Xycom XVME-230
Intelligent Counter Module

In Stock

Used and in Excellent Condition

Open Web Page

https://www.artisantg.com/60911-3

All trademarks, brandnames, and brands appearing herein are the property of their respective owners.

ARTISAN TECHNOLOGY GROUP

Your definitive source for quality pre-owned equipment.

Artisan Technology Group

(217) 352-9330 | sales@artisantg.com | artisantg.com

- Critical and expedited services
- In stock / Ready-to-ship
- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

XVME-230

Intelligent Counter Module

P/N 74230-001C

© 1997 XYCOM, INC.

Printed in the United States of America

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Revision	Description	Date
A	Manual Released	9/84
B	Manual Updated	2/85
C	Manual Updated (incorporated PCN #115)	6/93

Trademark Information

Brand or product names are registered trademarks of their respective owners.

Copyright Information

This document is copyrighted by Xycom Incorporated (Xycom) and shall not be reproduced or copied without expressed written authorization from Xycom.

The information contained within this document is subject to change without notice. Xycom does not guarantee the accuracy of the information and makes no commitment toward keeping it up to date.

xycom
Technical Publications Department
750 North Maple Road
Saline, MI 48176–1292
734-429-4971 (phone)
734-429-1010 (fax)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS (continued)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS (continued)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS (continued)

LIST OF TABLES

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS (continued)

LIST OF FIGURES

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

The XYCOM XVME-230 Intelligent Counter Module (ICM) is a VMEbus compatible slave I/O module engineered to provide the following high level measurement and control functions for VMEbus systems:

- Event Counting
- Frequency/Duty Cycle Generation
- Period Measurement
- Frequency Measurement
- Frequency Division
- Position Measurement
- Stepper Motor Control
- Period/Pulse Width Generation
- Periodic Interrupts
- Period/Pulse Width Measurement

The ICM performs these functions in a "stand-alone" role after a VMEbus host processor initiates them. Multiple functions can operate simultaneously.

1.2 MANUAL STRUCTURE

This chapter introduces the functional architecture and general specifications of the XVME-230 Intelligent Counter Module. Also mentioned is the XYCOM Standard I/O Architecture. Successive chapters explain the operation of the ICM using the following procedure:

Chapter Two -- A presentation of the physical and logical functional architecture of the ICM with specific focus on module base addressing, module identification data and module operational status.

Chapter Three -- A description of XYCOM's Interprocessor Communications protocol, the method used to ensure reliable information transfer between XYCOM intelligent modules.

Chapter Four -- A description of the ICM counter architecture and the general procedure used for initializing function operation using Interprocessor Communications protocol.

Chapter Five -- Module installation information covering module specific system requirements, jumpers, and external connections.

Chapters Six, Seven, Eight, Nine, Ten, and Eleven -- Explain the operation of the ICM's measurement and control functions: Event Counting, Periodic Interrupts, Frequency/Duty Cycle Control, Period/Pulse Width Generation, Period/Pulse Width Measurement, Period Measurement, Frequency Division, Frequency Measurement, Position Measurement, and Stepper Motor Control.

The appendices contain information on XYCOM VMEbus Standard I/O Architecture, backplane signal/pin descriptions, module schematics, and a quick reference of important jumpers, addresses, and commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3 ICM FUNCTIONAL OVERVIEW

The ICM is comprised of two sections: the application circuitry needed to perform the measurement and control functions, and a standard XYCOM intelligent I/O "kernel" that interfaces the application circuitry to the VMEbus. An operational block diagram showing these sections is presented in Figure 1-1 below.

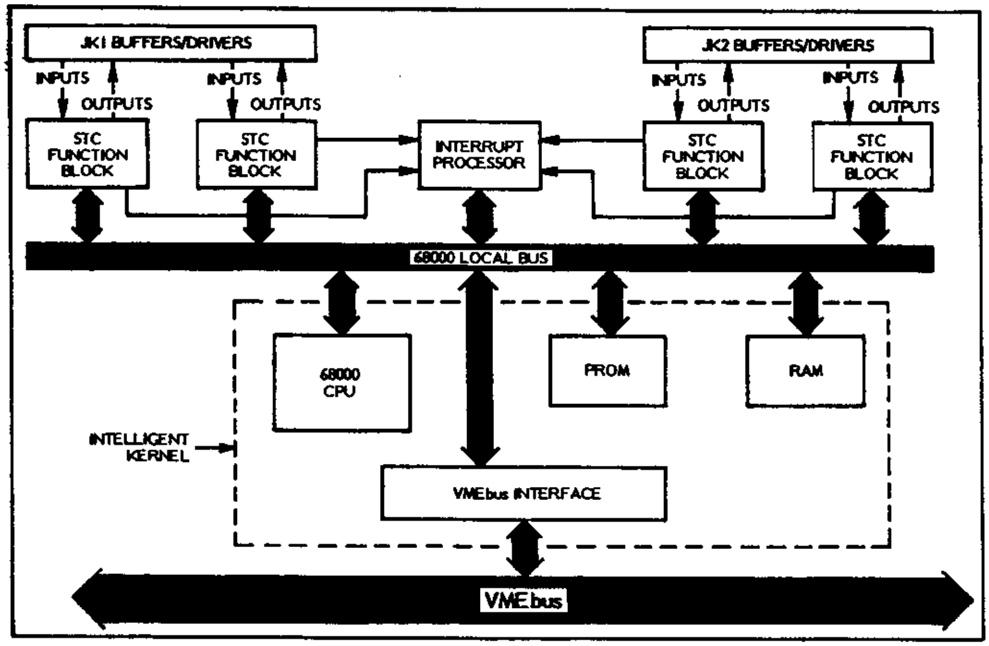


Figure 1-1. Operational Block Diagram of ICM XVME-230

1.3.1 Counter Circuitry

The ICM counter circuitry consists mainly of four programmable System Timing Controllers (STCs), (see Figure 1-1). Each STC provides the ICM with four high-speed counters, for a total of 16 counters.

The counters are configured individually, or in groups of up to four, in order to perform the ICM's various measurement and control functions. All the configuring and controlling of the counters is automatically handled by the on-board 68000 micro-processor-based intelligence (firmware stored in EPROM).

The counter input/output signals are buffered and made available on the ICM's two 50 pin connectors, JK1 and JK2, (see Chapter 4, Section 4.6 for a description of the counter interface signals).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The ICM's measurement or control functions are specified by commands sent to one of eight logical "channels." This allows the user to program the ICM to run many different functions concurrently (see Chapter 4 for the maximum number of functions able to operate simultaneously). The commands are sent using the "IPC protocol" which is defined in Chapter 3. The use of the terms "channel" and "counter," as part of the ICM's logical architecture, are defined in Chapter 4.

1.3.2 XYCOM Intelligent I/O Kernel

The ICM is based upon a XYCOM intelligent I/O kernel, which is built around a 10MHz 68000 microprocessor unit. This kernel, in combination with on-board firmware, performs many of the time-consuming tasks that are inherent to measurement and control functions. The intelligent kernel, in conjunction with interrupt control circuitry, interfaces the ICM measurement and control functions to the VMEbus (see Figure 1-1).

The XYCOM intelligent I/O kernel has the following features:

- 68000 microprocessor with 7 internal interrupt levels
- 2 PROM and 2 RAM sockets which accept various sizes
- Slave functions (A16:D16)
- VMEbus Requester (R0-R3)
- VMEbus Interrupter (I1-I7)
- Programmable Address Decode Circuitry
- Self-Test Status Register
- Module Identification
- PASS, FAIL, and HALT LED indicators

1.4 XYCOM STANDARD I/O ARCHITECTURE

The ICM is designed in accordance with XYCOM's Standard I/O Architecture which is specified for all XVME I/O modules in order to provide a simple and consistent method of programming for the entire module line. The features standardized by the XYCOM I/O Architecture are the following:

1. Module Addressing -- Where a module is positioned in memory and how software can read from it or write to it.

2. Module Identification -- How software can identify which modules are installed in a system.

3. Module Operational Status -- How the operator can determine, through software, the "health" of specific modules within the system.

4. Interrupt Control -- How software controls and monitors the capability of the module to interrupt the system.

5. Communication between Modules -- How master processors and intelligent I/O modules communicate through shared or global memory, or through the I/O modules dual-access RAM.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6. Design Consistency -- How intelligent and non-intelligent "kernels" are common to all XYCOM I/O modules.

The standard is defined in detail in Appendix A at the rear of this manual.

1.5 SPECIFICATIONS

The following is a list of operational and environmental specifications for the Intelligent Counter Module.

Table 1-1. Intelligent Counter Module Specifications

Characteristic	Specification
System Time Base	
Frequency	5MHz
Accuracy/Stability	±.005%
Function Specifications	
Event Counting	
Circuit input range	Up to 5 MHz
Count range returned	Up to 2^{16}
(two options)	Up to 2^{32}
Frequency Measurement	
Circuit input range	Up to 5 MHz
Accuracy ranges:	
5 MHz to 400 Hz	0.025% or less error from total measurement
400 Hz to 10 Hz	1.0% or less error from total measurement
Below 10 Hz	% Error = $\frac{0.1 \text{ Hz}}{\text{Freq. in Hz}} * 100$
Period Measurement	
Circuit input range	858 sec to 200 nsec
Accuracy ranges:	
858 sec to 819.2 usec	0.025% or less error from total measurement
819.2 usec to 20 usec	1.0% or less error from total measurement
From 20 usec to 200 nsec	% Error = $\frac{200 \text{ nsec}}{\text{Period in nsecs}} * 100$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 1-1. Intelligent Counter Module Specifications (continued)

Characteristic	Specification
Period/Pulse Width Measurement	
Circuit input range	858 sec to 200 nsec
Accuracy ranges:	
858 sec to 819.2 usec	0.025% or less error from total period measurement
819.2 usec to 20 usec	1.0% or less error from total period measurement
From 20 usec to 200 nsec	% Error = $\frac{200 \text{ nsec}}{\text{Period in nsecs}}$ * 100 * Note: Pulse width must be at least 200nsec. for measurements to be taken.
Frequency/Duty Cycle Generation	
Circuit output range	1 - 100,000 Hz
Accuracy ranges:	
1 Hz to 1220 Hz	0.025% or less error from specified range
1220 Hz to 50 KHz	1.0% or less error from specified range
50 KHz to 100 KHz	% Error = $\frac{\text{Freq. in Hz}}{5 \text{ MHz}}$ * 100
Period/Pulse Width Generation	
Circuit output range	10 us - 10 s
Accuracy ranges:	
10 sec to 819.2 usec	0.025% or less error from specified range
819.2 usec to 20 usec	1.0% or less error from specified range
From 10 usec to 20 usec	% Error = $\frac{200 \text{ nsec}}{\text{Period in nsecs}}$ * 100
Position Measurement	
SIN or COS input range	Up to 500 KHz
Accuracy	$\pm$ 1/4 Wave
Stepper Motor Control	
Maximum step rate	65535 Hz
Accuracy ranges:	
1 Hz to 1220 Hz	0.025% or less error from specified range
1220 Hz to 50 KHz	1.0% or less error from specified range
From 50 KHz to 65 KHz	% Error = $\frac{\text{Freq. in Hz}}{5 \text{ MHz}}$ * 100

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 1-1. Intelligent Counter Module Specifications (continued)

Characteristic	Specification
Power Requirements	+5 volts ±5% 4.6 Amps typical 5.2 Amps maximum
Input Buffers	
Low level input current	-200 uA
High level input current	20 uA
Output Buffers	
Low level output current	24 mA
High level output current	-15 mA
Temperature	
Operating	0 to 65°C
Non-Operating	-40° to 85°C
Humidity	5 to 95% RH non-condensing (Extreme low humidity conditions may require special protection against static discharge.)
Altitude	
Operating	Sea-level to 20,000 ft. (6096m)
Non-Operating	Sea-level to 50,000 ft. (15240m)
Vibration	
Operating	5 to 2000 Hz .015 inches peak-to-peak displacement 2.5 g peak (maximum) acceleration
Non-Operating	5 to 2000 Hz .030 inches peak-to-peak displacement 5.0 g peak (maximum) acceleration
Shock	
Operating	30 g peak acceleration, 11 msec duration
Non-Operating	50 g peak acceleration, 11 msec duration
VMEbus Compliance	• Fully compatible with VMEbus standard • A16:D16 DTB Slave • Selectable base starting address on 1K boundaries within the first block of 16K in the short I/O address space • Responds to supervisor or user A.M. codes (STAT) • Any of I1 - I7 Interrupter (DYN) • NEXP board size (160mm x 233.4mm) (P2 used for power only)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 2

ICM ARCHITECTURE

2.1 INTRODUCTION

The ICM design incorporates the XYCOM Standard I/O Architecture (described in Appendix A). The following architecture aspects are specific to the ICM's installation and functional operation:

Base Addressing: where the ICM can be addressed in VMEbus memory.

ICM I/O Address Space: layout of ICM's "I/O Interface Block" presenting addresses used for ICM operation.

ICM Identification Data: location and description of module identification data.

ICM Operational Status: location and description of information pertaining to the operational status of the ICM.

ICM Communication: location of the ICM's eight communication channels.

2.2 BASE ADDRESSING

All XYCOM I/O modules are designed to be addressed within the VMEbus-defined 64K short I/O address space. Since each I/O module connected to the bus must have its own unique base address, the base addressing for XYCOM VME I/O modules is jumper-selectable. When the ICM is installed in the system, it occupies a 1K byte block of the short I/O memory (called the I/O interface block). The base address decoding scheme for XVME I/O modules is such that the starting address for each I/O Interface Block resides on a 1K boundary (refer to Figure 2-1).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

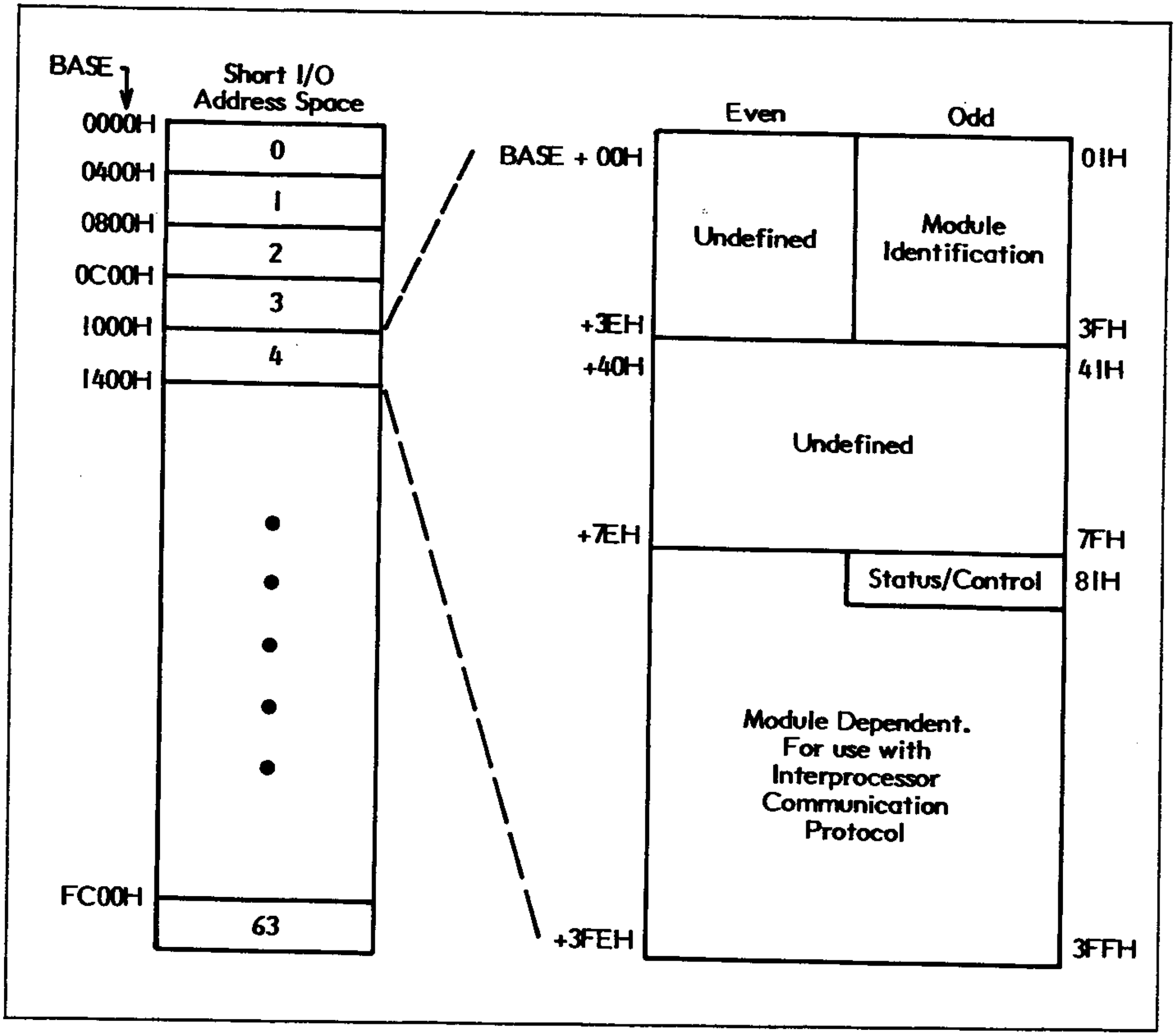


Figure 2-1. The ICM I/O Interface Block and
the Short I/O Address Space

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The ICM decodes only Short I/O Address bits A10 through A13 (A14 and A15 must be zero), and is thus limited to using one of sixteen of the sixty-four possible 1K segments in the short I/O memory. The 16 1K boundary addresses which can be used as base addresses are:

Table 2-1. Legal ICM Base Addresses

0000H
0400H
0800H
0C00H
1000H
1400H
1800H
1C00H
2000H
2400H
2800H
2C00H
3000H
3400H
3800H
3C00H

Any location within the 1K I/O Interface Block can be accessed by adding the Base Address to the I/O Interface Block offset. For example, the module status register is located at address 81H, and if the base address is jumpered to 0C00H, then the status register can be accessed at address 0C81H.

(Module Base Address)		(I/O Interface Block Offset)		(Status Register)
0C00H	+	81H	=	0C81H

For memory-mapped CPU modules (such as 68000 CPU modules), the Short I/O Address Space is memory-mapped to begin at a specific address. For such modules, the I/O Interface Block offset is an offset from the start of this memory-mapped Short I/O Address Space. For example, if the Short I/O Address space of a 68000 CPU module starts at F90000H and the base address of the ICM is jumpered to 0C00H, the actual module base address would be F90C00H.

2.3 ICM I/O INTERFACE BLOCK

The I/O Interface Block (1K of address space allotted to the ICM, see Figure 2-1) is divided into three specific areas dedicated to performing the following functions:

- Identification of the ICM
- ICM Operational Status
- Communication with other VMEbus modules

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.1 ICM Identification

For easy identification in a VMEbus system configuration, I.D. information is located at addresses (base address + 1) to (base address + 31H) in the odd bytes only (see Figure 2-1).

The ICM uses the XYCOM module identification scheme which provides a method of uniquely storing module-specific information in an ASCII encoded format. The I.D. data is provided as thirty-two ASCII encoded characters consisting of the board type, manufacturer identification, module model number, number of 1K byte blocks occupied by the module, and model functional revision level information. This information can be read by the system processor on power-up to verify the system configuration and operational status. Figure 2-2 defines the Identification information locations.

Offset Relative to Module Base	Contents	ASCII Encoding (in hex)	Descriptions
1	V	56	
3	M	4D	ID PROM identifier,
5	E	45	always "VMEID"
7	I	49	(5 characters)
9	D	44	
B	X	58	Manufacturers I.D.,
D	Y	59	always "XYC" for XYCOM
F	C	43	Modules (3 characters)
11	2	34	
13	3	32	Module Model Number
15	0	30	(3 characters and
17		20	4 trailing blanks)
19		20	
1B		20	
1D		20	
1F	1	31	Number of 1K byte blocks of I/O space occupied by this module (1 character)
21		20	Major functional revision
23	1	31	level with leading blank (if single digit)
25	0	30	Minor functional revision
27		20	level with trailing blank (if single digit)
29	Undefined		
2B	"		
2D	"		
2F	"		
31	"		
33	"		
35	"		Manufacturer
37	"		Dependent Information,
39	"		Reserved for Future Use
3B	"		
3D	"		
3F	"		

Figure 2-2. Identification Data

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Because each module I.D. data location is accessed by odd backplane addresses only, the 32 bytes of ASCII data are assigned to the first 32 odd I/O interface block bytes (odd bytes 1H - 3FH) allowing I.D. information to be accessed by addressing the module base, offset by the specific address for the character(s) needed.

For example, if the base address of the board is jumpered to 1000H, and if you wish to access the module model number (I/O interface block locations 11H, 13H, 15H, 17H, 19H, 1BH, and 1DH), you will individually add the offset addresses to the base addresses to read the hex coded ASCII value at each location. Thus, in this example, the ASCII values which make up the module model number are found sequentially at locations 1011H, 1013H, 1015H, 1017H, 1019H, 101BH, and 101DH.

2.3.2 **ICM Operational Status**

The ICM performs a self-test upon power-up or reset. The user can verify the results of this diagnostic test, either by observing the red and green module status LEDs located on the front panel of the ICM, or by reading the module's status register. The Status register occupies byte address; base address+81H.

