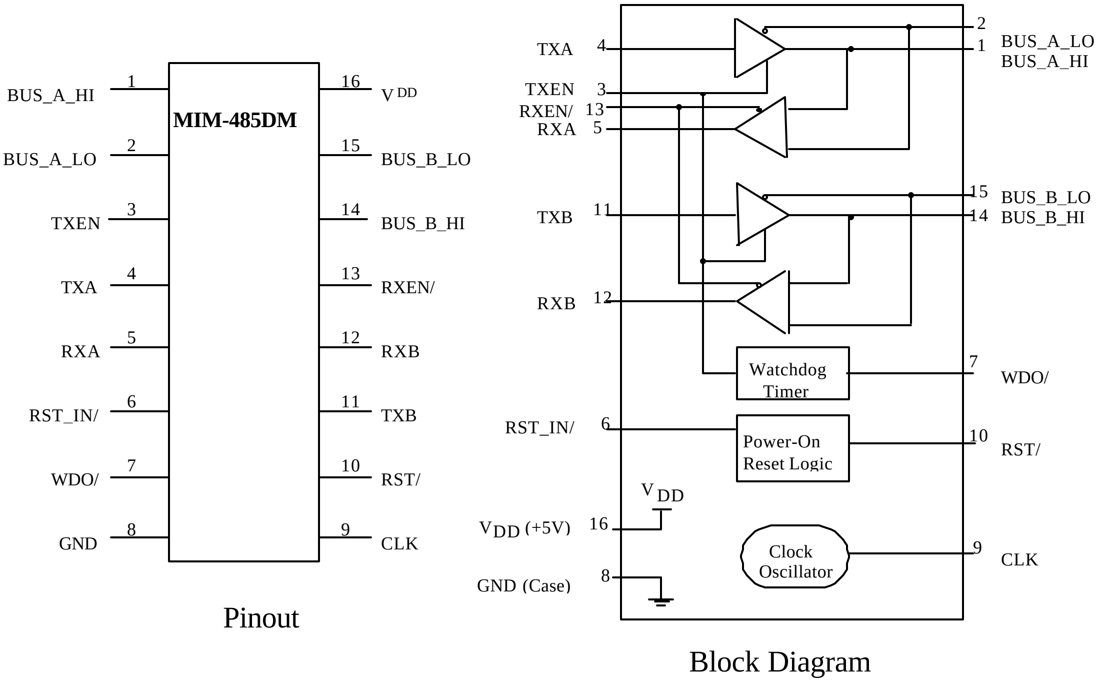


Note: *RS-485 component manufactures recommend using 120 twisted and shielded wire pairs with a 120 termination resistor located at each end of the longest span of bus. RS-485 networks are limited to 32 nodes per bus.*

Figure 41. RS-485 Transceiver Interface to MIC

Vetronix MIM-485DM

Vetronix has developed a MIC Bus Interface Module which contains 2 RS-485 transceivers, power-on-reset logic, and a clock oscillator. This component provides a simplified interface to the MIC when compared to discrete components. Figure 42 shows the MIM-485DM pinout and functional block diagram. The MIM-485DM is packaged in a 16 pin, 300 mil wide, dual in-line metal package. Contact Vetronix for further information about the MIM-485DM and related components.



Pinout

Block Diagram

Figure 42. MIM-485DM Block Diagram

APPLICATIONS

MIC Bus Development System - MBDS

The MBDS is a serial data bus development/analysis system designed specifically for the communications protocol of the MIC. The MBDS-EX is a full size IBM PC-AT printed circuit card and software which, when configured with a PC-AT compatible can be used to aid in remote module design, application software development, systems integration, and system verification for applications using the MIC Bus architecture. User-friendly menus and software allow the operator to use the MBDS as a bus controller, bus monitor, or remote module emulator. The MBDS is capable of implementing the full MIC protocol and contains diagnostics functions to test RS-485 bus networks.

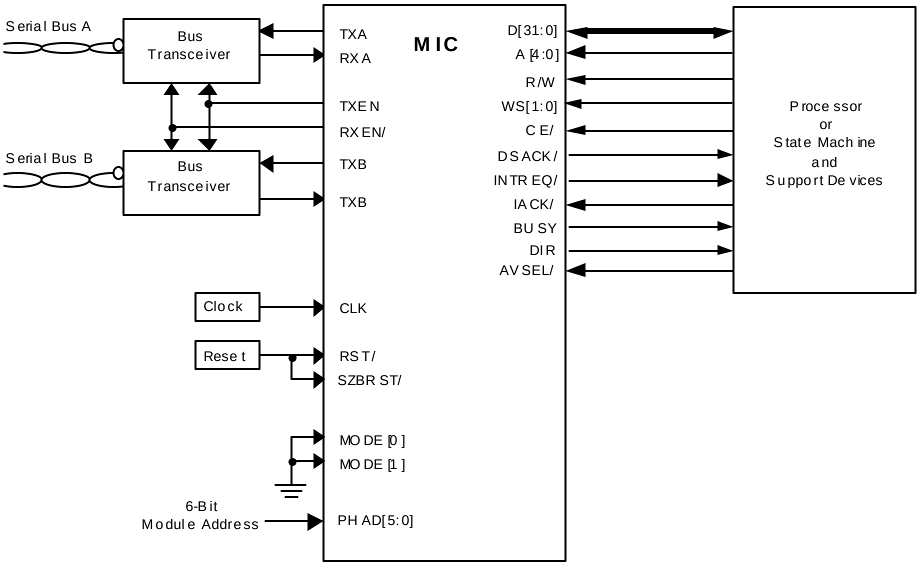
The MBDS-VME is a serial data bus development and analysis system designed specifically for the communications protocol of the MIC. The MBDS-VME-545 is a 6U X 160mm VME printed circuit card while the MBDS-VME-525 is a 3U X 160mm VME printed circuit card. When configured with a VME bus master, this development and analysis system can be used to aide in remote module design, software development, systems integration, and system verification for applications using the MIC Bus architecture. The on-board firmware allows the MBDS-VME to be used as a bus controller, bus monitor, or remote module. The MBDS-VME is capable of implementing the full MIC protocol and contains diagnostics functions to test MIC Bus networks using RS-485 transceivers.

For further information about the MBDS or other MIC Bus related products please contact Vetronix.

Application Examples

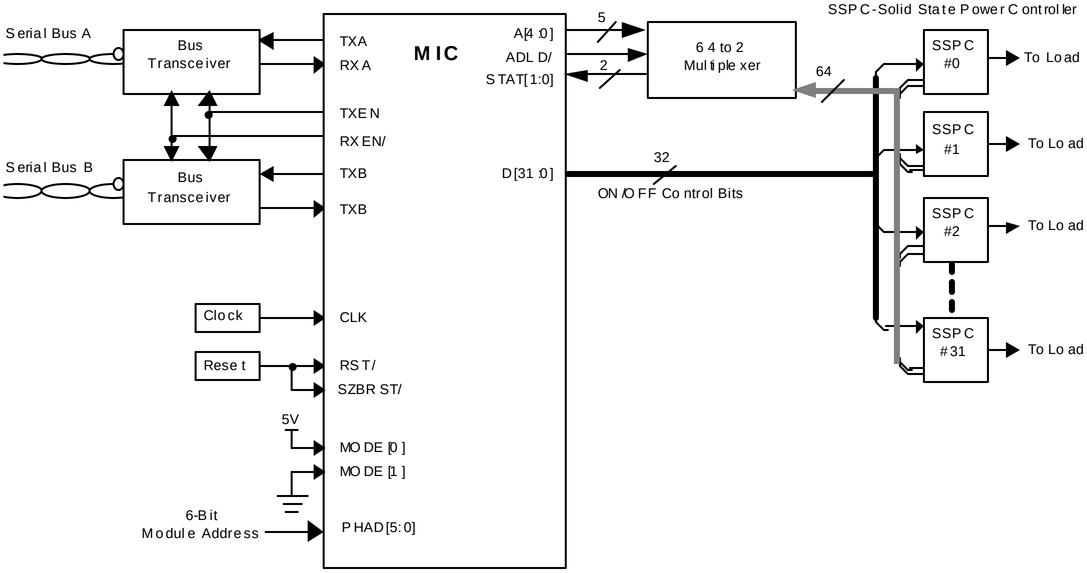
The following figures are intended to give examples of various devices and data that the MIC Bus architecture can be used to control. For specific information about a current application or assistance determining the suitability of the MIC to a new application please contact Vetronix.

Processor Interface Mode.