The Status register format shown below (address = base address+81H) is standard for all XYCOM intelligent I/O modules:

MODULE STATUS/CONTROL

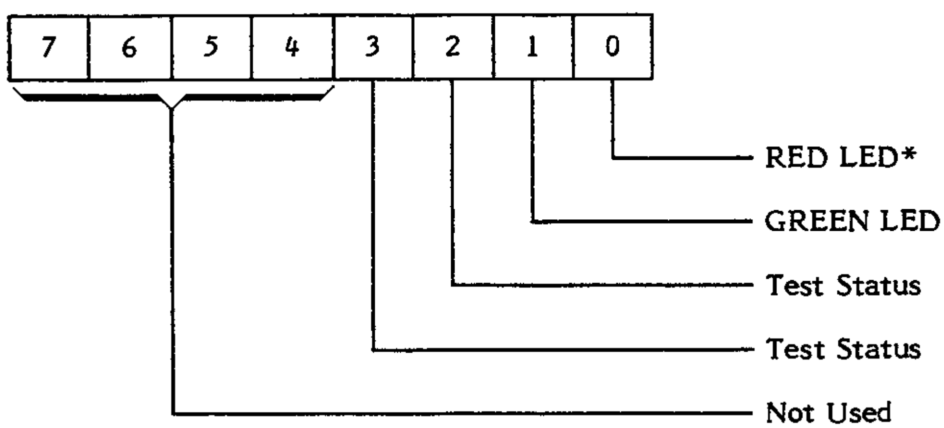


Figure 2-3. The Status/Control Register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The relation of the ICM's LEDs to its operational status is shown in Figure 2-4 below:

INTELLIGENT MODULE STATUS

Status Bits				LEDs			
3	2	1	0	Green	Red*	SYSFAIL*	Status
0	0	0	0	OFF	ON	ON	Module not tested yet
1	0	0	0	OFF	ON	ON	Module failed test
0	0	0	1	OFF	OFF	ON	Inactive module
0	1	1	0	ON	ON	ON	Module undergoing test
1	1	1	1	ON	OFF	OFF	Module passed test
all others				X	X	X	Invalid and undefined

Figure 2-4. LED States

2.3.3 ICM Interprocessor Communication Architecture

A portion of the I/O Interface Block on the ICM is dedicated for use with the XYCOM "Interprocessor Communication Protocol" (IPC protocol is explained in Chapter 3 of this manual). Communication is conducted in the module locations (base address + 82H) to (base address+27EH) (see Figure 2-1) which provide eight logical I/O channels for communication with the host and other intelligent processors.

To implement the eight communication channels, these module locations are divided into the following three areas:

- Eight I/O Request Registers
- Eight Command Block Pointers
- Dual Access RAM used for Command Block location

2.3.3.1 I/O Request Registers

The ICM has eight I/O request registers each of which occupy one byte of the ICM's I/O Interface Block. Each I/O request register is assigned to one of the ICM's eight channels as shown in the diagram below:

82H	Channel 0 I/O Request Register	Channel 1 I/O Request Register	83H
84H	Channel 2 I/O Request Register	Channel 3 I/O Request Register	85H
86H	Channel 4 I/O Request Register	Channel 5 I/O Request Register	87H
88H	Channel 6 I/O Request Register	Channel 7 I/O Request Register	89H

Figure 2-5. ICM I/O Request Registers

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.3.2 Command Block Pointers

The ICM has eight Command Block Pointer locations each of which consumes six bytes of the ICM's I/O Interface Block. The Command Block Pointer specifies to the I/O module the location of the Interprocessor Communication Command Blocks which have been written to the ICM's dual access RAM.

One Command Block Pointer is assigned to each one of the eight channels and has the format shown in the diagram below:

	EVEN	ODD	
92H	Word Alignment Byte	Command Block Pointer Address Modifier Code	93H
94H	Channel 0 Command Block Pointer MSB	Channel 0 Command Block Pointer LSB+2	95H
96H	Channel 0 Command Block Pointer LSB+1	Channel 0 Command Block Pointer LSB	97H
98H	Channel 1 Command Block Pointer		
9EH	Channel 2 Command Block Pointer		
A4H	Channel 3 Command Block Pointer		
AAH	Channel 4 Command Block Pointer		
B0H	Channel 5 Command Block Pointer		
B6H	Channel 6 Command Block Pointer		
BCH	Channel 7 Command Block Pointer		

NOTE

All Command Block Pointers have the format demonstrated by Channel Zero.

Figure 2-6. Command Block Pointers

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.3.3 Command Block and Data Buffer Area

The remaining I/O Interface Block (byte) locations on the ICM (C2H through 27EH) are available to be used to store Command Blocks and Data Buffers. The Command Blocks written to the ICM, per IPC protocol, must reside in the module's Dual Access RAM (locations base + C2H through locations base + 27EH), and cannot be located in Global memory. This is because the ICM is a "slave" module and thus it cannot acquire the Data Transfer Bus in order to access global memory.

2.4 UNDEFINED AND RESERVED I/O INTERFACE BLOCK LOCATIONS

In addition to the reserved locations already pointed out in the memory map segments earlier in this chapter, it should be noted that the byte at location 27F is a reserved hardware location and must not be written to by user programs.

NOTE

Reading from or writing to undefined addresses may make the user software incompatible with future versions of this module.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 3

INTERPROCESSOR COMMUNICATION PROTOCOL

3.1 INTRODUCTION

XYCOM's Interprocessor Communication (IPC) protocol is a method of communication between a VMEbus host processor module and any intelligent XVME I/O module residing on the VMEbus. The IPC protocol uses "Command Blocks" to transfer commands and data between the host and the XVME I/O module.

A Command Block is a twenty-byte block of memory accessible to both the host and the I/O module. This could be either global RAM, or the 1K of Short I/O Address Space allotted to each XYCOM I/O module.

A Command Block contains the command word along with any additional parameters needed to execute the command. For example, a parameter could specify the interrupt level and vector to be generated by the I/O module upon completion of the command. The I/O module writes any returned data and error/completion codes to the same Command Block.

In addition to Command Blocks, the IPC protocol defines the 1K of Short I/O Address Space allotted to each I/O module. This 1K block, accessible to any VMEbus master, is called the I/O Interface Block.

A "master" I/O module can read Command Blocks in either global system memory or in its own I/O Interface Block. When communicating with a "slave" I/O module, Command Blocks must be written to the module's I/O Interface Block.

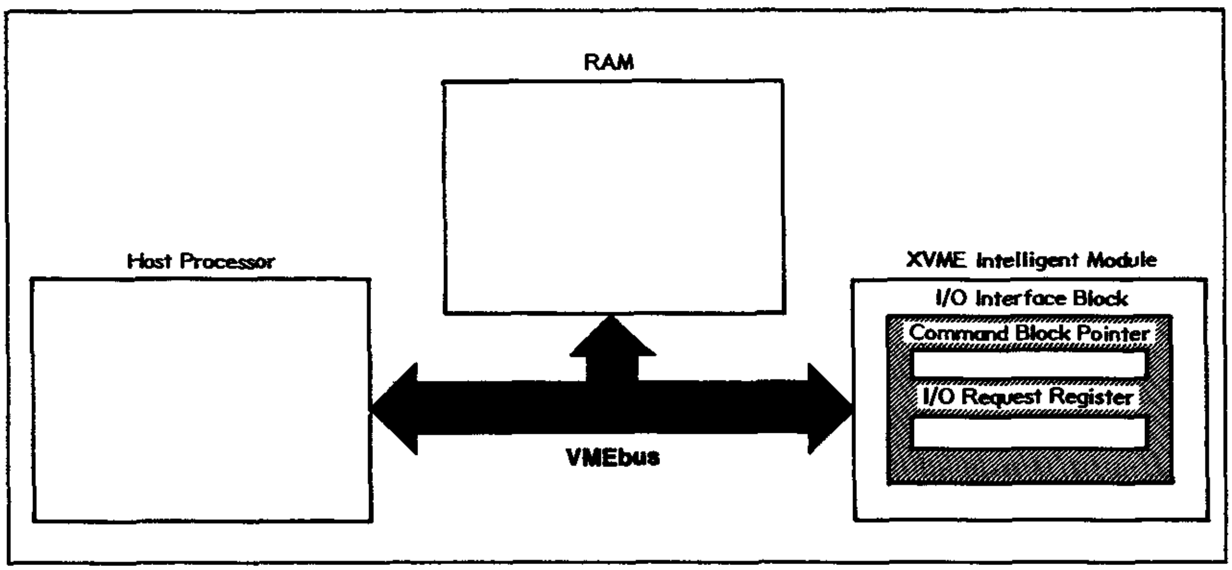


Figure 3-1. IPC Communication Elements

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The IPC protocol defines several specific addresses within the I/O Interface Block called "Command Block Pointers" and "I/O Request Registers" (see Figure 3-1). After creating a Command Block, the host must inform the I/O module of the location of the Command Block by writing the Command Block's address into a Command Block Pointer. To actually initiate execution of the command, the host must write into an I/O Request Register.

The I/O Interface Block has several Command Block Pointer-I/O Request Register pairs (the exact number varies from module to module). Each one of these pairs represents a logical I/O channel.

The IPC protocol defines the format of a Command Block, the I/O Interface Block, and the sequence in which the host and I/O module must exchange information over an I/O channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2 IPC OVERVIEW

This section presents a detailed overview of events that occur when a host module communicates with an I/O module using the IPC protocol.

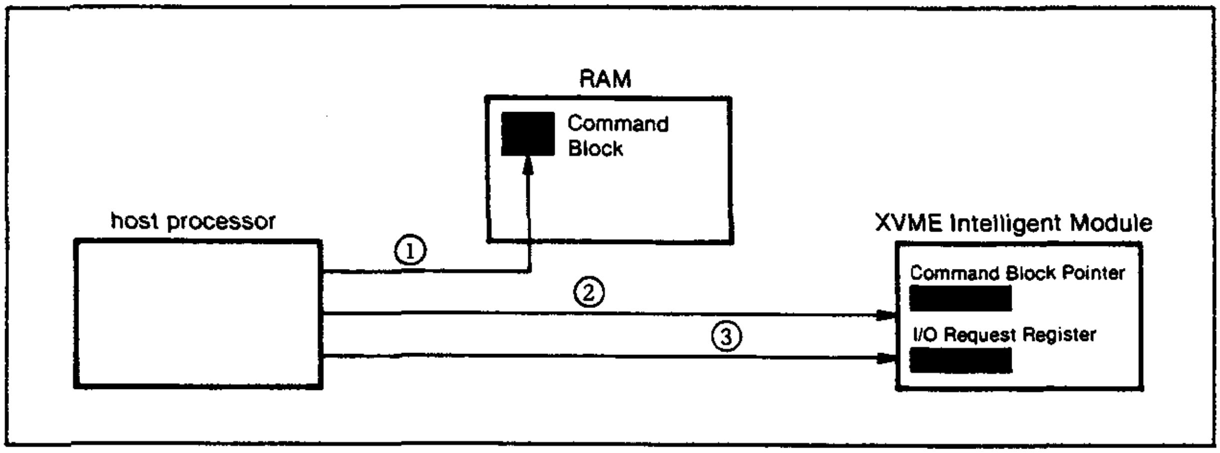


Figure 3-2. Host to I/O Module Communication

1) The host processor writes a command block anywhere in global memory or the I/O module's I/O Interface Block, see Figure 3-2.

2) The host processor sends the I/O module the address of the Command Block by writing this address in one of the Command Block Pointer fields of the I/O Interface Block, see Figure 3-2.

3) The host processor notifies the I/O module that a command block is available by writing 01H ("read the command block pointer") to an I/O request register in the I/O Interface Block, see Figure 3-2.

Step 2 may precede step 1, as long as step 3 occurs last. The Command Block must be created and its Command Block Pointer written to the I/O module before command execution can be initiated.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

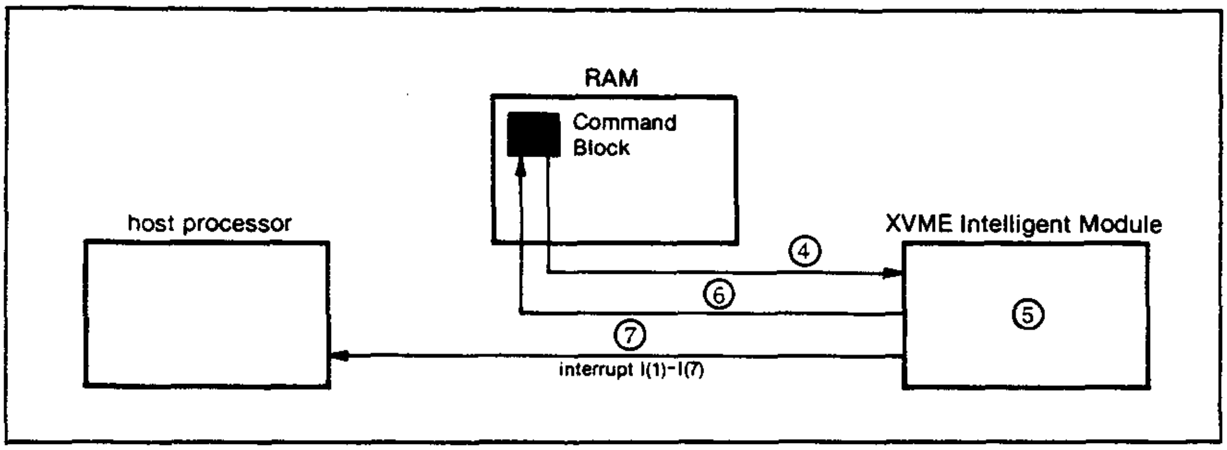


Figure 3-3. I/O Module to Host Communication

The following four steps are performed by the Intelligent I/O module without intervention from the host:

4) The I/O module reads the Command Block. If an I/O request acknowledge interrupt (see Section 3.7) has been specified to be generated, the I/O module will generate the proper interrupt level and vector in acknowledgement of the command, see Figure 3-3.

5) The I/O module executes the command, see Figure 3-3.

6) The I/O module writes results and data (or pointer to the data) back to the command block, see Figure 3-3.

7) A command completed interrupt will then be generated, at a specified interrupt level and vector, if so requested by the Command Block, see Figure 3-3.

3.3 I/O INTERFACE BLOCK

The I/O Interface Block is the 1K block of short I/O memory occupied by the I/O module. This Interface Block is accessible to any other master located on the VMEbus.

The I/O Interface Block contains all of the registers which the host uses when communicating with the I/O module via IPC protocol. For example, after the host writes a Command Block, it must write the address of the Command Block in a Command Block Pointer, and then initiate the command by writing to an I/O Request Register. Command Block Pointers and I/O Request Registers are all addresses located within an I/O Interface Block. In addition, the Command Block itself may be located in certain areas of the I/O Interface Block.

Figure 3-4 presents a general address map of the portion of the I/O Interface Block used by the IPC Protocol.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

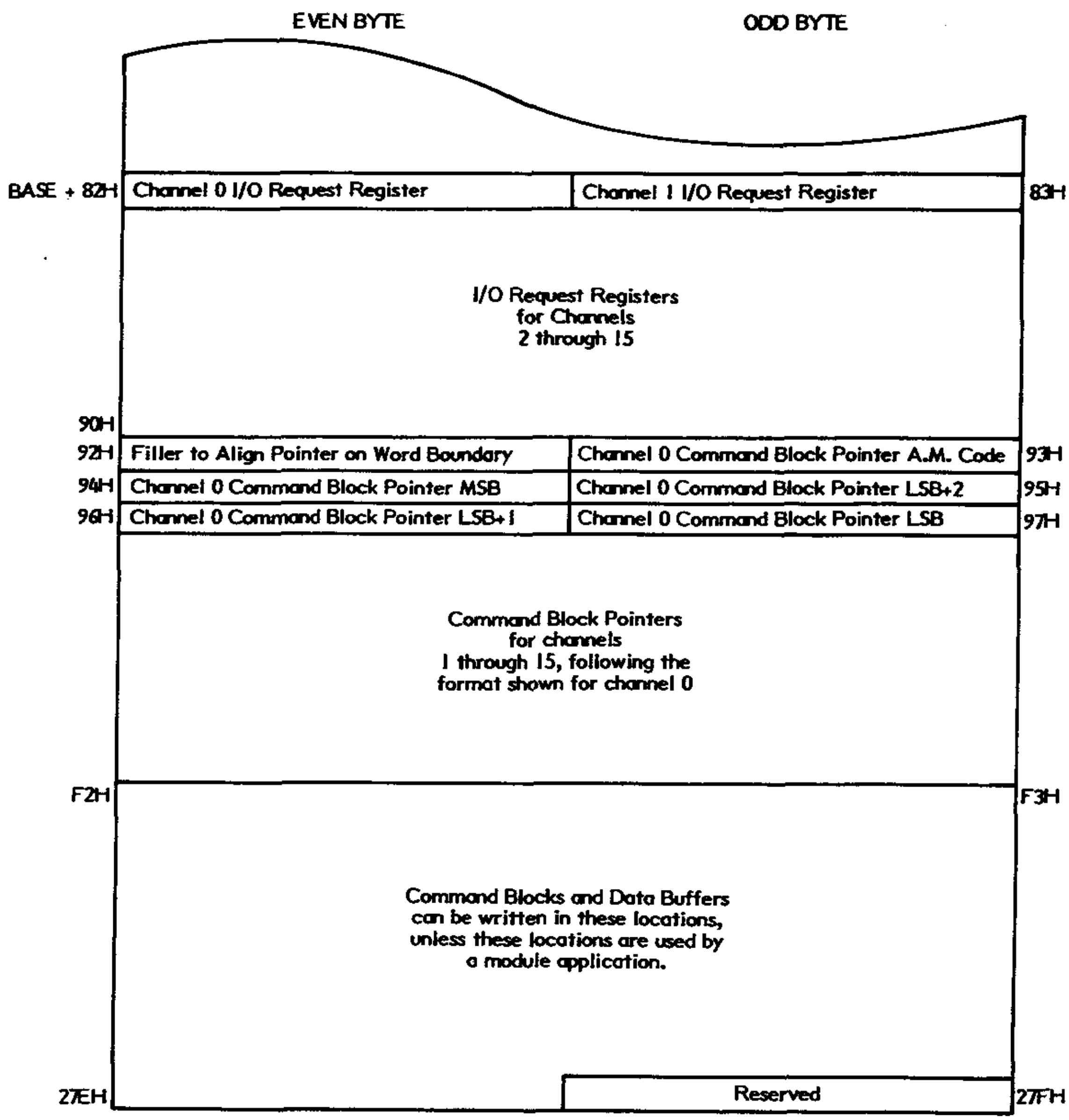


Figure 3-4. Portion of the I/O Interface Block Used by the IPC Protocol

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4 I/O CHANNELS

When viewed logically, each intelligent I/O module consists of several I/O channels. The destination of a particular Command Block is not the I/O module in general, but a specific I/O channel on that module. A feature of these I/O channels is that each one can execute a different Command Block at the same time. This allows one I/O module to execute several commands concurrently.

An I/O channel physically consists of one Command Block Pointer and one I/O Request Register. Together these two memory locations form one logical I/O Channel. For example, I/O Channel 0 consists of I/O Request Register 0 and Command Block Pointer 0, I/O Channel 1 consists of I/O Request Register 1 and Command Block Pointer 1, and so on. Since an intelligent I/O module may have up to 16 I/O Request Registers and 16 Command Block Pointers, an I/O module can have up to 16 I/O channels. (Not all I/O modules will use the full complement of 16 channels.)

After a host creates a Command Block, it can select any available I/O Channel to execute that command. The choice of an I/O channel is made by the host when it writes to that channel's Command Block Pointer. Execution of the command does not begin until the host writes 01H to the channel's I/O Request Register. Then the channel reads the Command Block Pointer to obtain the address of the Command Block, and initiates execution of this command.

For example, to select I/O Channel 2 to execute a particular Command Block, the host will:

1. Write the address of the Command Block to Command Block Pointer 2.
2. Initiate the command by writing 01H to I/O Request Register 2.

If instead the host wrote the address of that Command Block to Command Block Register 0 and activated the command by writing to I/O Request Register 0, it would be selecting I/O Channel 0 to execute the command.

I/O Channel Options

The I/O channels can execute different Command Blocks concurrently. Each channel may be used by a different host module, or one host may use several I/O channels. A single channel may also execute several commands concurrently. Note, however, that some I/O channels on a particular module may be dedicated to specific functions, so possibly not all of the I/O channels will execute the commands available on that module.

3.5 COMMAND BLOCKS

All commands sent to an intelligent I/O module, along with the parameters required to execute the command, must be written by the host in a Command Block format. A Command Block can be written in either of these two areas:

- Global memory, i.e., VMEbus address space accessible to both the I/O module and any host module.
- Certain areas within the I/O module's I/O Interface Block (see Figure 3-5).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

An I/O module acting as a bus "master" can access command blocks in either area. An I/O module acting as a bus "slave" can only access command blocks in its own I/O interface block.

If a Command Block (and its associated data buffer, if any) can be written to an intelligent module's dual-access RAM, no global memory is required to operate the module.

When writing the Command Block and/or Data Buffer, follow these rules:

1) All Command Blocks and Data Buffers written to "slave" modules must be completely contained within the I/O Interface Block.

2) The Command Block and Data Buffer must begin on even address boundaries. If the Command Block Pointer begins on an odd boundary, the I/O Request will be acknowledged, but the Command Block will be ignored.

3) The Command Block and Data Buffer cannot straddle 800000H in VME memory.

The IPC Command Block has seven parameter fields used to specify the following information (see Figure 3-5):

	EVEN	ODD	
byte 0	Command Word		byte 1
2	Response Word		3
4	"Command Completed" interrupt level	"Command Completed" interrupt vector	5
6	Response Flag	Next Command A/M Code	7
8	Pointer to Next Command Block		9
12	Operand Field (Command parameters, Data, or Data Pointer)		13
18			19

Figure 3-5. Command Block Format

The seven fields of the Command Block are defined as follows:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.1 Command Word

The Command Word field identifies the command that is to be processed by the intelligent I/O module. The I/O module never writes to this byte and does not return any data in this field.

3.5.2 Response Word

The I/O module writes a value in this field to indicate whether or not it was able to executed the command successfully. A value of 0 indicates that the command was successfully completed and no problems were detected. A non-zero value indicates an error code. Each type of I/O module has its own set of error codes.

3.5.3 Command Completion Interrupt Level and Vector

The value contained in this field is the VMEbus interrupt level and vector that will be generated by the I/O module when it has completed the command. This signifies to the host that the Response Word, Response Flag, and return data are available to be read. If the interrupt level is zero, then no interrupt will be generated. The I/O module does not write to this field.

3.5.4 Response Flag

The I/O module writes a zero in this field when the command has been completed. This field does not indicate whether the command was successful. To determine that, the host must read the Response Word field.

The primary use for the Response Flag is to allow a host polling the I/O module to determine whether a given command has finished execution.

When writing a Command Block, the host should write a non-zero value here because the I/O module only clears this field.

3.5.5 Next Command Address Modifier

If another Command Block is chained to this one, the value contained in this field specifies the address space in which the next Command Block is located. FFH should be written in this field if this Command Block is not part of a chain of Command Blocks, or if this is the last Command Block in a chain.

An Address Modifier code of 2DH in this field signifies that the next Command Block resides in the I/O Interface Block. All other Address Modifier codes specify VME standard address space. The address of the chained Command Block is specified in the "Pointer to the Next Command Block" field (bytes 8 through 11).

The I/O module does not return any data in this field.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.6 Pointer to the Next Command Block

Command Block chaining enables several Command Blocks to automatically be executed sequentially by having each Command Block in the chain contain the address of the next Command Block to be executed. Thus, each command that is executed points to the next Command Block in the chain.

This field is used only if another Command Block is chained to this one. The I/O module determines if command block chaining is requested by examining the "Address Modifier" field, byte 7. If the Address Modifier field is not FFH, the Pointer to the Next Command Block field is regarded as the address of the next Command Block in the chain. If it is FFH, the Pointer to the Next Command Block field is ignored.

The I/O module does not write to this field.

3.5.7 Operand Field

The Operand field (bytes 12 through 19 of the Command Block) is the area where operands and/or data resulting from or pertaining to the command sent must be exchanged. This field can contain any of the following:

1) Nothing (no operands or data associated with the command sent).

2) The actual operands and/or data associated with the command sent.

3) A pointer to, and size of, a data buffer located in global memory or the I/O Interface Block. This data buffer contains the operands and/or data associated with the command sent.

The only instance in which the Operand field may contain nothing is when the command sent does not have any operands or data associated with it. In this case the I/O module will ignore the Operand field.

3.5.7.1 Operand Field (containing operands or data)

Most commands require that additional parameters and/or data, not just the command word, must be sent to or received from the I/O module. If this information is six bytes or less it may be contained inside the Operand field.

In this case the first byte of the Operand field (byte 12) is the number (1-6) of bytes. The format of the Operand field when containing actual operands or data is shown below.

Byte	Use
12	Data buffer size (1 to 6)
13	Not used (filler to align data bytes on a word boundary)
14-19	Contains command operand when written by the host. Contains return data after the I/O module executes the command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.7.2 Operand Field (containing a pointer to a data buffer)

When a command requires that operands and/or data in excess of six bytes must be sent or received, these operand or data bytes must be contained in a data buffer. The location and size of this data buffer must be specified in the Operand field. The data buffer must be located in global memory or the I/O Interface Block.

In this case the first byte of the Operand field (byte 12) is zero. This indicates to the I/O module the location and size of a data buffer will follow. The Operand field format is shown below.

Byte	Use
12	00
13	Address Modifier which specifies the address space in which the data buffer is located
14-17	Data buffer address: byte 14 = MSB byte 15 = LSB+2 byte 16 = LSB+1 byte 17 = LSB
18-19	Length of Data buffer (0-65535). MSB followed by LSB. Host must write max. length of buffer in this field. The I/O module returns the actual buffer size here when the command is completed.

3.6 SENDING MULTIPLE COMMANDS

3.6.1 Command Queues

It is not necessary for an I/O channel to complete execution of a command before another command is sent to the same I/O channel. In other words, while I/O Channel n is executing a command, a host can write the address of another Command Block in Command Block Pointer n and trigger this next command by writing 01H in I/O Request Register n.

Commands sent to an I/O channel while the channel is still executing a previous command are stored automatically in a Pending Command Queue within the I/O module. This queue contains the address of the next Command Block to be executed. In order for a command to be placed in the queue, it must be triggered by writing 01H to the channel's I/O Request Register. It is not sufficient to just write the Command Block address in the channel's Command Block Register.

Each I/O channel has its own Pending Command Queue. Commands are placed in the queue in the order in which they are sent to the I/O channel. The I/O Channel ensures that all chained commands are executed in sequence by treating a command chain as a single entry in the Pending Command Queue (see Section 3.6.2 for a discussion of command chains).

3.6.2 Command Block Chaining

The IPC protocol provides the user with the ability to send Command Blocks to an I/O module in a "chained" format using the Pointer to Next Command Block field as described in the Command Block description in Section 3.3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chaining Command Blocks together allows lengthy command procedures (i.e., those which require sending many Command Blocks), to be linked together and sent to the I/O module as a group. This results in a substantial savings of host processor overhead because the host processor no longer has to wait for the I/O module to acknowledge the receipt of the Command Block Pointer before each subsequent Command Block Pointer can be written.

Command Block chaining works in the following fashion:

1. The "Pointer to Next Command Block Field" in the first Command Block contains the starting address of the second Command Block in the sequence.

2. The second Command Block is constructed.

3. Both Command Blocks are written to their respective places in memory.

4. The pointer to the first Command Block is written to the Command Block Pointer field of a particular I/O channel.

5. The host then writes a 01H to the I/O Request Register of the specified channel, telling it to get the Command Block pointer of the first Command Block. The I/O module then acknowledges the I/O request, stating that it has received the Command Block pointer.

6. The I/O module executes the first command, or puts it in the Pending Command queue.

7. When the first command has been completed, the I/O channel will detect that it contains a pointer to another Command Block, and it will place this pointer in its Pending Command Queue.

8. After the first command has been completely executed, the I/O channel executes the second command. Note that no I/O Request Acknowledge Interrupt is sent back for the second command.

A Command Completed interrupt is generated for each Command Block that requests an interrupt. To have the I/O module interrupt the host when an entire command chain completes, simply set the interrupt level and vector fields in all Command Blocks, except the last, to zeroes, and set the desired "Command Chain Complete" interrupt level and vector in the last Command Block in the chain.

The above procedure will allow any number of command blocks, limited only by the amount of memory they consume, to be chained.

3.6.3 Using the Same Command Block Address

Each Command Block Pointer location stores the last address written to it. Therefore, if successive Command Blocks which have the same Command Block Address are sent to an I/O channel, that address does not have to be rewritten to the Command Block Pointer. (In other words, step 2 in the procedure outlined in Section 3.1 can be omitted.) For example, suppose that I/O Channel 0 has just executed a Command Block located at address 1A2470 (hex). If a host then uses the same 20-byte block

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

beginning at address 1A2470 to write the next Command Block, it does not need to write the address in channel zero's Command Block Pointer again.

3.7 INTERRUPTS

The IPC protocol allows the I/O module to generate two optional interrupts. The interrupts are:

1. An I/O Request Acknowledge Interrupt. This interrupt is generated after a byte is written to a Channel's I/O Request Register to initiate a command. It indicates to the host that another byte can now be written to the same I/O Request Register (to initiate the next command on this channel). Although provision is not made for this interrupt in the Command Block, each intelligent XVME module offers specific IPC commands which allow the user to enable I/O Request Acknowledge interrupts and select the interrupt level/vector.
2. A Command Completion Interrupt. This interrupt occurs just after a command has been completed. It indicates to a host module that a Response Word, Response Flag, and return data are available to be read from the Command Block and Data Buffer. The Command Block contains a field (bytes 4-5) which determines whether or not a Command Completion Interrupt will be generated and also the interrupt level/vector of the interrupt (see Section 3.5.3).

3.8 IPC PROTOCOL EXAMPLES

This section provides detailed examples of how the IPC protocol can be used with the ICM module. The command selected for this example is the Move Motor command. (Chapter 11, Section 11.2.3 explains Stepper Motor Control on the ICM.) Since the IPC protocol can be used in either polled or interrupt-driven operation, two Move Stepper Motor examples are included, one for each of these two modes of operation.

Polled Operation

In this example, the ICM is programmed for polled operation. In other words, no interrupts will be generated when the Move Motor command is completed. Instead, the host module must poll the Response Flag of the Command Block to determine when the command is completed (i.e., the stepper motor has moved the specific number of steps). The command operands are stored in a Data Buffer. However, since the Operand Field is no more than six bytes long, it may be stored in the Command Block itself (see the last example).

Assumptions

For the purpose of this example, the following assumptions have been made about the base address and the address of the Command Block and Data Buffer.

- The host module performs memory-mapped I/O, and the base of its Short I/O Address space (referred to as base0) is F90000H.
- The jumper-selectable base address of the ICM's I/O Interface Block (referred to as base1) is 1000H.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- The Command Block will be located in the ICM's I/O Interface Block at the following address:

  base0 + base1 + 200H = F91200H
- The Data Buffer which will hold the operands required for the Move Motor command is located at the following address:

  base0 + base1 + 0214H = F91214H

  Note that the Data Buffer is located immediately after the 20-byte Command Block though this is not necessary.

- The size of the Data Buffer is six bytes.

The record will be read from I/O Channel 0.

Procedure to Move a Stepper Motor Under Polled Operation

1. The host must write the following Command Block at address F91200H:

Address	Even Byte	Odd Byte	
F91200H	00	42	Command Word for Move Motor command
F91202H	XX (see 1)	XX	Response Word
F91204H	00	00	No interrupts
F91206H	FFH (see 2)	FFH	No Command Block chaining
F91208H	XX	XX	
F9120AH	XX	XX	
F9120CH	00	2DH	AM code for ICM I/O Interface Block (Command/Data Buffer area)
F9120EH	00	F9H	Address of Data Buffer
F91210H	12H	14H	
F91212H	00	06H	Number of bytes in Data Buffer

NOTES

1. XX = don't care
2. The host must write a non-zero value in the Response Flag. The ICM will set the Response Flag to zero when the command is completed.

Figure 3-6. Sample Move Motor Command Block (Polled Operation)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. The host must write the address of the Command Block (F91200H) in the Channel 0 Command Block Pointer.

Address	even byte	odd byte	
F91092H	XX	2DH	AM code for ICM I/O Interface Block
F91094H	00	F9H	Address of
F91096H	12	00	Command Block

3. The host must write 01H to the Channel 0 I/O Request Register (address = base0 + base1 + 82H) to initiate the command.

Address of Ch. 0 I/O Request Register	Value
F91082H	01H

4. The host must poll the Response Flag of the Command Block until a 0 value is detected, which indicates that the stepper motor has completed its movement. Since the Response Flag is byte 6 (the 7th byte) of the Command Block, which begins at F91200H, the host must read location F91206H until a 0 value is read.

   Note that this type of polling can tie up much of the VMEbus' time if there is not sufficient pause between successive reads of the Response Flag.

Interrupt Operation

In this example, the ICM is programmed to generate an interrupt when the Move Motor command is completed (i.e., when the stepper motor has completed its specified movement).

This example is similar to the polled operation example above. Since the Operand Field is no more than six bytes in size, the user has the option of storing it in a Data Buffer (as shown in the previous example) or in the Command Block (as shown in this example).

Note that the option of storing a six-or-fewer byte Operand Field in the Data Buffer or Command Block is independent of whether the I/O channel is operated in polled or interrupt mode. Since the Operand Field is six or fewer bytes long, the polled example above could also be modified to store the Operand Field in the Command Block.

Assumptions

All the assumptions about addresses and I/O channel number made in the polled example also apply to the following interrupt example. In addition, the following assumptions are made:

- o The Command Completed interrupt will be a level 1 interrupt with interrupt vector 80H.
- o The command operands are placed within the Command Block.

Procedure to Move a Stepper Motor Under Interrupt Operation

1. The host must write the following Command Block at address F91200H.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Address	Even Byte	Odd Byte	
F91200H	00	42H	Command Word
F91202H	XX	XX	Response Word
F91204H	01H	80H	Interrupt level 1 & vector 80H
F91206H	XX	FFH	No Command Block chaining
F91208H	XX	XX	
F9120AH	XX	XX	
F9120CH	06	XX	Even byte specifies number of operand bytes
F9120EH	01	XX	
F91210H	12H	14H	Operand Field
F91212H	42H	75H	

XX = don't care

Figure 3-7. Sample Move Motor Command Block (Interrupt-Driven Operation)

2. The host must write the address of the Command Block (F91200H) in the Channel 0 Command Block Pointer.

Address	even byte	odd byte	
F91092H	XX	2DH	AM code for ICM I/O Interface Block
F91094H	00	F9H	Address of
F91096H	12	00	Command Block

3. The host must write 01H to the Channel 0 I/O Request Register to initiate the command.

Address of Ch. 0 I/O Request Register	Value
F91082H	01H

4. When the stepper motor has moved the specified number of steps, the ICM will generate a level 1 interrupt with interrupt vector 80H. The host's interrupt service routine can then take the appropriate action.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE

An additional interrupt, the I/O Request Acknowledge interrupt, can also be generated by the ICM. The Set I/O Request Interrupt Level and Vector command can configure any I/O channel for I/O Request Acknowledge interrupts. If a channel is so configured, the ICM also generates an interrupt shortly after a value is written to the channel's I/O Request Register. This interrupt indicates to the host that the channel is ready to accept another command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 4

USING THE ICM

4.1 INTRODUCTION

This chapter presents the information required to send commands via the IPC protocol in order to initiate and control the ICM's various functions. The information is discussed in the following order:

- Counter Architecture
- How the functions use the Counter Architecture
- ICM I/O channels
- ICM Command format in relation to the IPC Command Block format
- Sending a typical command
- Listing of general commands
- Command queues
- Interrupts
- Error codes generated by the ICM
- Floating point format used in ICM functions

Commands that are specific to a measurement and control function are listed in the chapter describing that function.

4.2 COUNTER ARCHITECTURE

The ICM has 16 counters that perform its measurement and control functions. They are divided into groups of four counters apiece. Each group of four counters is called a Function Block. The Function Blocks are referred to as blocks A, B, C and D. Each Function Block is divided into two channels of two counters each.

Each of the 16 counters is referenced according to the Function Block it belongs to, in the following manner: A0, A1, A2, A3, B0, B1, B2, B3, C0, C1, C2, C3, D0, D1, D2, and D3.

Each of the 16 counters has the following three signals that are available on connectors JK1 and JK2:

- CLOCK: this signal provides the measurement or control function input.
- OUTPUT: this signal provides the measurement or control function output.
- GATE: this is used as an enabling signal or additional measurement or control function input.

These signals are prefixed by the letter of the Function Block they belong to and suffixed by the number of the individual counter within the Function Block. For example, the signals of Counter A0 would be referenced on connector JK1 as:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ACLOCK0
AGATE0
AOUT0

All three of these signals are not necessarily needed by every function. The user should refer to the chapter where the function is described for complete function configuration information.

4.3 HOW MEASUREMENT AND CONTROL FUNCTIONS USE THE COUNTER ARCHITECTURE

The ICM's 16 counters are accessed by sending commands to the ICM's eight I/O channels. The channel/counter logical relationship is shown below:

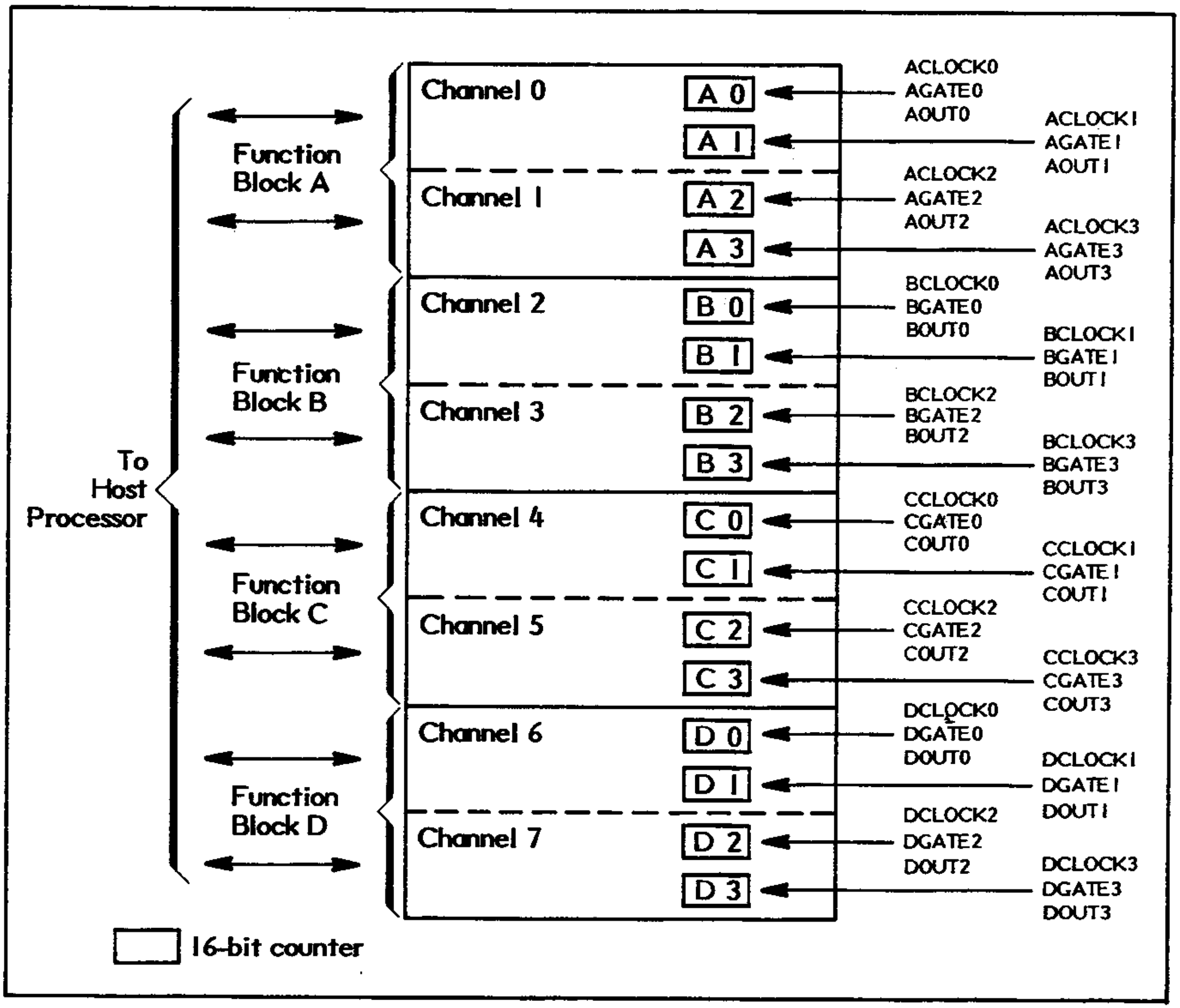


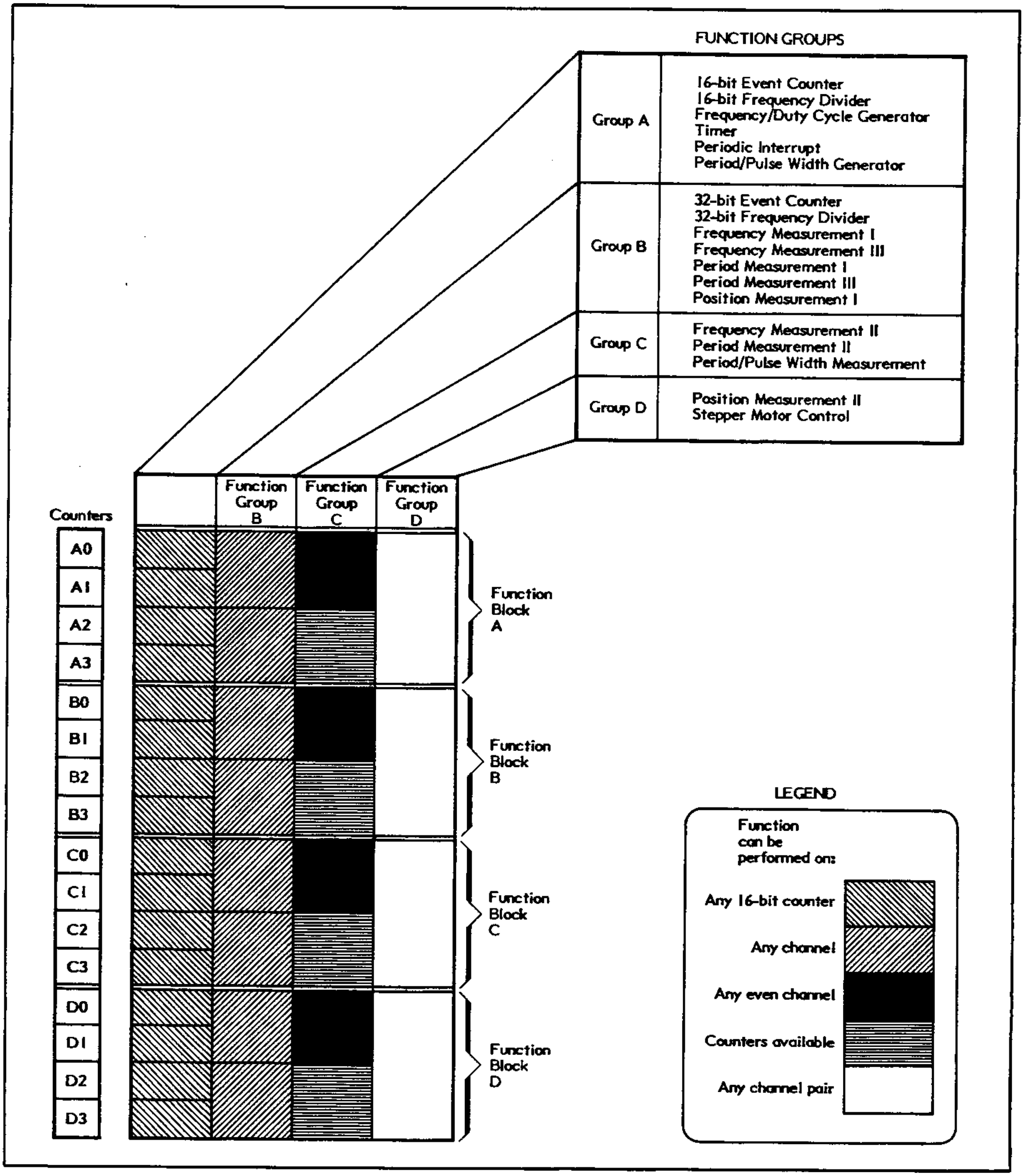
Figure 4-1. Logical Architecture

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When the user sends a command initializing a measurement and control function, the command will be processed and the counters required by the function will be configured automatically. The counters used to implement the function are chosen according to the I/O channel to which the command was sent and according to the complexity of the function. As can be seen from Figure 4-1, each Function Block is controlled by two channels. Table 4-1 shows the number of counters needed for each function compared to the total number of counters available, therefore determining the ICM's functional capacity.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-1. The Area of Counter Architecture Required for Operation of Each Function



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 ICM I/O CHANNELS

The ICM has eight I/O channels that can receive IPC Command Blocks (see Figure 4-2). The Command Blocks sent to the ICM must reside entirely in its I/O Interface Block and not in global memory, because the ICM is a slave module and cannot access global memory.