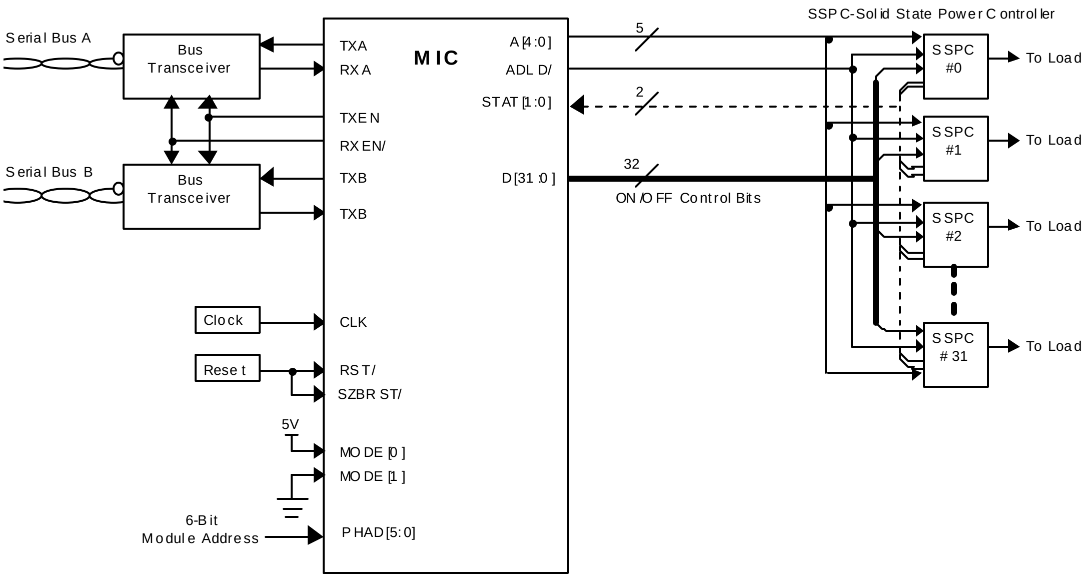


Remote Switch Mode Interface to Solid-State Power Controllers

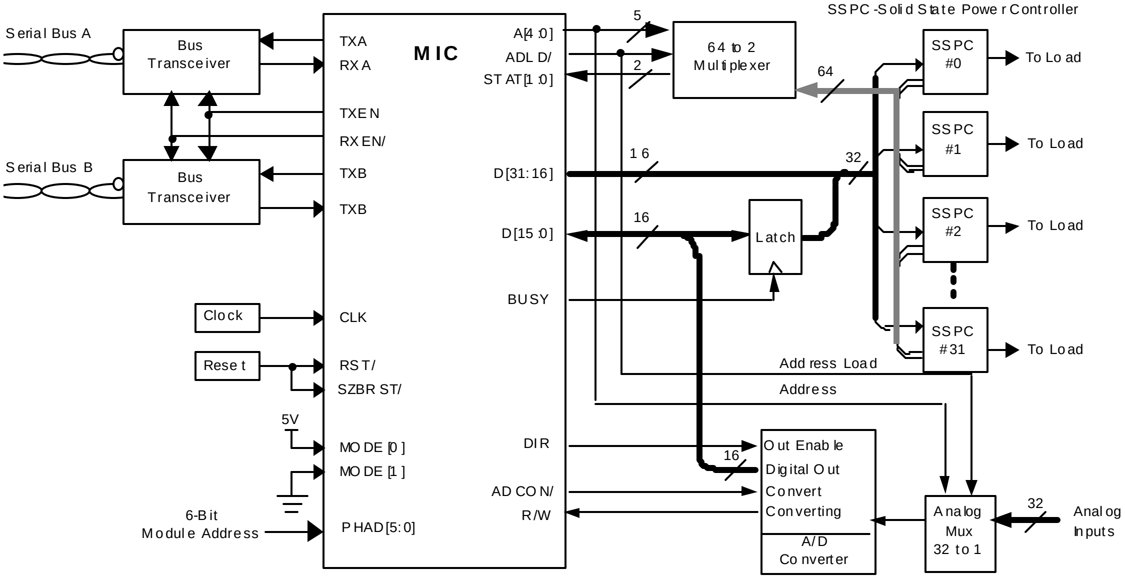
(Without Addressable Status Bits).