	EVEN BYTE	ODD BYTE	
Base + 00H 3EH	Undefined	Module Identification Data	01H 3FH
40H 7EH	Undefined		41H 7FH
		Status/Control	81H
82H	Ch. 0 I/O Request	Ch. 1 I/O Request	83H
84H	Ch. 2 I/O Request	Ch. 3 I/O Request	85H
86H	Ch. 4 I/O Request	Ch. 5 I/O Request	87H
88H	Ch. 6 I/O Request	Ch. 7 I/O Request	89H
8AH 91H	Unused		
92H	Channel 0 Command Block Pointer		
98H	Channel 1 Command Block Pointer		
9EH	Channel 2 Command Block Pointer		
A4H	Channel 3 Command Block Pointer		
AAH	Channel 4 Command Block Pointer		
B0H	Channel 5 Command Block Pointer		
B6H	Channel 6 Command Block Pointer		
BCH	Channel 7 Command Block Pointer		
C2H 27EH	Command Block and Data Buffer Area	Reserved	27FH
3FEH	Undefined		3FFH

Figure 4-2. ICM I/O Interface Block

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4.1 I/O Request Registers

The ICM has eight I/O request registers each of which occupies one byte of its I/O Interface Block and are assigned to the eight channels as follows:

82H	Channel 0 I/O Request Register	Channel 1 I/O Request Register	83H
84H	Channel 2 I/O Request Register	Channel 3 I/O Request Register	85H
86H	Channel 4 I/O Request Register	Channel 5 I/O Request Register	87H
88H	Channel 6 I/O Request Register	Channel 7 I/O Request Register	89H

Figure 4-3. ICM I/O Request Registers

4.4.2 Command Block Pointers

The ICM's Command Block Pointer locations each occupy six bytes of its I/O Interface Block.

The Command Block Pointers for each of the eight Channels have the following format:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

	EVEN	ODD	
92H	Word Alignment Byte	Command Block Pointer Address Modifier Code	93H
94H	Channel 0 Command Block Pointer MSB	Channel 0 Command Block Pointer LSB+2	95H
96H	Channel 0 Command Block Pointer LSB+1	Channel 0 Command Block Pointer LSB	97H
98H	Channel 1 Command Block Pointer		
9EH	Channel 2 Command Block Pointer		
A4H	Channel 3 Command Block Pointer		
AAH	Channel 4 Command Block Pointer		
B0H	Channel 5 Command Block Pointer		
B6H	Channel 6 Command Block Pointer		
BCH	Channel 7 Command Block Pointer		

NOTE

All Command Block Pointers have the format demonstrated by Channel Zero.

Figure 4-4. Command Block Pointers

4.5 ICM COMMAND FORMAT

The ICM commands are structured so their command format can be "packaged" in the IPC command block format (described in Chapter 3) and written to the ICM. An example of a command format is that of the Start Frequency/Duty Cycle Generation function listed below:

Start Frequency/Duty Cycle Generator - Command Format

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Word: 30H

10 operand bytes:

Byte 1: Counter number ———————— (0,1,2 or 3)

Byte 2: Data format selected ———— (0 = integer, 1 = IEEE single)

Bytes 3-6: Frequency ——————— (units of .01Hz, integer format
(units of 1Hz, IEEE single)

Bytes 7-10: % On ———————— (units of .01%, integer format)
(units of 1%, IEEE single)

To package a command format inside a Command Block, the appropriate command word is placed in the Command Block's command word field. For example, the host would write 30H for the Start Frequency/Duty Cycle function. Next, any parameter bytes required by the command are placed either in the operand field (if there are six bytes or less) or in an external data buffer. For example, the Start Frequency Duty Cycle command requires 10 parameter bytes, therefore these parameter bytes must be located in an external data buffer (see example in Section 4.6.4).

NOTE

The complete Command Block and data buffer <u>must</u> both reside in the "Command and Data Buffer Area" within the ICM's I/O Interface Block.

4.6 SENDING COMMANDS

Any one of the ICM's eight channels can be configured to perform a specific function (see Table 4-1 for limitations) by sending it a Command Block containing the command initializing the desired function. The function then begins, with a response generated upon completion.

What Happens When A Command Generates A Response

When a command terminates, it will generate a response by doing the following:

- Clear the Response Word (or deposit an error code).
- Clear the Response Flag.
- Optionally generate Command Completion interrupt.

Some commands return a response immediately. For example, the Frequency/Duty Cycle Generation function does this because its function, which is to produce the specified output signal, is completed immediately. Some commands return a delayed response because the command's function requires a significant amount of time to complete; for example, programming the stepper motor controller to move a stepper motor a certain number of steps. See the individual command listings for full explanations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6.1 Single Counter Functions

Some functions are implemented on one counter (half a channel). Commands to initialize these functions can be sent to any of the eight channels. These functions are:

- 16-bit Event Counting Functions
- Frequency/Duty Cycle Generation
- Timer
- 16-bit Frequency Divider
- Periodic Interrupt
- Period/Pulse Width Generation

4.6.2 Single Channel Functions

Commands to initialize the following lesser complex functions can be sent to any of the eight channels:

- Frequency Measurement I
- Period Measurement I
- 32-bit Event Counting functions
- 32-bit Frequency Divider
- Position Measurement I
- Frequency Measurement III
- Period Measurement III

4.6.3 Complex Functions

Commands to initialize the following complex functions must be sent to one of the even numbered channels. The following are the complex functions:

- Stepper Motor Control
- Position Measurement II
- Frequency Measurement II
- Period Measurement II
- Period/Pulse Width Measurement

The last three functions (Frequency Measurement II, Period Measurement II, and Period/Pulse Width Measurement) allow the corresponding odd channel to be used for any single channel function, whereas the first two functions (Stepper Motor Control and Position Measurement) require both the even and the odd corresponding channel to operate.

4.6.4 Programming Example

The following four sections present a programming example showing the sending of the Command Block and related parameters needed to configure counter A0 to perform the Frequency/Duty Cycle Generation function. A frequency of 100Hz and a duty cycle of 30% have been selected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The programming example shows the following sequence:

- construction of the Command Block
- construction of the data buffer containing the operand bytes
- construction of the Command Block Pointer
- generation of the I/O Request
- reading of the Response Flag to detect command completion

Each part of the message is written to the ICM's I/O Interface Block as it is constructed.

Assumptions

For the purpose of this example, the following assumptions have been made about the base address and the address of the Command Block and Data Buffer.

- The host module specifies the Short I/O Address Space as starting at location F90000H (referred to as Base One).

- The jumper-selectable Base Address of the ICM's I/O Interface Block is 1000H (referred to as Base Four).

- The Command Block will be located in the ICM I/O Interface Block at the following address:

  Base One + Base Four + C2H = F910C2H

- The Data Buffer holding the command's operand bytes is located at the following address:

  Base One + Base Four + D6H = F910D6H

Note that in this example the Data Buffer is located immediately after the 20-byte Command Block.

4.6.5 Command Block

The Command Block format is now filled in at address F910C2H (please refer to Section 3.5 for field definitions).

Command Word = 0030H, as specified by the Start Frequency/Duty Cycle Generation command.

Response Word = FFFFH, arbitrary value so a change in the response word is noticed.

Interrupt Level and Vector = 0000, for no interrupts.

Response Flag = FFH, arbitrary value so it is noticed if the I/O module sets this to zero.

Address Modifier = FFH, because Command Block chaining is not being used.

Next Command Address = 00000000, not used, value is irrelevant.

As mentioned, the Start Frequency/Duty Cycle command format has 10 parameter bytes (refer to Section 4.4 for complete command format). Therefore, the Operand Field must be configured to indicate the use of an external data buffer, which, in this example, is located at address Base + D6H.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

	Operand Field	
Byte 12: =	00H	because data buffer is being used.
Byte 13: =	2DH	address modifier qualifies Short I/O Non-Privileged access
Byte 14: =	00	MSB of external data buffer address
Byte 15: =	F9H	LSB+2
Byte 16: =	10H	LSB+1
Byte 17: =	D6H	LSB

Bytes 18-19: = 000AH for length of data buffer.

The final Command Block is shown below:

VMEbus Address	Even Byte	Odd Byte	
F910C2H	00H	30H	Command word
F910C4H	FFH	FFH	Response word
F910C6H	00H	00H	No interrupts
F910C8H	FFH	FFH	No Command Block chaining
F910CAH F910CCH	XXH XXH	XXH XXH	Next Command Block field not used
F910CEH	00H	2DH	A.M. code for ICM I/O Interface Block
F910D0H F910D2H	00H 10H	F9H D6H	Address of Data Buffer
F910D4H	00H	0AH	Length of Data Buffer

NOTE

XX = don't care

Figure 4-5. Start Frequency/Duty Cycle Command Block

4.6.5.1 Data Buffer

The format of the operand bytes required by the Start Frequency/Duty Cycle Generation command is shown in Section 4.5. The parameter bytes must be stored in an external data buffer as follows:

Byte 1: counter number (either 0,1,2 or 3)

In this example, zero is used to specify counter zero.

00

Byte 2: data format (0 for integer or 1 for floating point)

In this example, zero is used to specify integer data.

00

Bytes 3-6: specify the desired frequency in units of .001Hz

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In this example, knowing that 100Hz is desired, we must divide it by the units of .001Hz; 100Hz/.01Hz = 10000. This number is now converted to hexadecimal format for programming (10000 = 2710H):

00002710

Bytes 7-10: specify the desired duty cycle in units of .01%

In this example, if 30% is desired, multiply 30% by 100 to obtain 3000 (in units of percent). This number is now converted to hexadecimal format for programming (3000 = 0BB8H):

00000BB8

The resulting operand field as stored in the Data Buffer is shown below:

VMEbus Address	Even Byte	Odd Byte
F910D6H	00H	00H
F910D8H	00H	00H
F910DAH	27H	10H
F910DCH	00H	00H
F910DEH	0BH	B8H

which is 10 bytes long. In this case the data buffer is written into the I/O Interface Block at location base + D6H. This location has been specified in the Command Block's Operand Field.

4.6.5.2 Command Block Pointer

The Command Block Pointer is constructed and written to channel zero's Command Block Pointer:

VMEbus Address	Even Byte	Odd Byte	
F91092H	XX	2DH	A.M. code for ICM I/O Interface Block
F91094H	00H	F9H	Address of Command
F91096H	10H	C2H	Block

4.6.5.3 I/O Request

Now all the important information needed to initialize counter A0 as a Frequency/Duty Cycle controller has been written to channel zero in the ICM's I/O Interface Block. An I/O request (01H, which means "Read the Corresponding Command Block Pointer") is now written to channel zero's I/O request register at location base + 82H.

Address of Ch. 0 I/O Request Register	Value
F91082H	01H

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Upon receiving the I/O request, channel zero processes the command and configures counter A0 as a Frequency/Duty Cycle Controller.

The user should now be able to monitor the signal on pin AOUT0 on JK1 with an oscilloscope and see a pulse waveform with a frequency of 100Hz and a Duty Cycle of 30%.

4.6.5.4 Reading the Response Flag

The host can check to see if the command was completed and a response received by reading the Response Flag of the Command Block. A value of zero indicates command completion.

<u>Address of Response Flag</u>

F910C8H

4.7 COMMAND QUEUES

For each of the eight channels on the ICM there are two command queues to organize the processing of commands: the Pending Command queue and the Immediate Command queue.

4.7.1 Pending Command Queue

The Pending Command queue is a buffer where a command is stored, ready to execute pending completion of the current command. It can store up to four commands. All current commands, except for the ICM general commands described in Section 4.9, are placed in the Pending Command queue.

4.7.2 Immediate Command Queue

The Immediate Command Queue is a buffer that contains commands to be executed immediately. It can store up to four commands. All ICM general commands are placed in the Immediate Command queue.

4.8 INTERRUPTS

The functions running on the ICM have the capability to interrupt the host processor when they are finished as well as when they are just acknowledging an I/O request.

In either case, the user can specify the interrupt level and vector the ICM uses when generating the interrupt.

4.8.1 Command Completion Interrupts

Command completion interrupts have the interrupt acknowledge vector specified by the host in bytes 4 and 5 of the Command Block labeled the Completion Interrupt level and vector field. After receiving an interrupt from the ICM, the host must read the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

response word of the Command Block to determine whether the command was successfully completed (see Section 4.10 for a description of error codes).

4.8.2 I/O Request Acknowledge Interrupt

The following command will affect the interrupt handling:

- Set I/O Request Interrupt Level and Vector

This command specifies the Acknowledge Interrupt level and vector the host processor will receive after writing an I/O request.

4.8.3 SYSFAIL

The ICM handling of the VME SYSFAIL interrupt is determined by the SYSFAIL Option command. The default condition is the shutdown option. Upon receipt of the SYSFAIL interrupt, all counters are turned off and all outputs are driven to their low state. The other option is to ignore the SYSFAIL interrupt and rely on the host to issue commands in the event of a SYSFAIL.

4.8.4 ACFAIL

When the ICM receives an ACFAIL interrupt, it shuts itself down with the following procedure:

1) Turns off all counters.
2) Drives all outputs to their low state.
3) Executes a Stop command.

Only a system reset can restart the ICM after this shutdown procedure occurs.

4.9 ICM GENERAL COMMANDS

The commands pertaining to all functions (also labeled "immediate" commands) will be defined in this section. The commands for a specific function will be defined in the chapter describing that function. Table 4-2 provides a list of the ICM's general purpose commands:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-2. ICM General Commands

Command Word	Command
10H	Board Reset
11H	Run Diagnostics
12H	Set Sysfail Option
18H	Stop
19H	Set I/O Request Interrupt Level and Vector

All ICM command blocks follow the IPC protocol as defined in Chapter 3, Section 3.5. In the following command format listings, only the command word and the operand bytes are shown.

4.9.1 Module Reset Command

The Module Reset Command automatically resets the ICM, clears all functions presently running, and invokes the power-up diagnostics. This command will return a response immediately. The format is listed below:

Command Word: 10H

(This command does not require any operands.)

4.9.2 Run Diagnostics Command

The Run Diagnostics Command runs the diagnostics routine normally performed upon module power-up and returns an acknowledge response immediately. The command format is listed below:

Command Word: 11H

(This command does not require any operands.)

4.9.3 Set SYSFAIL Option

The Set SYSFAIL Option command allows the user to specify how the SYSFAIL interrupt will be handled. It returns an acknowledge response immediately. Please refer to Section 4.9.5 for a description of options. The command format is listed below:

Command Word: 12H

Operand bytes

Byte 1: Option flag ——— 0 means shutdown upon receiving SYSFAIL
1 means ignore SYSFAIL

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.9.4 Stop Command

The Stop Command stops the function presently running on the channel that it is written to. If the channel is running two functions, the individual counter 0, 1, 2 or 3 may be specified. The format is listed below.

Command Word: 18H

Operand bytes

Byte 1: Optional Counter number (FFH if no counter specified)

4.9.5 Set I/O Request Interrupt Level and Vector

The Set I/O Request Interrupt Level and Vector sets the interrupt level and vector the ICM uses to acknowledge an I/O request. The format is listed below:

Command Word: 19H

Operand bytes

Byte 1: Interrupt Level (1-7)
Byte 2: Interrupt Vector (2-255)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.10 ERROR CODES

The following conditions will cause the corresponding error return code to be returned to the host processor in the Response Word of the Command Block:

Error Code	Error Condition
01H	Illegal address
02H	Illegal command
03H	Illegal counter
04H	Configuration conflict
05H	Illegal direction
06H	Download error
07H	Illegal duty cycle
08H	Floating point error
09H	Illegal frequency
0AH	High limit exceeded
0BH	Counting limit exceeded
0CH	Low limit exceeded
0DH	Illegal interrupt level or vector
0EH	Motor already moving
0FH	Pending Command Queue full
10H	Odd block specified where even block required
11H	Counter Overflow
12H	Illegal period
13H	Illegal stepping rate
14H	Illegal period-units combination
15H	Illegal number of steps
16H	Illegal time
17H	Illegal units
18H	Illegal Frequency Divisor
19H	Numerical Overflow
1AH	Illegal number of intervals
1BH	Counter test failure
1CH	ROM test failure
1DH	Illegal pulse width

4.11 FLOATING POINT FORMAT

The ICM can return the calculated results of functions such as the Period Measurement and Frequency Measurement functions, in the IEEE single precision standard floating point format which is described in this section.

The IEEE single precision floating point format is a 32 bit (0 to 31) representation of a floating point value divided up into sections as shown in the figure below.

31	30 23	22 0
s	exponent (e)	significand (f)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The value of the "s" bit (bit 31) indicates the sign of the number:

0 = positive
1 = negative

The value of the eight bit exponent "e" (bits 23 thru 30) is an unsigned integer with a bias of 127 (i.e., e-127). The exponent has a range of greater than zero and less than 255 ($0<e<255$).

The value of the significand "f" (bits 0 thru 22) is a "normalized" number that lies in a range greater than or equal to one, or less than two ($1.0 \leq f < 2.0$). Since the significand is normalized, there is an implied one (1) to the left of the binary decimal point.

For example, the number 10.00 is expressed in the IEEE single floating point format as follows:

10.00 (decimal) will be expressed as 1.01 (in binary) x 2^3 = 10.00

$e = 127 + 3 = 130 = 82H$

$f = 1.01 - 1.00 = .01$ (due to the implied one to the left of bit 22)

$s = 0$

The binary representation is:

31	30 23	22 0
0	1 0 0 0 0 0 1 0	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

4.12 FLOATING POINT OPERANDS

Some commands have operands which may be specified in either a floating point or integer data format.

An operand expressed in floating point format requires four bytes (as explained in Section 4.11). For example, if the decimal number 20 (41A00000 in floating point) is to be sent as a floating point number to address FF00D8, the operand will look as shown below:

Address	Data
FF00D8	41A0
FF00DA	0000

An operand expressed in integer format is limited to the size of 16-bits which appears in the higher addressed word of the four-byte operand. For example, if the decimal number 20 (0014 in hexadecimal integer) is sent as an integer to address FF00D8, the operand will look as shown below:

Address	Data
FF00D8	0000
FF00DA	0014

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.13 OPTIONAL LIMIT CHECKING

Limit Checking is a user-specified option, designed to operate with the Event Counting, Period Measurement, Period/Pulse Width Measurement, and Frequency Measurement functions. When used with the Event Counting function, Limit Checking allows the user to specify an upper counting limit at which the counting function will terminate and a response word (0BH - Counting limit exceeded) will be issued. If Limit Checking is not used with Event Counting, the event counter will initialize counting and continue counting until the counter overflows (refer to Chapter 6 for information on the initialization and operation of the Event Counting functions).

When used with the Period Measurement, Period/Pulse Width Measurement, and Frequency Measurement functions, the Limit Checking feature allows the user to specify an upper and lower limit (i.e., a range) within which the input measurement functions will operate successfully. However, if the signal being measured should go outside of the specified limits (i.e., the input period, pulse width, or frequency goes either too high or too low), the measurement function will terminate and one of two response words will be issued (0AH - High limit exceeded, or 0CH - Low limit exceeded). Once a specified limit has been exceeded (either a high limit or a low limit), the pertinant measurement function start command must be reissued in order to again begin the measurement function.

In the case of Period/Pulse Width measurement, the Limit Checking option only applies to the measurement of the input signal pulse width. Limit Checking does not apply to the period measurement in the Period/Pulse Width measurement function.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 5

INSTALLATION

5.1 INTRODUCTION

This chapter explains how to configure the ICM prior to installation in a VMEbus backplane. It then explains the connection of external signals, and outlines the physical installation procedure.

5.2 SYSTEM REQUIREMENTS

The ICM is a double height VMEbus compatible module. To operate it must be properly installed in a VMEbus backplane cardcage.

The minimum system requirements for operation of the ICM are one of the following:

A)
- A host processor properly installed on the same backplane.
- A properly installed controller subsystem with the following attributes:

  -- Data Transfer Bus ARBITER
  -- System Clock driver
  -- System Reset driver
  -- Bus time-out module

-- OR --

B)
- A host processor which incorporates an on-board controller subsystem.

An example of such a controller subsystem is the XYCOM XVME-010 System Resource Module (SRM).

Prior to installing the ICM, it will be necessary to configure several jumper-selectable options. These options are:

1) Module Base Address within short I/O memory.
2) Address Modifier codes to which the ICM will respond.
3) Memory Range Selection. This option is used only if on-board memory devices are modified.
4) Memory Access Time Control. This option is used only if on-board memory devices are modified.

The ICM's two connectors, JK1 and JK2, require two 50-pin ribbon cables for interfacing the control functions to their external tasks.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.3 JUMPER LOCATIONS

The jumpers relevant to installation are shown in Figure 5-1 below.

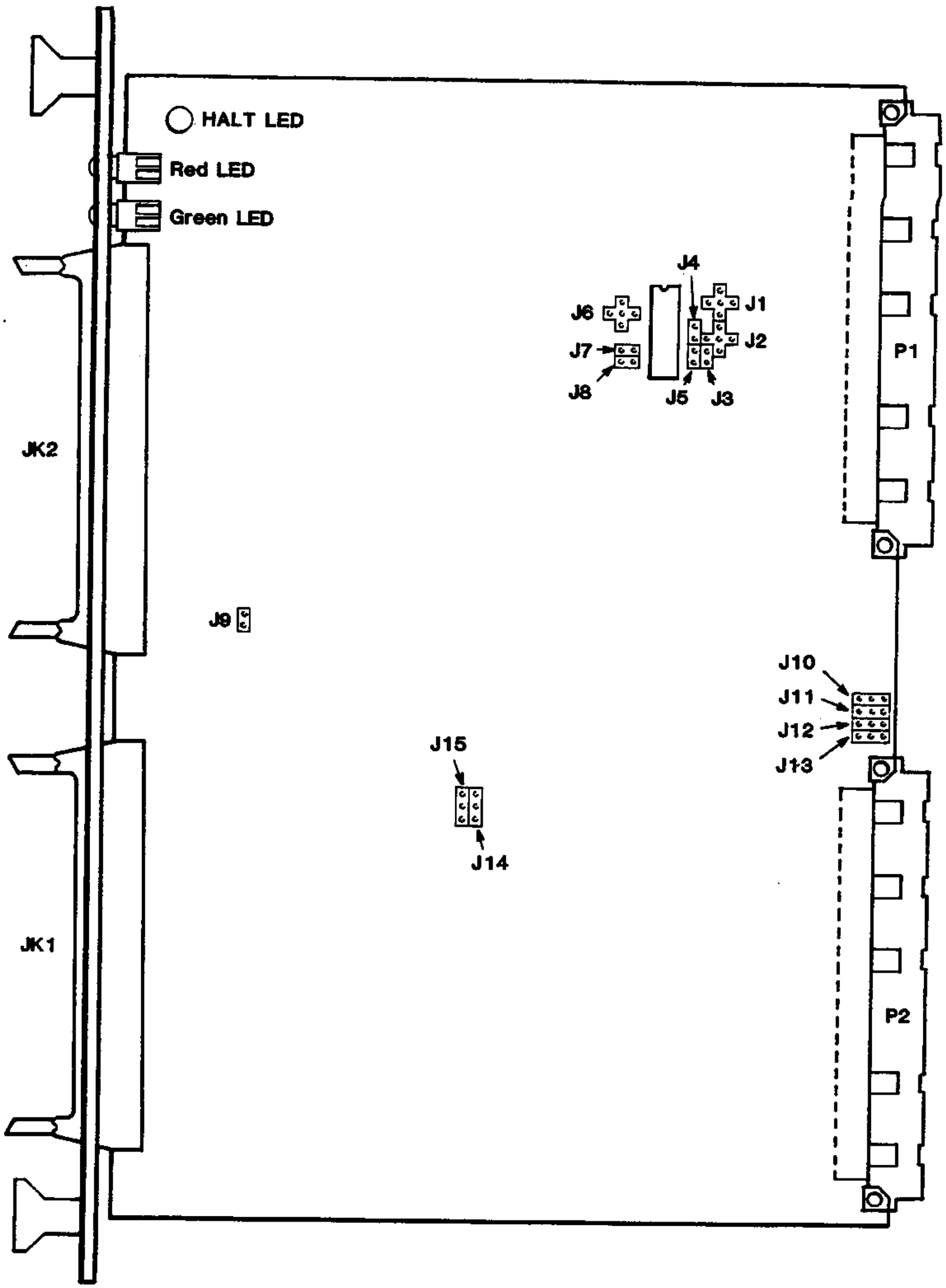


Figure 5-1. Jumper Locations

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4 JUMPERS

Table 5-1. ICM Jumper List

Jumper	Use
J1,J2,J6	The ICM does not use these jumpers (see Section 5.4.3, "Data Bus Request Jumpers" for proper setting)
J3	VMEbus Address Modifier Selection
J4,J5,J7,J8	Module Base Address Selectors
J9	Selects PROM cycle timing to accommodate different speed devices if the factory-installed PROMs are changed.
J10,J11,J12, and J13	If the factory-installed PROMs are changed, these jumpers are used to indicate the memory size of the new devices.
J14,J15	If the factory-installed RAM chips are changed, these jumpers are used to indicate the memory size of the new devices.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.1 Base Address Jumpers

The ICM can be configured to be addressed at the beginning of any one of 16 different 1K blocks within the first 16K (i.e., 0000H thru 3C00H) of the Short I/O Address space by positioning the jumpers J4, J5, J7 and J8 (see Figure 5-1 for board location) as shown below:

Table 5-2. Base Address Jumper Options

Jumpers J8	J7	J5	J4	VME base address in VME Short I/O Address space
IN	IN	IN	IN	0000H
IN	IN	IN	OUT	0400H
IN	IN	OUT	IN	0800H
IN	IN	OUT	OUT	0C00H
IN	OUT	IN	IN	1000H
IN	OUT	IN	OUT	1400H
IN	OUT	OUT	IN	1800H
IN	OUT	OUT	OUT	1C00H
OUT	IN	IN	IN	2000H
OUT	IN	IN	OUT	2400H
OUT	IN	OUT	IN	2800H
OUT	IN	OUT	OUT	2C00H
OUT	OUT	IN	IN	3000H
OUT	OUT	IN	OUT	3400H
OUT	OUT	OUT	IN	3800H
OUT	OUT	OUT	OUT	3C00H

Figure 5-2 shows the location of the Base Address jumpers on the board. A jumper is defined as "in" when it is connecting the two posts at the jumper position together.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

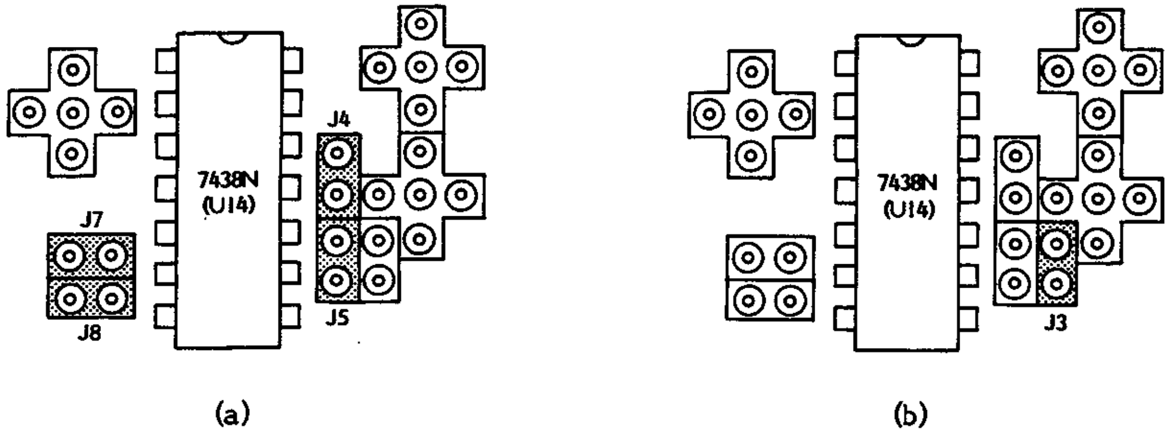


Figure 5-2. Base Address (a) and Address Modifier (b) Jumpers

5.4.2 Address Modifier Jumpers

The ICM has one jumper that determines which Address Modifier Codes it will respond to. This jumper is J3 (see Figure 5-2 for board location).

As an I/O module located in the VMEbus Short I/O Address space, the ICM is only concerned with two VMEbus Address Modifiers. These address modifiers are:

2DH A module generating this address modifier has "user" and "supervisory" Short I/O access privileges.

29H A module generating this address modifier has only "user" Short I/O access privileges.

If jumper J3 is installed, the slave module will respond to either "supervisory" or "user" address modifier codes (2DH or 29H). Removing J3 causes the slave to respond to the supervisory address modifier code (2DH) only (see Table 5-3).

Table 5-3. Address Modifier Jumper Options

Jumper J3	Address Modifier Code ICM Will Respond To
In	2DH or 29H
Out	2DH

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.3 Bus Request Level Jumpers

The ICM is based on the XYCOM Intelligent I/O Kernel, which is capable of "master" or "slave" operation. However, the standard firmware functions, as implemented for the ICM, do not provide for utilization of the kernel as a master. Hence, the Data Transfer Bus (DTB) Requestor is not used, and jumper J6 which activates the requestor, is not installed.

Since the ICM module is a slave only in the shipped configuration, it is configured to level 3, but not to pass the bus grant signals through the on-board logic. The user must install jumpers on the P1 backplane to pass these signals across. This is done to minimize the time required to pass these signals through this slave module.

For users generating their own firmware who wish to utilize the "master" capabilities of this module, the following table (Table 5-4) defines the available jumper configurations.

Table 5-4. Requestor Jumper Positions

Level (x)	J1 BGxIN	J2 BGxOUT	J6 BRx
0	D	C	B
1	C	D	D
2	B	A	C
3	A	B	A

NOTE

Jumpers are factory-positioned:

J1-A, J2 not installed, J6 not installed.

This deactivates the Bus Request/Bus Grant circuitry.

The user must install jumpers on the P1 backplane to pass BG0*, BG1*, BG2*, and BG3* for the slot the XVME-230 is installed in.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

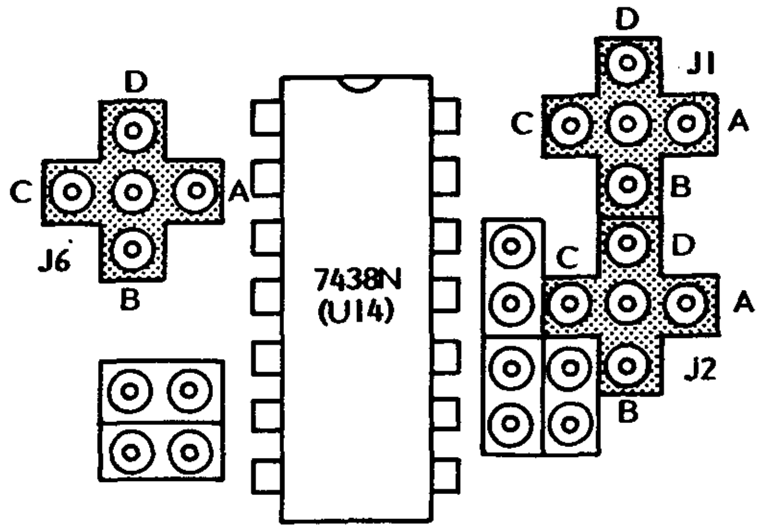


Figure 5-3. Bus Request Jumpers

5.4.4 Memory Range Selection Jumpers

The Memory Range Selection jumpers are briefly described in this section only to inform the user of the jumper's function. These jumpers are factory configured and should not be modified for normal module operation.

The ICM has two pairs of 28-pin sockets for byte-wide memory devices, one pair for EPROM and one pair for RAM. The EPROM sockets can be configured with jumpers to accept devices from 4Kx8 to 32Kx8 (ICM uses 16Kx8). Socket pins 1, 2, 27, and 28 are not used for 24-pin EPROMs. The RAM sockets can be jumpered for 2Kx8 or 8Kx8 (ICM uses 8Kx8).

The relative positions of jumpers J10 thru J15 on the board are shown in Figure 5-1. All of these jumpers are "two position" jumpers (i.e., they have an "A" or a "B" position). Figure 5-4 shows the jumper posts and the location points for "A" position jumpers and "B" position jumpers.

The following table shows how to configure jumpers J10, J11, J12, J13, J14, and J15 to allow the use of different size memory devices.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For normal ICM operation, the Memory Range Selection jumpers should be in the following configuration.

Table 5-5. Memory Jumper Options

Jumper	Jumper Position
J10	A
J11	B
J12	A
J13	B
J14	B
J15	B

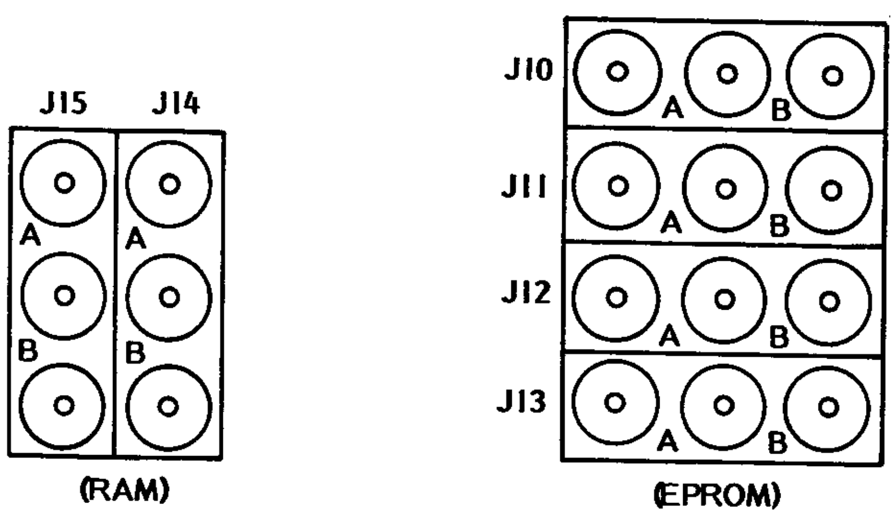


Figure 5-4. Memory Range Jumpers

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.5 Memory Access Time Control Jumper

Jumper J9 is used to set the EPROM access cycle timing to accommodate different speed devices. This jumper is factory installed and should not be removed. The location of the Memory Access Time Control jumper is shown in Figure 5-1.

5.5 CONNECTOR DESCRIPTIONS

The ICM interfaces to the outside world via two 50-pin connectors (see Figure 5-5). These connectors provide all required input and output signals.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

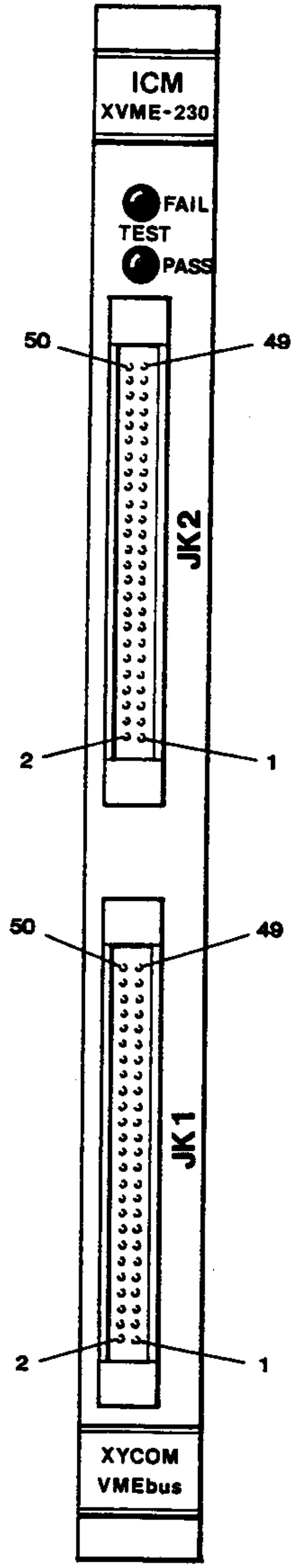


Figure 5-5. View of Connectors JK1 and JK2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.5.1 Signal Description

The CLOCK, GATE and OUTPUT signals available on the pin connectors are the three signals provided by each of the ICM's 16 counters described in Section 4.2. All odd pin numbers on JK1 and JK2 are signal ground.

5.5.2 Input Buffers

Each CLOCK and GATE input on the ICM can interface directly to TTL voltage levels. The inputs are buffered and have a minimum of 200 mV of hysteresis for improved noise immunity over standard TTL levels, and are also tied to VCC through a 4.7K ohm pull-up resistor.

The Input Buffers have the following specifications:

Voltage (low level - max.)	0.7 V
Voltage (high level - min.)	2.4V
Hysteresis	200 mV
Input Current (low level)	-200 uA
Input Current (high level)	20 uA

5.5.3 Output Drivers

Each ICM Output signal has an output driver with tri-state capability. During power-up diagnostics testing, the output drivers are in the high impedance state.

The Output Drivers have the following specifications:

Voltage (low level - max.)	0.5 V
Voltage (high level - min.)	2.0V
High Level Output Current	-15 mA
Low Level Output Current	24 mA
Iozl	-20 uA
Iozh	20 uA

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.5.4 Signal Location on Pin Connectors

Table 5-6. Signal Location On JK1 and JK2

JK1				JK2			
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	Ground	26	BCLOCK0	1	Ground	26	DCLOCK0
2	ACLOCK0	27	Ground	2	CCLOCK0	27	Ground
3	Ground	28	BGATE0	3	Ground	28	DGATE0
4	AGATE0	29	Ground	4	CGATE0	29	Ground
5	Ground	30	BCLOCK1	5	Ground	30	DCLOCK1
6	ACLOCK1	31	Ground	6	CCLOCK1	31	Ground
7	Ground	32	BGATE1	7	Ground	32	DGATE1
8	AGATE1	33	Ground	8	CGATE1	33	Ground
9	Ground	34	BCLOCK2	9	Ground	34	DCLOCK2
10	ACLOCK2	35	Ground	10	CCLOCK2	35	Ground
11	Ground	36	BGATE2	11	Ground	36	DGATE2
12	AGATE2	37	Ground	12	CGATE2	37	Ground
13	Ground	38	BCLOCK3	13	Ground	38	DCLOCK3
14	ACLOCK3	39	Ground	14	CCLOCK3	39	Ground
15	Ground	40	BGATE3	15	Ground	40	DGATE3
16	AGATE3	41	Ground	16	CGATE3	41	Ground
17	Ground	42	BOUT0	17	Ground	42	DOUT0
18	AOUT0	43	Ground	18	COUT0	43	Ground
19	Ground	44	BOUT1	19	Ground	44	DOUT1
20	AOUT1	45	Ground	20	COUT1	45	Ground
21	Ground	46	BOUT2	21	Ground	46	DOUT2
22	AOUT2	47	Ground	22	COUT2	47	Ground
23	Ground	48	BOUT3	23	Ground	48	DOUT3
24	AOUT3	49	Ground	24	COUT3	49	Ground
25	Ground	50	Ground	25	Ground	50	Ground

5.6 MODULE INSTALLATION

The XYCOM VMEbus modules can accommodate typical VMEbus backplane construction. Figure 5-6 shows a standard VME chassis and a typical backplane configuration. There are two rows of backplane connectors depicted (i.e., the P1 backplane and the P2 backplane). The ICM requires both the P1 and P2 backplane.

NOTE

The P2 connector on the ICM is not always required due to the fact it provides only power and ground signals (see Appendix B).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

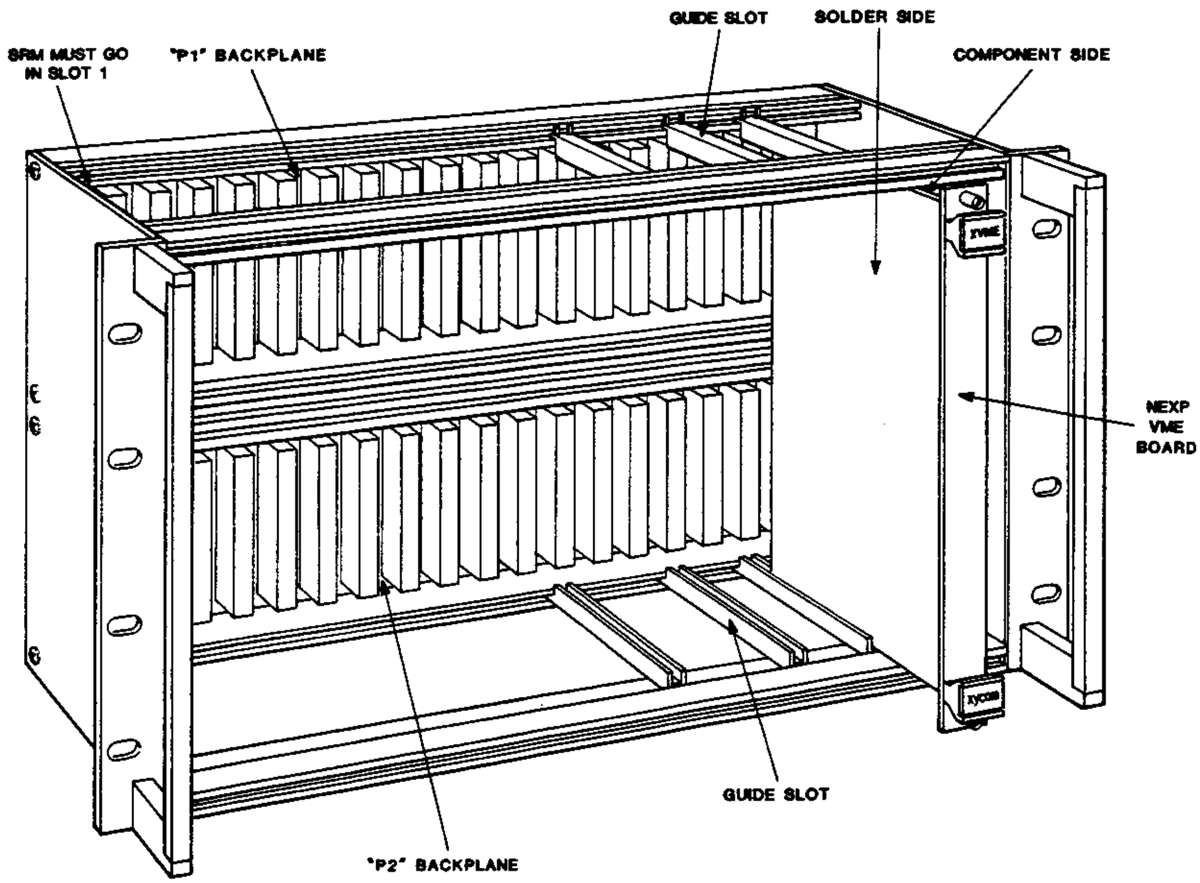


Figure 5-6. VMEbus Chassis

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.6.1 Installation Procedure

CAUTION

Do not attempt to install or remove any boards before turning off the power to the bus, and all related external power supplies.