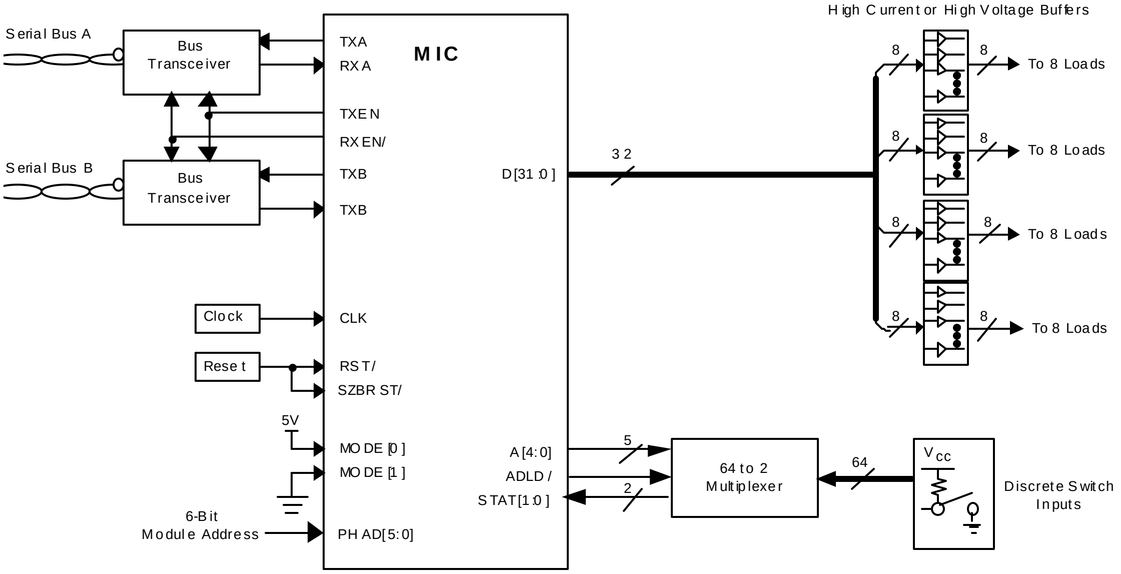
Remote Switch Mode Interface to Solid-State Power Controllers
(With Addressable Status Bits).



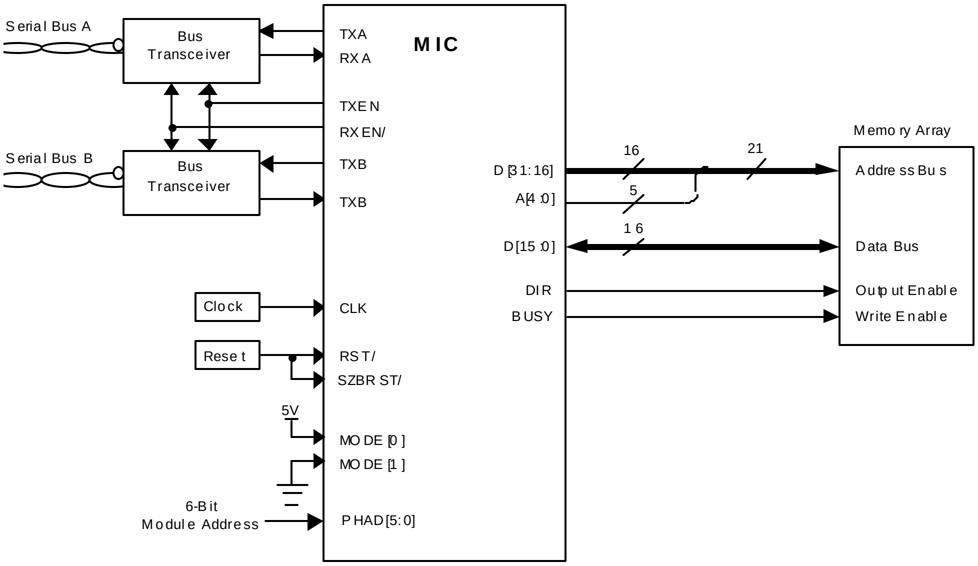
Remote Switch Mode Interface to Solid-State Power Controllers and Analog-to-Digital Convert.