Prior to installing a module, you should determine and verify all relevant jumper configurations and all connections to external devices or power supplies. (Please check the jumper configuration with the diagram and lists in the manual.)

To install a board in the cardcage, perform the following steps:

1) Make certain that the particular cardcage slot you are going to use is clear and accessible.

2) Center the board on the plastic guides so that the solder side is facing to the left and the component side is facing to the right (refer to Figure 5-6).

3) Push the card slowly toward the rear of the chassis until the connectors engage (the board should slide freely in the plastic guides).

4) Apply straight-forward pressure to the two handles on the outer edge of the board until the connectors are fully engaged and properly seated.

NOTE

It should not be necessary to use excess pressure or force to engage the connectors. If the board does not properly connect with the backplane, remove the module and inspect all connectors and guide slots for possible damage or obstructions.

5) Once the board is properly seated, it should be secured by tightening the two machine screws at the extreme top and bottom of the board.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 6

BASIC FUNCTIONS

6.1 INTRODUCTION

This chapter explains the operation, hardware connection, and programming of the following four measurement and control functions:

- 16-bit or 32-bit Event Counter
- 16-bit or 32-bit Event Counter with Limit Check
- 16-bit or 32-bit Frequency Divider
- 16-bit Timer capable of monitoring time periods expressed in units varying from one microsecond up to one second.
- 16-bit Periodic Interrupt function

6.2 EVENT COUNTER

Event counting is simply the accumulation of events. The Event Counter views an event as any rising edge signal pulse that occurs on its CLOCK input. Typical industrial applications would include part piece counting or machine cycle counting. The Event Counting function has two versions, a 16-bit and a 32-bit.

The 16-bit version of the Event Counting function requires only one counter to operate, so any or all of the 16 counters on the ICM can perform Event Counting duties.

The 32-bit version of the Event Counting function requires two counters to operate, therefore using an entire channel.

The Event Counter (16 or 32-bit mode) begins accumulating events immediately upon initialization. It can then be read at any time without disturbing the event counting process.

If the Event Counter overflows, it generates an error code (see Section 4.10 for code definitions) indicating an overflow has occurred, and then terminates.

When using the Event Counter the user can optionally supply a "gating" pulse to the counter's GATE input to enable and suspend counting.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.1 Hardware Configuration

The 16-bit version of the Event Counter is connected as follows:

1. Connect the signal from the event being counted to the CLOCK input of the counter to be configured for event counting.

The 32-bit version of the Event Counter is connected as follows:

1. Connect the signal from the event being counted to the CLOCK input of the even numbered counter on the channel. For example, if a 32-bit Event Counter is initialized on channel 1, the event pulse is connected to ACLOCK2.

If gating is not desired, only step 1 is required.

If upon initialization, a gating pulse has been specified to enable the event counting process (see Start Event Counter commands, Section 6.2.2), the gating pulse is connected as follows:

2. Connect the gating pulse to the GATE input of the counter being used for event counting. For example, if a 16-bit Event Counter is initialized on channel 3 Counter B2, the GATE pulse is connected to the BGATE2 input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 6-1. 16 and 32-Bit Event Counter Connection

32-Bit Event Counter			16-Bit Event Counter		
Channel	Input	Optional GATE	Counter	Input	Optional GATE
0	ACLOCK0	AGATE0	A0 A1	ACLOCK0 ACLOCK1	AGATE0 AGATE1
1	ACLOCK2	AGATE2	A2 A3	ACLOCK2 ACLOCK3	AGATE2 AGATE3
2	BCLOCK0	BGATE0	B0 B1	BCLOCK0 BCLOCK1	BGATE0 BGATE1
3	BCLOCK2	BGATE2	B2 B3	BCLOCK2 BCLOCK3	BGATE2 BGATE3
4	CCLOCK0	CGATE0	C0 C1	CCLOCK0 CCLOCK1	CGATE0 CGATE1
5	CCLOCK2	CGATE2	C2 C3	CCLOCK2 CCLOCK3	CGATE2 CGATE3
6	DCLOCK0	DGATE0	D0 D1	DCLOCK0 DCLOCK1	DCLOCK0 DCLOCK1
7	DCLOCK2	DGATE2	D2 D3	DCLOCK2 DCLOCK3	DCLOCK2 DCLOCK3

6.2.2 Programming

The 16 and 32-bit Event Counter functions are configured by sending the following two commands to any of the ICM's eight channels.

20H = Start 16-Bit Event Counter with Optional Limit Check
21H = Start 32-Bit Event Counter with Optional Limit Check

The present values of the 16 and 32-bit Event Counters are read by sending the following two commands:

24H = Read 16-Bit Counter
25H = Read 32-Bit Counter

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.2.1 Start 16-Bit Event Counter with Optional Limit Check

The Start 16-bit Event Counter with Optional Limit Check command configures the specified counter as an Event Counter. The command specifies if a gating pulse is to be applied to the counter's GATE input, and if so, what polarity this gating pulse must be. An event counting limit, after which the Event Counter generates a completion interrupt, can be specified optionally. If a limit has not been specified the command will terminate with a "counter overflow" response. The format is listed below:

Start 16-bit Event Counter with Optional Limit Check Command Format

Command Word: 20H

Four Operand Bytes:

Byte 1: Counter number	(0,1,2, or 3)
Byte 2: GATE indicator	0 = no gating 1 to 7FH = active high 80H to FFH = active low
Bytes 3-4: Optional Limit	This is a two byte value in units of event counts that specifies the limit. A value of zero indicates no limit checking.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A

XYCOM STANDARD I/O ARCHITECTURE

INTRODUCTION

The purpose of this Appendix is to define XYCOM's Standard I/O Architecture for XVME I/O modules. This Standard I/O Architecture has been incorporated on all XVME I/O modules in order to provide a simpler and more consistent method of programming for the entire module line. The I/O Architecture specifies the logical aspects of bus interfaces, as opposed to the "physical" or electrical aspects as defined in the VMEbus specifications. The module elements which are standardized by the XYCOM I/O Architecture are the following:

1. Module Addressing -- Where a module is positioned in the I/O address space and how software can read from it or write to it.
2. Module Identification -- How software can identify which modules are installed in a system.
3. Module Operational Status -- How the operator can (through software) determine the operational condition of specific modules within the system.
4. Interrupt Control -- How software is able to control and monitor the capability of the module to interrupt the system.
5. Communication between Modules -- How master (host) processors and intelligent I/O modules communicate through shared global memory or the dual-access RAM on the I/O modules.
6. The I/O Kernel -- How intelligent and non-intelligent "kernels" facilitate the operation of all XYCOM I/O modules.

MODULE ADDRESSING

All XYCOM I/O modules are designed to be addressed within the VMEbus-defined 64K short I/O address space. The restriction of I/O modules to the short I/O address space provides separation of program/data address space and the I/O address space. This convention simplifies software design and minimizes hardware and module cost, while at the same time, providing 64K of address space for I/O modules.

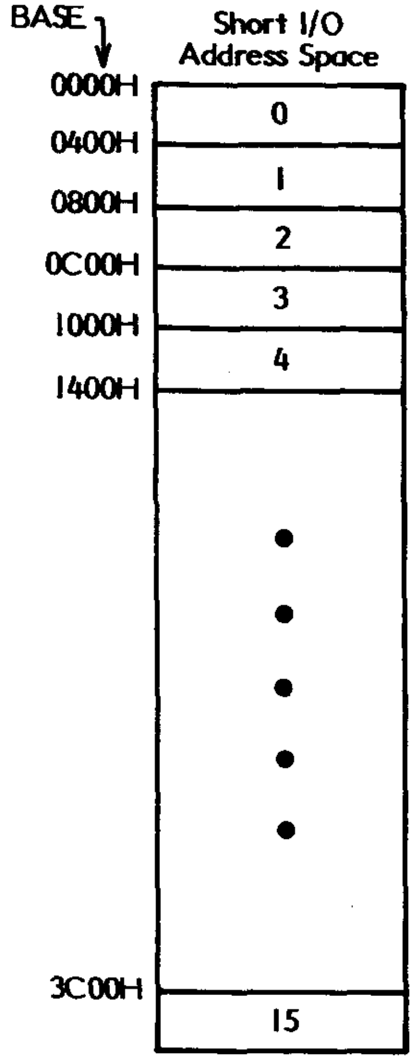
Base Addressing

Since each I/O module connected to the bus must have its own unique base address, the base addressing scheme for XYCOM VME I/O modules has been designed to be jumper-selectable. Each XVME I/O module installed in the system requires at least a 1K byte block of the short I/O memory. This divides the 64K short I/O address space into 64 1K segments. Thus, each I/O module has a base address which starts on a 1K boundary. As a result, the XYCOM I/O modules have all been implemented to decode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

base addresses in 1K (400H) increments. On an intelligent XVME module, address signals A10-A13 are decoded, while A14 and A15 must be zero. (This implies that only the lowest 16 of the possible 64 1K segments are used for intelligent modules.) On non-intelligent XVME modules, the six highest order short I/O address bits are decoded, while the remaining lower order bits are ignored. This arrangement provides the correct address configuration to allow each module address to begin on a 1K boundary. Non-intelligent XVME modules allow the use of six base address jumpers (representing bits A10-A15), and thus, they are able to reside on any of the 64 1K boundaries available in the short I/O address space. Intelligent XVME modules will only allow the use of four base address jumpers (representing bits A10-A13) which limits their selection of 1K boundaries to one of sixteen possible choices.

Figure A-1 shows an abbreviated view of the short I/O memory. The list shown next to the figure provides the sixteen possible addresses which could be used as base addresses by XVME intelligent I/O modules.



Base Address Jumper Options

Jumpers J8 A13	J7 A12	J5 A11	J4 A10	Module base address in VME Short I/O Address space
IN	IN	IN	IN	0000H
IN	IN	IN	OUT	0400H
IN	IN	OUT	IN	0800H
IN	IN	OUT	OUT	0C00H
IN	OUT	IN	IN	1000H
IN	OUT	IN	OUT	1400H
IN	OUT	OUT	IN	1800H
IN	OUT	OUT	OUT	1C00H
OUT	IN	IN	IN	2000H
OUT	IN	IN	OUT	2400H
OUT	IN	OUT	IN	2800H
OUT	IN	OUT	OUT	2C00H
OUT	OUT	IN	IN	3000H
OUT	OUT	IN	OUT	3400H
OUT	OUT	OUT	IN	3800H
OUT	OUT	OUT	OUT	3C00H

Figure A-1. 64K Short I/O Address Space

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Standardized Module I/O Map

The 1K block of short I/O addresses (called the I/O Interface Block) allocated to each XVME module is mapped with a standardized format in order to simplify programming and data access. The locations of frequently used registers and module-specific identification information are uniform. For example, the module identification information is always found in the first 32 odd bytes of the module memory block -- with these addresses being relative to the jumpered base address (i.e., Module I.D. data address = base address + odd bytes 1H - 3FH). The byte located at base address+81H on each module contains a Status/Control register which provides the results of diagnostics for verification of the module's operational condition. The next area of the module I/O Interface Block (base address + 82H - roughly 120F) is module-specific and it varies in size from one module to the next. It is in this area that the module holds specific I/O status, data, and pointer registers for use with IPC protocol. All intelligent XVME I/O modules have an area of their I/O Interface Blocks defined as "dual access RAM." This area of memory provides the space where XVME "slave" I/O modules access their command blocks and where XVME "master" modules could access their command blocks (i.e., "master" modules can also access global system memory).

The remainder of the I/O Interface Block is then allocated to various module-specific tasks, registers, buffers, ports, etc.

Figure A-2 shows an address map of an XVME I/O module interface block, and how it relates to the VMEbus short I/O address space. Notice that any location in the I/O Interface Block may be accessed by simply using the address formula:

Module Base Address + Relative Offset = Desired Location

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

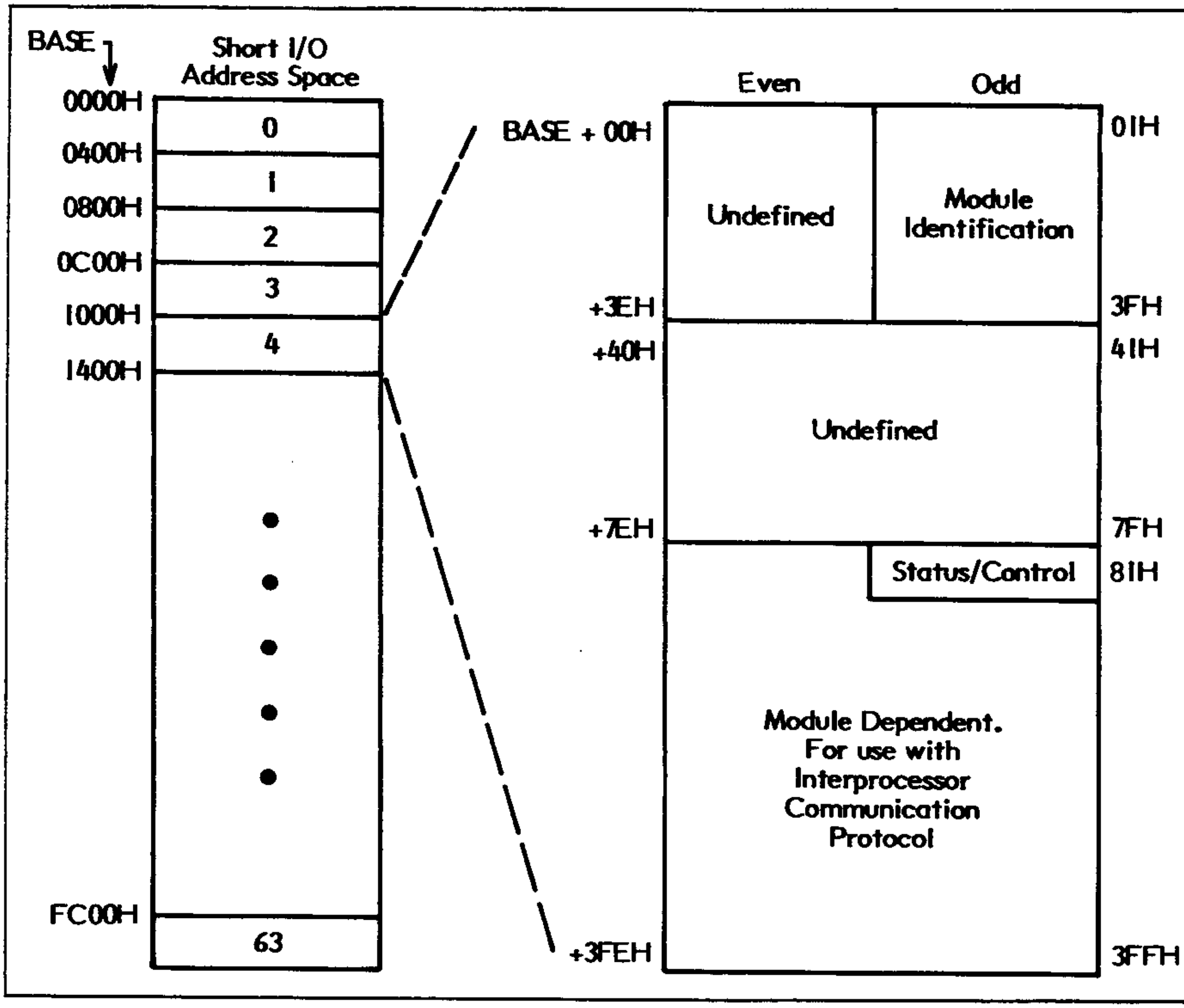


Figure A-2. XVME I/O Module Address Map

MODULE SPECIFIC IDENTIFICATION DATA

The module identification scheme provides a unique method of registering module specific information in an ASCII encoded format. The I.D. data is provided as thirty-two ASCII encoded characters consisting of the board type, manufacturer identification, module model number, number of 1K-byte blocks occupied by the module, and model functional revision level information. This information can be studied by the system processor on power-up to verify the system configuration and operational status. Table A-1 defines the Identification information locations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-1. Module I.D. Data

Offset Relative to Module Base	Contents	ASCII Encoding (in hex)	Descriptions
1	V	56	
3	M	4D	ID PROM identifier,
5	E	45	always "VMEID"
7	I	49	(5 characters)
9	D	44	
B	X	58	Manufacturers I.D.,
D	Y	59	always "XYC" for XYCOM
F	C	43	Modules (3 characters)
11	2	34	
13	3	32	Module Model Number
15	0	30	(3 characters and
17		20	4 trailing blanks)
19		20	
1B		20	
1D		20	
1F	1	31	Number of 1K byte blocks of I/O space occupied by this module (1 character)
21		20	Major functional revision
23	1	31	level with leading blank (if single digit)
25	0	30	Minor functional revision
27		20	level with trailing blank (if single digit)
29	Undefined		
2B	"		
2D	"		
2F	"		
31	"		
33	"		
35	"		Manufacturer
37	"		Dependent Information,
39	"		Reserved for Future Use
3B	"		
3D	"		
3F	"		

The module has been designed so that it is only necessary to use odd backplane addresses to access the I.D. data. Thus, each of the 32 bytes of ASCII data have been assigned to the first 32 odd I/O Interface Block bytes (i.e., odd bytes 1H-3FH).

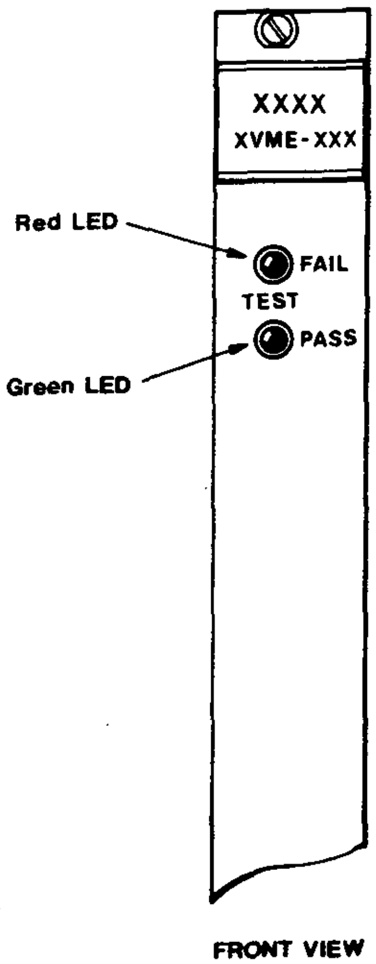
I.D. information can be accessed simply by addressing the module base, offset by the specific address for the character(s) needed. For example, if the base address of the board is jumpered to 1000H, and if you wish to access the module model number (I/O interface block locations 11H, 13H, 15H, 17H, 19H, 1BH, and 1DH), you will individually add the offset addresses to the base addresses to read the hex-coded ASCII value at each location. Thus, in this example, the ASCII values which make up the module model number are found sequentially at locations 1011H, 1013H, 1015H, 1017H, 1019H, 101BH, and 101DH within the system's short I/O address space.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MODULE OPERATIONAL STATUS/CONTROL

All XVME intelligent I/O modules are designed to perform diagnostic self-tests on power-up or reset. For non-intelligent modules, the user must provide the diagnostic program. The self-test provision allows the user to verify the operational status of a module by either visually inspecting the two LEDs which are mounted on the module front panel (see Figure A-3), or by reading the module status byte (located at module base address + 81H).

Figure A-3 shows the location of the status LEDs on the module front panel. The two tables included with Figure A-3 define the visible LED states for the module test conditions on both the intelligent I/O modules and the non-intelligent I/O modules.



Status/Control Bits 3	2	1	0	LEDs Green	Red*	SYSFAIL*	Status
0	0	0	0	OFF	ON	ON	Module not yet tested
1	0	0	0	OFF	ON	ON	Module failed test
0	0	0	1	OFF	OFF	OFF	Inactive module
0	1	1	0	ON	ON	ON	Module undergoing test
1	1	1	1	ON	OFF	OFF	Module passed test
all others				X	X	X	Invalid and undefined

INTELLIGENT MODULE STATUS

Status/Control Bits 1	0	LEDs Green	Red*	SYSFAIL*	Status
0	0	OFF	ON	ON	Module failed, or not yet tested
0	1	OFF	OFF	OFF	Inactive module
1	0	ON	ON	ON	Module undergoing test
1	1	ON	OFF	OFF	Module passed test

NON-INTELLIGENT MODULE STATUS

Figure A-3. Module LED Status

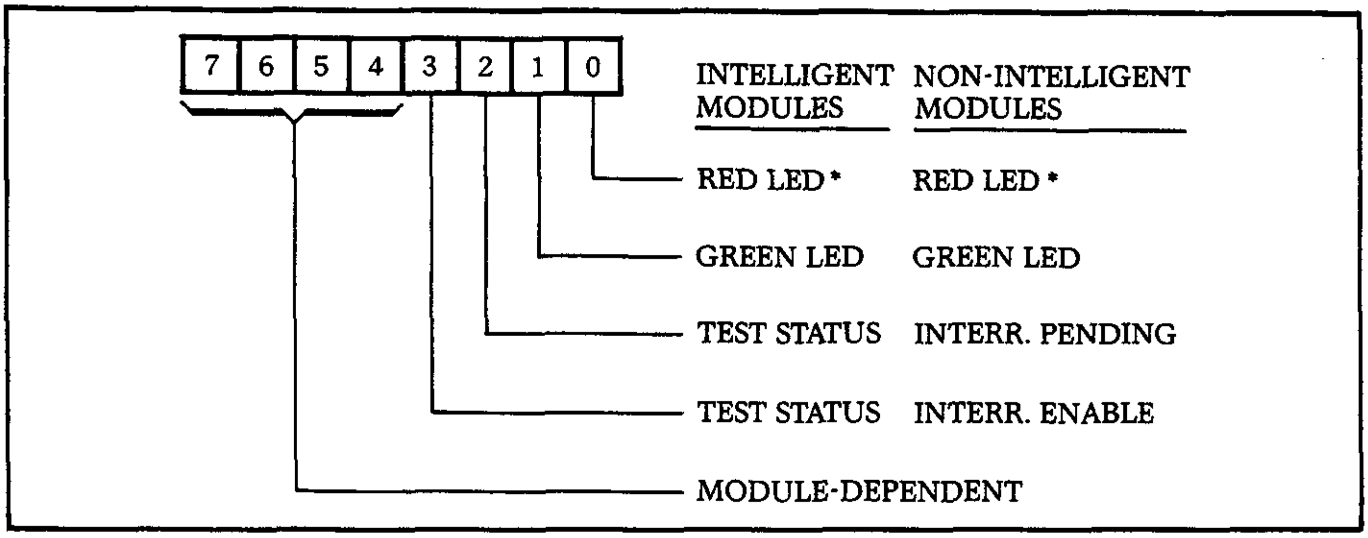
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The module status/control register (found at module base address + 81H) on intelligent XVME I/O modules provides the current status of the module self-test in conjunction with the current status of the front panel LEDs. The status register on intelligent modules is a "Read Only" register and it can be read by software to determine if the board is operating properly.

On non-intelligent XVME I/O modules, the status/control register is used to indicate the state of the front panel LEDs, and to set and verify module generated interrupts. The LED status bits are "Read/Write" locations which provide the user with the indicators to accomodate diagnostic software. The Interrupt Enable bit is also a Read/Write location which must be written to in order to enable module-generated interrupts. The Interrupt Pending bit is a "Read Only" bit which indicates a module-generated pending interrupt.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure A-4 shows the status/control register bit definitions for both intelligent and non-intelligent XVME I/O modules.



Bit	Non-Intelligent Modules
0	Read/Write - Red LED 0 = Red LED On 1 = Red LED Off
1	Read/Write - Green LED 0 = Green LED Off 1 = Green LED On
2	Read Only - Interrupt Pending 0 = No Interrupt 1 = Interrupt Pending
3	Read/Write - Interrupt Enable 0 = Interrupts Not Enabled 1 = Interrupts Enabled
4	Module dependent
5	Module dependent
6	Module dependent
7	Module dependent

Bit	Intelligent Modules
0	Read Only - Red LED 0 = Red LED On 1 = Red LED Off
1	Read Only - Green LED 0 = Green LED Off 1 = Green LED On
2 & 3	Read Only - Test Status Indicators
4	Module dependent
5	Module dependent
6	Module dependent
7	Module dependent

Bit 3	Bit 2		
0	0	=	Self-test not started
0	1	=	Selt-test in progress
1	0	=	Self-test failed
1	1	=	Self-test passed

Figure A-4. Status Register Bit Definitions

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INTERRUPT CONTROL

Interrupts for non-intelligent modules can be enabled or disabled by setting/clearing the Interrupt Enable bit in the module status register. The status of pending on-board interrupts can also be read from this register. Interrupt control for intelligent modules is handled by the Interprocessor Communications Protocol.

Communications Between Processors

Communications between an intelligent "master" and an intelligent "slave" I/O module is governed by XYCOM's Interprocessor Communication (IPC) Protocol. This protocol involves the use of 20-byte Command Block data structures, which can be located anywhere in shared global RAM or dual-access RAM on an I/O module, to exchange commands and data between a host processor and an I/O module. Interprocessor Communication Protocol is thoroughly explained in Chapter 3 of this manual.

THE KERNEL

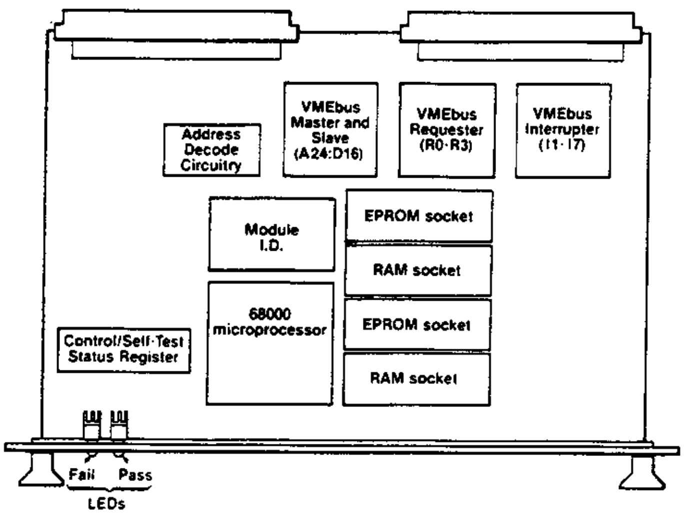
To standardize its XVME I/O modules, XYCOM has designed them around "kernels" common from module to module. Each different module type consists of a standard kernel, combined with module-dependent application circuitry. Module standardization results in more efficient module design and allows the implementation of the Standard I/O Architecture. The biggest benefit of standardization for intelligent modules is that it allows the use of a common command language or protocol (Interprocessor Communication Protocol in this case).

The intelligent kernel is based around a 68000 microprocessor. This design provides the full complement of VMEbus Requester and Interrupter options for master/slave interfacing, as well as all of the advantages provided by the various facets of the XYCOM Standard I/O Architecture (as covered earlier in this appendix).

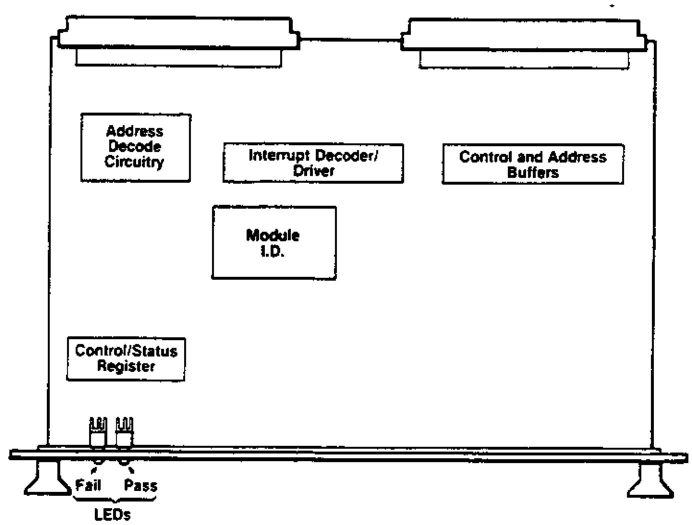
The non-intelligent kernel provides the circuitry required to receive and generate all of the signals for a VMEbus defined 16-bit "slave" module. The non-intelligent kernel also employs the features of the XYCOM Standard I/O Architecture (as described earlier in this Appendix).

The simplified diagrams below show the features of both the intelligent and the non-intelligent kernels.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



INTELLIGENT KERNEL



NON-INTELLIGENT KERNEL

Figure A-5. Intelligent and Non-Intelligent Kernels

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix B

VMEbus CONNECTOR/PIN DESCRIPTION

The Intelligent Counter module is physically configured as a non-expanded (NEXP), double-height, VMEbus compatible board. There are two 96 pin bus connectors on the rear edge of the board, labelled P1 and P2 (refer to Chapter 5, Figure 5-1 for the locations). The pin connections for P1 contain the standard address, data, and control signals necessary for the operation of NEXP modules. P2 is used by the ICM for power and ground signals.

Table B-1. P1 - VMEbus Signal Identification

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
ACFAIL*	1B: 3	AC FAILURE: Open-collector driven signal which indicates that the AC input to the power supply is no longer being provided or that the required input voltage levels are not being met.
IACKIN*	1A: 21	INTERRUPT ACKNOWLEDGE IN: Totem-pole driven signal. IACKIN* and IACKOUT* signals form a daisy-chained acknowledge. The IACKIN* signal indicates to the VME board that an acknowledge cycle is in progress.
IACKOUT*	1A: 22	INTERRUPT ACKNOWLEDGE OUT: Totem-pole driven signal. IACKIN* and IACKOUT* signals form a daisy-chained acknowledge. The IACKOUT* signal indicates to the next board that an acknowledge cycle is in progress.
AM0-AM5	1A: 23 1B: 16,17, 18,19 1C: 14	ADDRESS MODIFIER (bits 0-5): Three-state driven lines that provide additional information about the address bus, such as size, cycle type, and/or DTB master identification.
AS*	1A: 18	ADDRESS STROBE: Three-state driven signal that indicates a valid address is on the address bus.
A01-A23	1A: 24-30 1C: 15-30	ADDRESS bus (bits 1-23): Three-state driven address lines that specify a memory address.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-1. VMEbus Signal Identification (cont'd)

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
A24-A31	2B: 4-11	ADDRESS bus (bits 24-31): Three-state driven bus expansion address lines.
BBSY*	1B: 1	BUS BUSY: Open-collector driven signal generated by the current DTB master to indicate that it is using the bus.
BCLR*	1B: 2	BUS CLEAR: Totem-pole driven signal generated by the bus arbitrator to request release by the current DTB master if a higher level is requesting the bus.
BERR*	1C: 11	BUS ERROR: Open-collector driven signal generated by a slave. It indicates that an unrecoverable error has occurred and the bus cycle must be aborted.
BG0IN*-BG3IN**	1B: 4,6, 8,10	BUS GRANT (0-3) IN: Totem-pole driven signals generated by the Arbiter or Requesters. Bus Grant In and Out signals form a daisy-chained bus grant. The Bus Grant In signal indicates to this board that it may become the next bus master.
BG0OUT*-BG3OUT*	1B: 5,7, 9,11	BUS GRANT (0-3) OUT: Totem-pole driven signals generated by Requesters. Bus Grant In and Out signals form a daisy-chained bus grant. The Bus Grant Out signal indicates to the next board that it may become the next bus master.
BR0*-BR3*	1B: 12-15	BUS REQUEST (0-3): Open-collector driven signals generated by Requesters. These signals indicate that a DTB master in the daisy-chain requires access to the bus.
DS0*	1A: 13	DATA STROBE 0: Three-state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines (D00-D07).
DS1*	1A: 12	DATA STROBE 1: Three state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines (D08-D15).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-1. VMEbus Signal Identification (cont'd)

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
DTACK*	1A: 16	DATA TRANSFER ACKNOWLEDGE: Open-collector driven signal generated by a DTB slave. The falling edge of this signal indicates that valid data is available on the data bus during a read cycle, or that data has been accepted from the data bus during a write cycle.
D00-D15	1A: 1-8 1C: 1-8	DATA BUS (bits 0-15): Three-state driven bi-directional data lines that provide a data path between the DTB master and slave.
GND	1A: 9,11, 15,17,19 1B: 20,23 1C: 9 2B: 2,12 22,31	GROUND
IACK*	1A: 20	INTERRUPT ACKNOWLEDGE: Open-collector or three-state driven signal from any Master processing an interrupt request. It is routed via the backplane to Slot 1, where it is looped back to become Slot 1 IACKIN* to start the interrupt acknowledge daisy-chain.
IRQ1*-IRQ7*	1B: 24-30	INTERRUPT REQUEST (1-7): Open-collector driven signals, generated by an interrupter, which carry prioritized interrupt requests. Level seven is the highest priority.
LWORD*	1C: 13	LONGWORD: Three-state driven signal to indicate that the current transfer is a 32-bit transfer.
(RESERVED)	2B: 3	RESERVED: Signal line reserved for future VMEbus enhancements. This line must not be used.
SERCLK	1B: 21	A reserved signal which will be used as the clock for a serial communication bus protocol which is still being finalized.
SERDAT	1B: 22	A reserved signal which will be used as the transmission line for serial communication bus messages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-1. VMEbus Signal Identification (cont'd)

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
SYSCLK	1A: 10	SYSTEM CLOCK: A constant 16-MHz clock signal that is independent of processor speed or timing. It is used for general system timing use.
SYSFAIL*	1C: 10	SYSTEM FAIL: Open-collector driven signal that indicates that a failure has occurred in the system. It may be generated by any module on the VMEbus.
SYSRESET*	1C: 12	SYSTEM RESET: Open-collector driven signal which, when low, will cause the system to be reset.
WRITE*	1A: 14	WRITE: Three-state driven signal that specifies the data transfer cycle in progress to be either read or written. A high level indicates a read operation; a low level indicates a write operation.
+5V STDBY	1B: 31	+5 Vdc STANDBY: This line supplies +5 Vdc to devices requiring battery backup.
+5V	1A: 32 1B: 32 1C: 32 2B: 1,13,32	+5 Vdc Power: Used by system logic circuits.
+12V	1C: 31	+12 Vdc Power: Used by system logic circuits.
-12V	1A: 31	-12 Vdc Power: Used by system logic circuits.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BACKPLANE CONNECTOR P1

The following table lists the P1 pin assignments by pin number order. (The connector consists of three rows of pins labeled rows A, B, and C.)

Table B-2. P1 Pin Assignments

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GDN
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK (1)	A17
22	IACKOUT*	SERDAT (1)	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12V	+5V STDBY	+12V
32	+5V	+5V	+5V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table B-3. P2 - Power and Ground Signals

Signal Mnemonic	Connector and Pin Number	Signal Name
GND	2B:2 2B:12 2B:22 2B:31	Ground
+5V	2B:1 2B:13 2B:32	+5Vdc Power used by the system logic circuits

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BACKPLANE CONNECTOR P2

The following table lists the P2 pin assignments by pin number order. (The connector consists of three rows of pins labeled A, B, and C, but the B row is not used.)

Table B-4. P2 Pin Assignments

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	not	+5 Volts	not
2	used	GND	used
3		reserved	
4		not used	
5		not used	
6		not used	
7		not used	
8		not used	
9		not used	
10		not used	
11		not used	
12		GND	
13		+5 Volts	
14		not used	
15		not used	
16		not used	
17		not used	
18		not used	
19		not used	
20		not used	
21		not used	
22		not used	
23		GND	
24		not used	
25		not used	
26		not used	
27		not used	
28		not used	
29		not used	
30		not used	
31		GND	
32		+5 Volts	

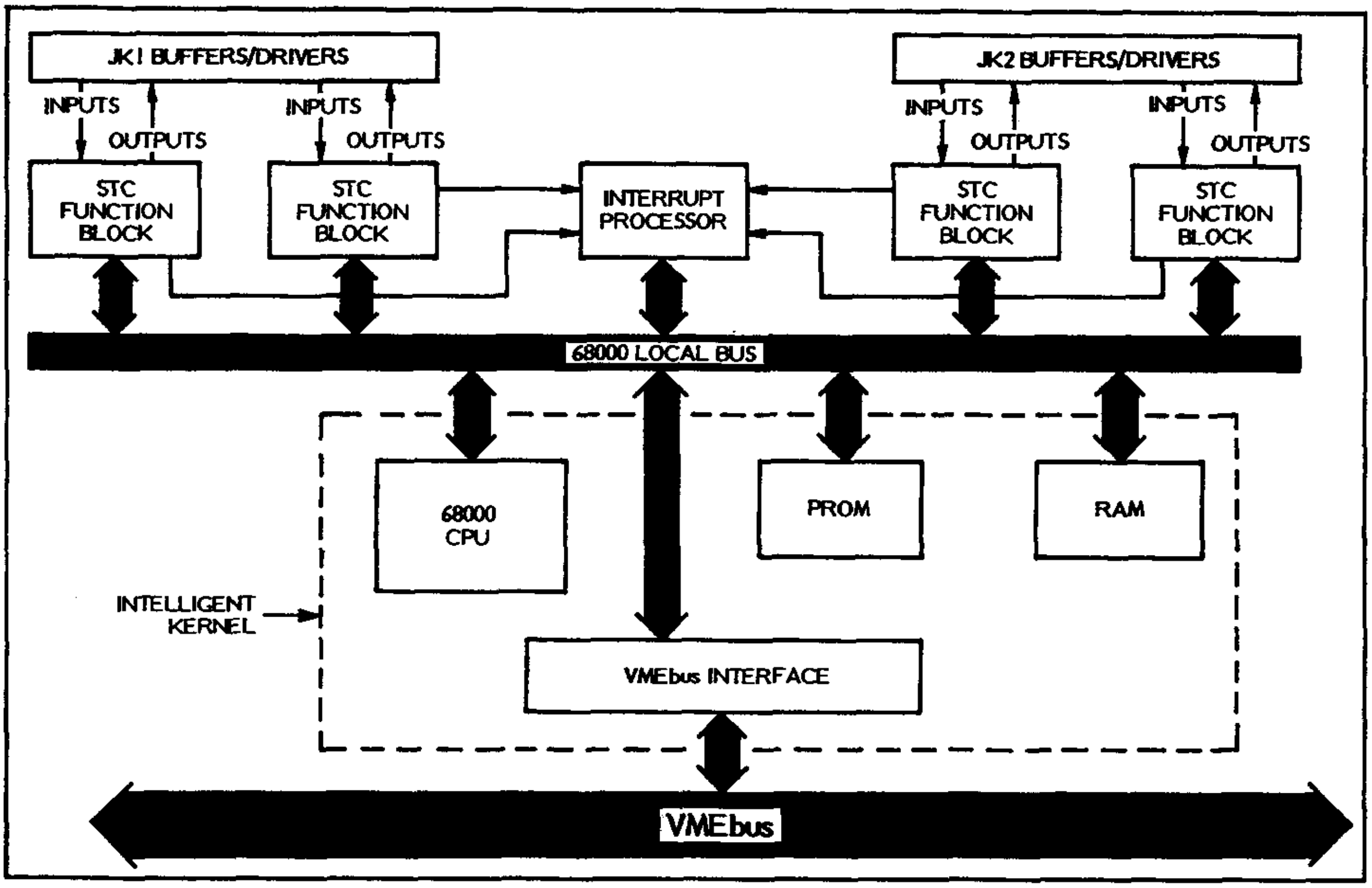
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix C

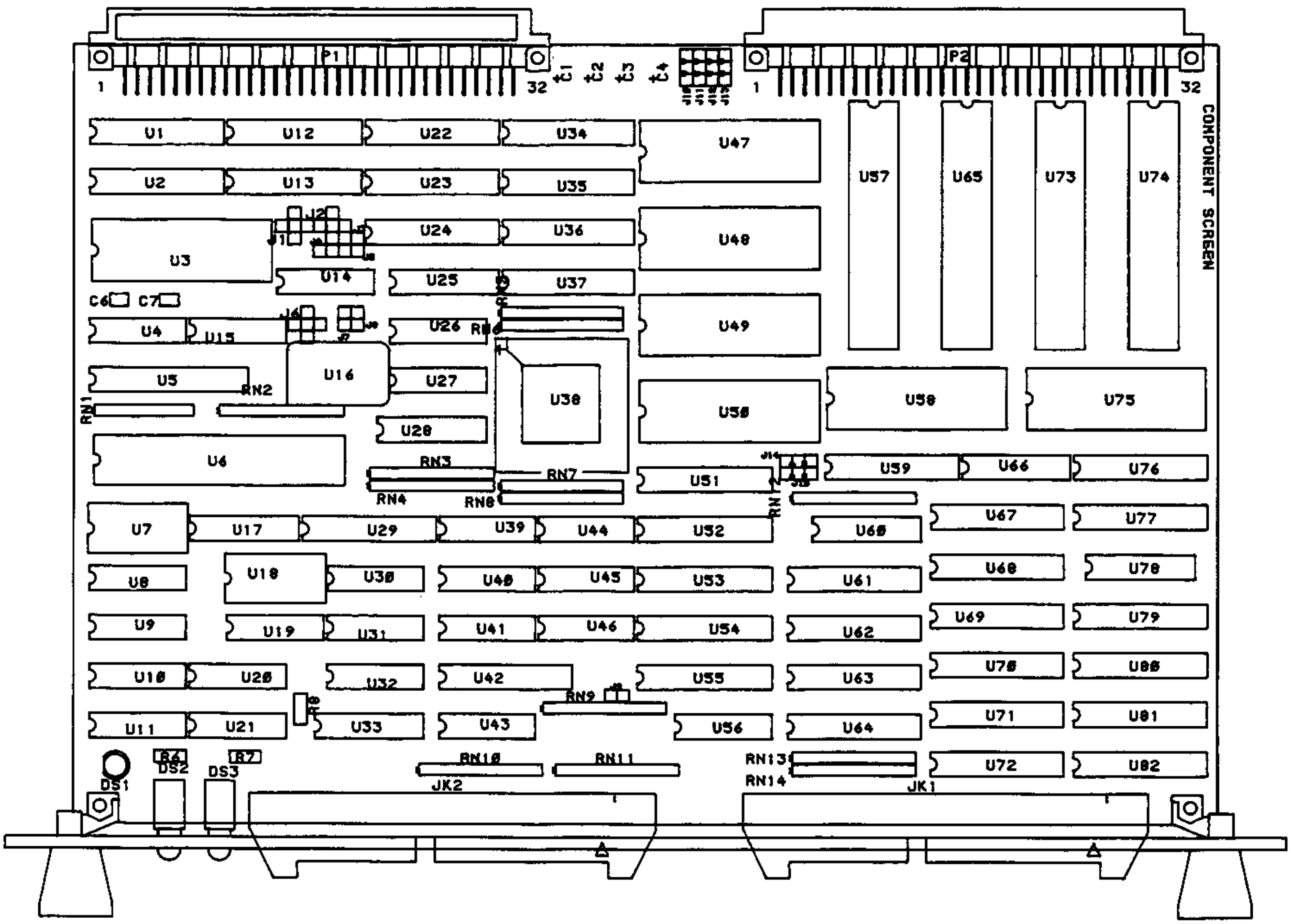
SCHEMATICS AND DIAGRAMS

Block Diagram



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Assembly Drawing



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.2.2 Start 32-Bit Event Counter with Optional Limit Check

The Start 32-bit Event Counter with Optional Limit Check command configures the specified channel as a 32-bit Event Counter. It specifies if a gating pulse is required, and if so, the polarity of this gating pulse. An event counting limit after which the counter generates a completion interrupt, can be specified optionally. If a limit has not been specified the command will terminate with a "counter overflow" response. The format is listed below:

Start 32-bit Event Counter with Optional Limit Check Command Format

Command Word: 21H

Six Operand Bytes:

Byte 1: Unused

Byte 2: GATE indicator ———— 0 = no gating
1 to 7FH = active high
80H to FFH = active low

Bytes 3-6: Optional Limit ———— This is a four byte value in units of event counts that specifies the limit. A value of zero indicates no limit checking.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.2.3 Read 16-Bit Counter

The Read 16-bit Counter command returns the value currently accumulated by the specified counter. This command returns a response immediately. The format is listed below:

Read 16-bit Counter Command Format

Command Word: 24H

Four Operand Bytes:

Byte 1: Counter number ——————— (0,1,2, or 3)

Byte 2: Unused

Bytes 3-4: Returned Counter Value ———— This area of the operand field is a two-byte value that represents the current value of the 16-bit counter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.2.4 Read 32-Bit Counter

The Read 32-bit Counter command returns the value currently accumulated by the specified channel. This command returns a response immediately. The format is listed below:

Read 32-bit Counter Command Format

Command Word: 25H

Six Operand Bytes:

Bytes 1-2: Unused

Bytes 3-6: Returned Counter Value ——— This is a four-byte value that contains the current value of the 32-bit counter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3 FREQUENCY DIVIDER

The Frequency Divider function divides the incoming signal by a divisor which is specified upon initialization. The function then provides an output signal that has the period of the divided incoming signal. Similar to the Event Counter function, the Frequency Divider function has a 16-bit version and a 32-bit version.