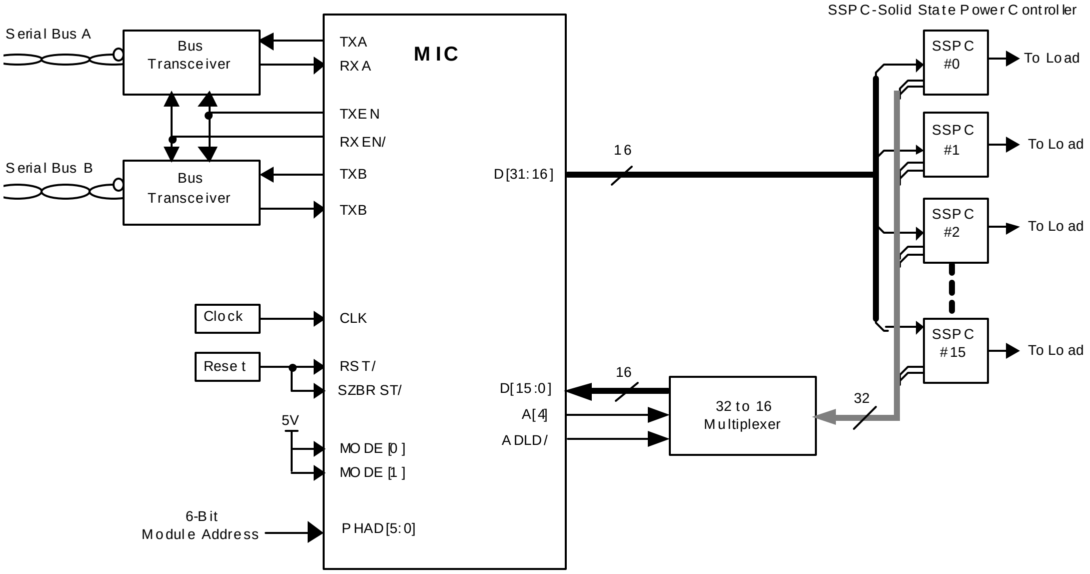
Remote Switch Mode Interface to High Voltage and/or High Current Buffers and Discrete Switch Inputs.



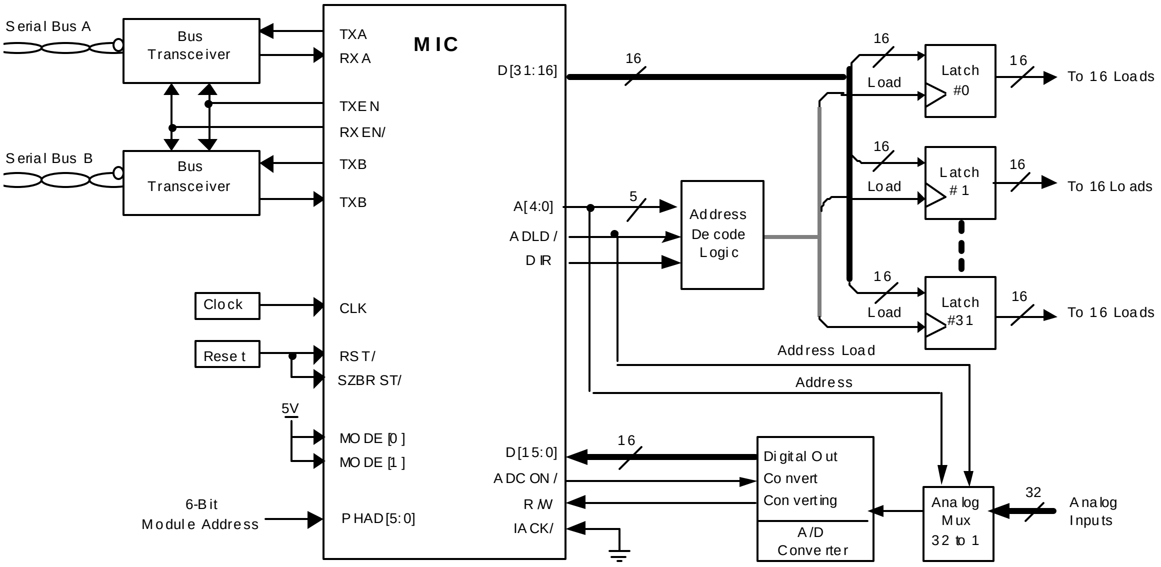
Remote Switch Mode Interface to Memory.



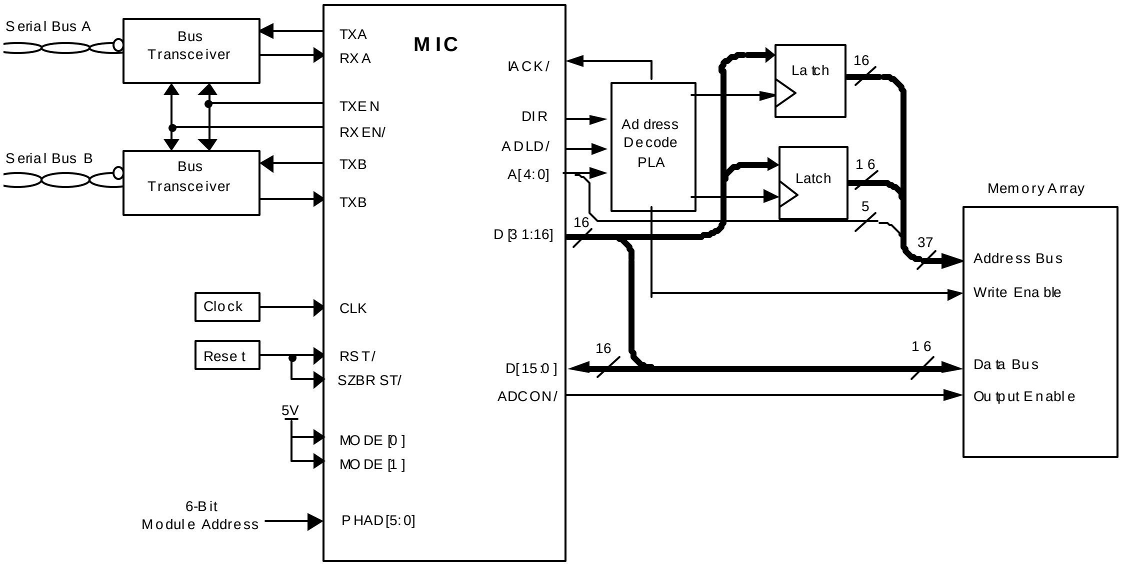
Data Output Mode Interface to Solid-State Power Controllers.



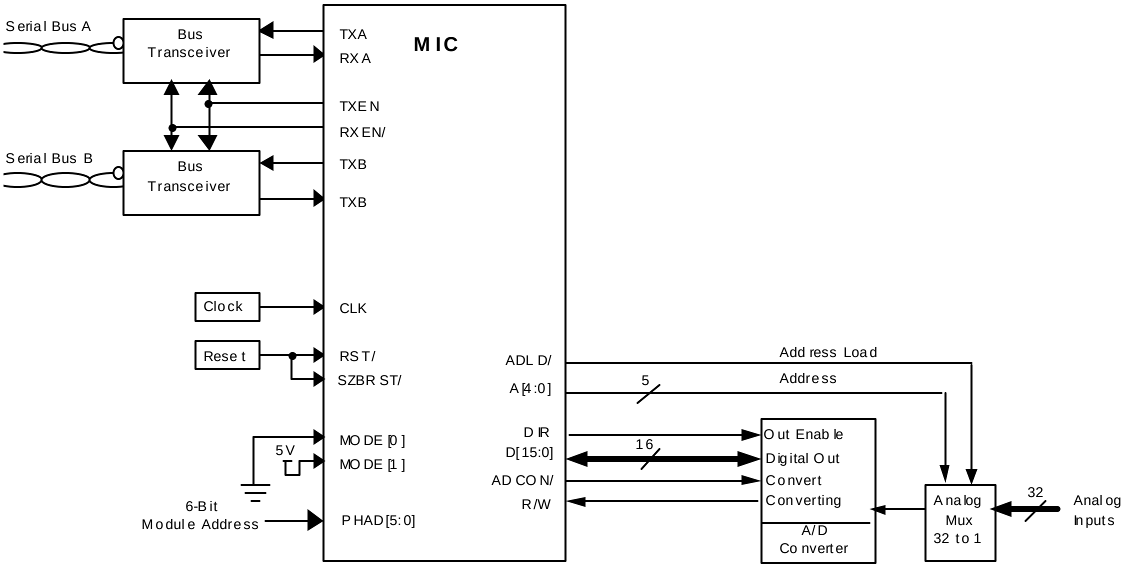
Data Output Mode Interface to Latches and Analog-to-Digital Converter.



Data Output Mode Interface to Memory.



Data Input Mode Interface to Analog-to-Digital Converter.


Serial Bus A
Bus Transceiver
Serial Bus B
Bus Transceiver
Clock
Reset
5V
6-Bit Module Address
TXA
RXA
MIC
TXEN
RXEN/
TXB
TXB
CLK
RST/
SZBRST/
MODE[0]
MODE[1]
PHAD[5:0]
ADLD/
A[4:0]
DIR
D[15:0]
ADCON/
R/W
Address Load
Address
5
16
Out Enable
Digital Out
Convert
Converting
A/D Converter
Analog Mux 32 to 1
32
Analog Inputs