6.3.1 Hardware Configuration

As in the Event Counter function, for 16-bit operation the counter must be specified. The frequency to be divided is then connected to the CLOCK input of the chosen counter. External gating is not an option on this function.

For 32-bit operation the following hardware modification must be made. The output of the odd-numbered counter on the channel being used must be jumpered to the GATE of the even-numbered counter of the channel. For example, if a 32-bit frequency divider is to operate on channel 2 BOUT1 must be jumpered to BGATE0.

Table 6-2. Frequency Divider Connection

32-Bit Frequency Divider			16-Bit Frequency Divider		
Channel	Input	Output	Counter	Input	Output
0	ACLOCK0	AOUT1	A0 A1	ACLOCK0 ACLOCK1	AOUT0 AOUT1
1	ACLOCK2	AOUT3	A2 A3	ACLOCK2 ACLOCK3	AOUT2 AOUT3
2	BCLOCK0	BOUT1	B0 B1	BCLOCK0 BCLOCK1	BOUT0 BOUT1
3	BCLOCK2	BOUT3	B2 B3	BCLOCK2 BCLOCK3	BOUT2 BOUT3
4	CCLOCK0	COUT1	C0 C1	CCLOCK0 CCLOCK1	COUT0 COUT1
5	CCLOCK2	COUT3	C2 C3	CCLOCK2 CCLOCK3	COUT2 COUT3
6	DCLOCK0	DOUT1	D0 D1	DCLOCK0 DCLOCK1	DCLOCK0 DCLOCK1
7	DCLOCK2	DOUT3	D2 D3	DCLOCK2 DCLOCK3	DCLOCK2 DCLOCK3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3.2 Programming

The Frequency Divider function is initialized by sending one of the following two commands to any one of the ICM's eight channels.

22H = Start 16-Bit Frequency Divider
23H = Start 32-Bit Frequency Divider

6.3.2.1 Start 16-Bit Frequency Divider

The Start 16-bit Frequency Divider command configures the specified counter to divide the frequency on its CLOCK input by the specified 16-bit divisor. This command returns a response immediately. The format is listed below:

Start 16-bit Frequency Divider Command Format

Command Word: 22H

Four Operand Bytes:

Byte 1: Counter number ——————— (0,1,2, or 3)

Byte 2: Unused

Bytes 3-4: 16-bit Divisor

6.3.2.2 Start 32-Bit Frequency Divider

The Start 32-bit Frequency Divider command configures the specified channel to divide the incoming frequency on its CLOCK input by the specified 32-bit divisor. This command returns a response immediately. The format is listed below:

Start 32-bit Frequency Divider Command Format

Command Word: 23H

Six Operand Bytes:

Bytes 1-2: Unused

Bytes 3-6: 32-bit Divisor - Must be greater than 65536

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4 TIMER

The Timer function monitors user-specified time periods. The time period is specified by declaring upon initialization of this function a four-byte operand labeled Time. The Time operand can be specified in floating point format in units of seconds or in integer format as a multiple of any one of the following units:

- one microsecond
- 10 microseconds
- 100 microseconds
- one millisecond
- 10 milliseconds
- 100 milliseconds
- one second

In no case can the Time value be greater than 858 seconds. When using the integer data format, the TIME value must be less than 65,536 units. The command terminates after the specified time period.

6.4.1 Hardware Configuration

This function is internal and no hardware connections are needed.

6.4.2 Programming

The Timer function is initialized by sending the following command to one of the ICM's eight channels:

26H = Start Timer

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.2.1 Start Timer

The Start Timer command configures the specified counter to generate the specified time period and notify the system host when it has expired. The format is listed below:

Start Timer Command Format

Command Word: 26H

Eight Operand Bytes:

Byte 1: Counter number	(0,1,2, or 3)
Byte 2: Data Format	0 = integer data format 1 to FFH = Floating Point data format
Byte 3: Units for integer format	0 = 1 us 1 = 10 us 2 = 100 us 3 = 1 ms 4 = 10 ms 5 = 100 ms 6 = 1 s
Byte 4: Unused	
Bytes 5-8: Time	A four-byte value referring to the length of the delay. This value is expressed in seconds if using the floating point data format, or as a function of the integer units chosen if using the integer data format. In no case can the Time value be greater than 858 seconds. The Time value specified for the integer format must be less than 65,536 units.

NOTE

This command will return a response (i.e., Clear the Response Flag and optionally interrupt) ONLY when the time has expired. If it has not expired, the corresponding counter on this channel cannot be used for any other commands for the duration of the Timer's period.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5 PERIODIC INTERRUPT

The Periodic Interrupt function allows a counter to generate interrupts which repeat at a specified period until the process is terminated by a stop command. The Periodic Interrupt function has the same operand block format as the Timer function, but in this case the Time operand expresses the period at which the interrupts will be issued, as opposed to a single time period for a single interrupt. The Time operand can be specified in floating point format in units of seconds or in integer format as a multiple of any of the following units:

- one microsecond
- 10 microseconds
- 100 microseconds
- one millisecond
- 10 milliseconds
- 100 milliseconds
- one second

In no case can the Time value be greater than 858 seconds. When using the integer data format, the Time value must be less than 65,536 units. This command must be terminated by a stop command (refer to Chapter 4, Section 4.9.4).

6.5.1 Hardware Configuration

This function is internal and no hardware connections are needed.

6.5.2 Programming

The Periodic Interrupt function is initialized by sending the following command to one of the ICM's eight channels:

27H = Start Periodic Interrupt

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.2.1 Start Periodic Interrupt

The Start Periodic Interrupt command configures the specified counter to generate interrupts in intervals of the time period specified until a stop command is issued. The command format is shown below:

Start Timer Command Format

Command Word: 27H

Eight Operand Bytes:

Byte 1: Counter number	(0,1,2, or 3)
Byte 2: Data Format	0 = integer data format 1 = IEEE single precision floating point format
Byte 3: Units for integer format	0 = 1 us 1 = 10 us 2 = 100 us 3 = 1 ms 4 = 10 ms 5 = 100 ms 6 = 1 s
Byte 4: Unused	
Bytes 5-8: Time	A four-byte value referring to the length of the delay period between interrupts. This value is expressed in seconds if using the floating point data format, or as a function of the integer units chosen if using the integer data format. In no case can the Time value can be greater than 858 seconds. The Time value specified for the integer format must be less than 65,536 units.

NOTE

This command will return a response (i.e., Clear the Response Flag and optionally interrupt) each time the specified time interval has expired. The corresponding counter on this channel cannot be used for any other commands for the duration of the current command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 7

PERIOD/PULSE WIDTH GENERATION AND FREQUENCY/DUTY CYCLE GENERATION

These two functions allow the user to generate a signal output with either a variable period and pulse width, or a variable frequency and duty cycle. This capability allows the user to generate the signal output by specifying the parameters which are the most appropriate for the application.

7.1 PERIOD/PULSE WIDTH GENERATION

The period/pulse width generation function produces a signal output of variable period and variable pulse width. The period represents the time it takes for a waveform to complete one full cycle, and the pulse width is typically defined as the measure of the active duration of the pulse. Figure 7-1 shows the relationship between the period and the pulse width for a single cycle of an output signal.

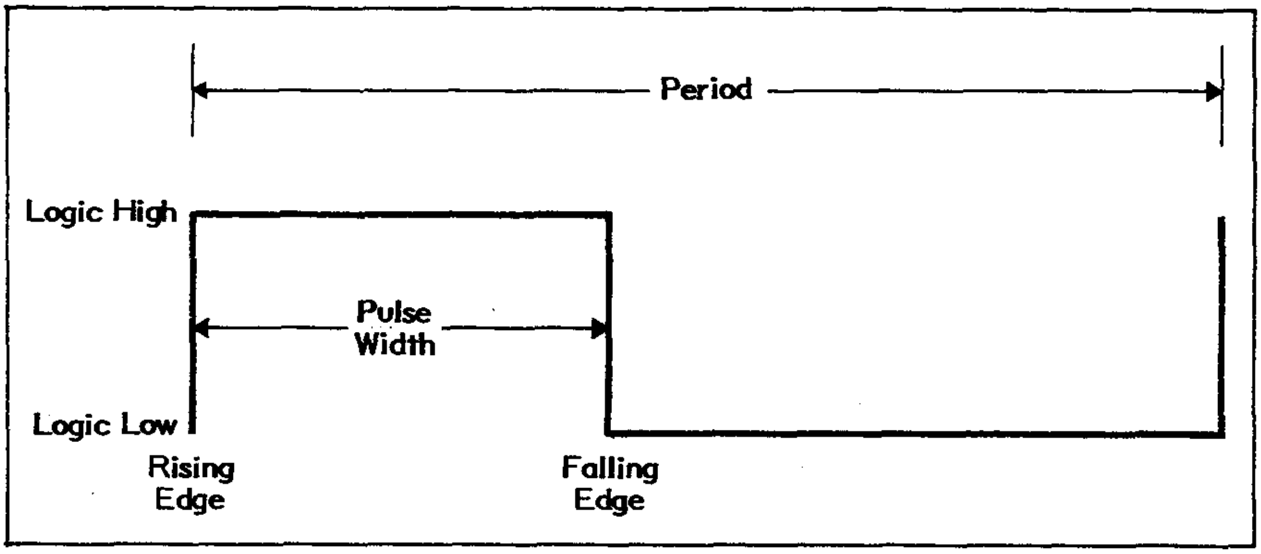


Figure 7-1. Period/Pulse Width Operand Relationship

The user provides four operands when initializing the Period/Pulse Width Generation function: Counter number, Data format, Period and Pulse Width.

A single counter is all that is required to operate the Period/Pulse Width function. This function does not require a CLOCK input signal, since the Period/Pulse width is internally generated.

The Period operand is a four-byte value which is expressed in units of microseconds and has a range of from 10 us to 10 sec. The unit value can be specified in seconds by

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

using the IEEE single standard floating point format as opposed to the integer format. The Pulse Width operand is a four-byte value which is also expressed in units of microseconds and has a usable range of from 10 us to 10 sec. The Pulse Width can be expressed in units of seconds if the IEEE single floating point format is used.

For example, an output signal that has a period of 50 us and a pulse width of 25 us can be generated by writing the Start Period/Pulse width generation command and by specifying 32H (50 decimal) as the period operand and 19H (25 decimal) as the pulse width operand. (For the complete programming sequence to utilize the Period/Pulse Width generator, refer to Section 7.3).

7.2 FREQUENCY/DUTY CYCLE GENERATION

The Frequency/Duty Cycle Generation function outputs a signal of variable frequency and duty cycle. The duty cycle represents the percentage of the complete cycle period that the output of the channel is active (i.e., in the "ON" state). Figure 7-2 shows the relationship between the frequency and the duty cycle for a single cycle of an output signal.

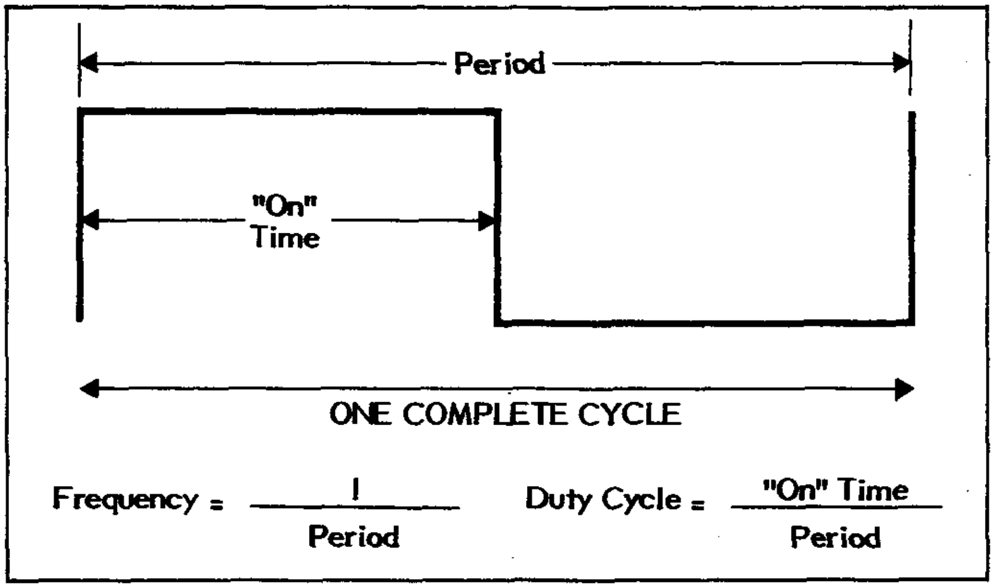


Figure 7-2. Frequency/Duty Cycle Operand Relationship

The user provides four operands when initializing the Frequency/Duty Cycle Generation function: Counter number, Data format, Frequency and % On.

A single counter is all that is required to operate the Frequency/Duty Cycle Generation function. This function does not require a CLOCK input, since the Frequency/Duty Cycle is internally generated.

The Frequency operand is a four-byte value that is expressed in units of .01Hz and has a range of 1Hz to 10000Hz. For example, an output frequency of 100Hz may be obtained by using a value of 2710H (i.e., 100Hz * 1000 = 10000 decimal, which = 2710H) for the Frequency operand. The frequency can also be expressed in IEEE single

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

standard floating point format (explained in Chapter 4, Section 4.11), in units of 1Hz, with a range of 1.00 - 100000.0.

The % On operand is a four-byte integer that is expressed in units of .01 percent (e.g., 30% would equal 30*100 or BB8H). The % On can also be expressed in the IEEE single precision floating point format. A % On value of 0 will produce a 50% duty cycle. The operand has a range of 0.01% - 99.99%.

7.3 HARDWARE CONFIGURATION

The Period/Pulse Width and the Frequency/Duty Cycle functions can be implemented on any of the ICM's 16 counters. The output of both functions are available on the output of the counter they are running on.

For example, if a function is running on counter A0, the output is connected to the signal on JK1, labeled AOUT0. The connections for implementation on each of the 16 counters are listed in Table 7-1 below.

Table 7-1. Frequency/Duty Cycle and Period/Pulse Width Connection

Channel	Counter	Output Pin
0	A0 A1	AOUT0 AOUT1
1	A2 A3	AOUT2 AOUT3
2	B0 B1	BOUT0 BOUT1
3	B2 B3	BOUT2 BOUT3
4	C0 C1	COUT0 COUT1
5	C2 C3	COUT2 COUT3
6	D0 D1	DOUT0 DOUT1
7	D2 D3	DOUT2 DOUT3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.4 PROGRAMMING PERIOD/PULSE WIDTH

The Period/Pulse Width function is initialized by sending the following command to any of the ICM's eight channels:

33H = Start Period/Pulse Width Generator

The period and the pulse width of the Period/Pulse Width function can be changed without disturbing the other command variables by using the following two commands:

34H = Change Pulse Width
35H = Change Period

7.4.1 Start Period/Pulse Width

The Start Period/Pulse Width command configures the specified counter to generate a signal composed of the period and pulse width specified in the command. The format is listed below:

Start Period/Pulse Width Command Format

Command Word: 33H

Ten Operand Bytes:

Byte 1: Counter number	(0,1,2, or 3)
Byte 2: Data format	(0 = integer, 1 - FFH = IEEE single precision floating point data format)
Bytes 3-6: Period Byte 3: MSB Byte 4: LSB+2 Byte 5: LSB+1 Byte 6: LSB	This is a four-byte value that indicates the period to be generated.
Bytes 7-10: Pulse Width Byte 7: MSB Byte 8: LSB+2 Byte 9: LSB+1 Byte 10: LSB	This is a four-byte value that specifies the pulse width of the generated output.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.4.2 Change Pulse Width

The Change Pulse Width command changes only the prescribed pulse width of the output signal being generated. The change occurs immediately with little or no break in the output waveform. This command is only used to modify the pulse width of a signal generated by the Start Period/Pulse Width command (refer to Section 7.4.1). The format is listed below:

Change Pulse Width Command Format

Command Word: 34H

Ten Operand Bytes:

Byte 1: Counter number ———— (0,1,2, or 3)

Byte 2: Data format ———— (0 = integer, 1 - FFH = IEEE single precision floating point data format)

Bytes 3-6: Unused

Bytes 7-10: Pulse Width ———— This is a four-byte value that specifies the width of the pulse.
Byte 7: MSB
Byte 8: LSB+2
Byte 9: LSB+1
Byte 10: LSB

7.4.3 Change Period

The Change Period command changes only the specified period for the output signal being generated. This command is only used to modify the period of a signal generated by the Start Period/Pulse Width command (refer to Section 7.4.1). The format is listed below:

Change Period Command Format

Command Word: 35H

Six Operand Bytes:

Byte 1: Counter number ———— (0,1,2, or 3)

Byte 2: Data format ———— (0 = integer, 1 - FFH = IEEE single precision floating point data format)

Bytes 3-6: Period ———— This is a four-byte value that indicates the period.
Byte 3: MSB
Byte 4: LSB+2
Byte 5: LSB+1
Byte 6: LSB

Bytes 7-10: Unused

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5 PROGRAMMING FREQUENCY/DUTY CYCLE

The Frequency/Duty Cycle function is initialized by sending the following command to any of the ICM's eight channels:

30H = Start Frequency/Duty Cycle Generator

The duty cycle and frequency of the Frequency/Duty Cycle function can be changed without disturbing the other aspects by the following two commands:

31H = Change Duty Cycle
32H = Change Frequency

7.5.1 Start Frequency/Duty Cycle Generator

The Start Frequency/Duty Cycle Generator command configures the specified counter to generate a signal composed of the frequency and duty cycle specified in the command. The format is listed below:

Start Frequency/Duty Cycle Generator Command Format

Command Word: 30H

Ten Operand Bytes:

Byte 1: Counter number ———— (0,1,2, or 3)

Byte 2: Data format ———— 0 specifies integer data format.
1 - FFH specifies IEEE single precision floating point data format.

Bytes 3-6: Frequency ———— This is a four-byte value that indicates the frequency to be generated.
Byte 3: MSB
Byte 4: LSB+2
Byte 5: LSB+1
Byte 6: LSB

Bytes 7-10: % On ———— This is a four-byte value that specifies the percentage of time the output signal will be on.
Byte 7: MSB
Byte 8: LSB+2
Byte 9: LSB+1
Byte 10: LSB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.2 Change Duty Cycle

The Change Duty Cycle command changes only the duty cycle of the frequency/duty cycle signal being generated. The change occurs immediately with little or no break in the output waveform. This command is only used to modify the duty cycle of a signal generated by the Start Frequency/Duty Cycle Generator command (refer to Section 7.5.1). The format is listed below:

Change Duty Cycle Command Format

Command Word: 31H

Ten Operand Bytes:

Byte 1: Counter number ———— (0,1,2, or 3)

Byte 2: Data format ———— (0 = integer, 1 - FFH = IEEE single precision floating point format)

Bytes 3-6: Unused

Bytes 7-10: % On ———— This is a four-byte value that specifies the percentage of on time.
- Byte 7: MSB
- Byte 8: LSB+2
- Byte 9: LSB+1
- Byte 10: LSB

7.5.3 Change Frequency

The Change Frequency command changes only the frequency component of the frequency/duty cycle signal being generated. This command is only used to modify the frequency of a signal generated by the Start Frequency/Duty Cycle Generator command (refer to Section 7.5.1). The format is listed below:

Change Frequency Command Format

Command Word: 32H

Six Operand Bytes:

Byte 1: Counter number ———— (0,1,2, or 3)

Byte 2: Data format ———— (0 = integer, 1 - FFH = IEEE single precision floating point format)

Bytes 3-6: Frequency ———— This is a four-byte value indicating frequency.
- Byte 3: MSB
- Byte 4: LSB+2
- Byte 5: LSB+1
- Byte 6: LSB

Bytes 7-10: Unused

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 8

PERIOD MEASUREMENT

The Period Measurement function measures the time period of an input time interval. It can optionally generate an interrupt whenever the measured period falls outside of the user-specified range.

A typical industrial application for the Period Measurement function would be fluid flow monitoring via turbine metering. Each pulse generated by the turbine meter would represent a fixed amount of fluid flow and the time interval between pulses would indicate flow rate. An unacceptable flow rate would cause the ICM to generate an error code to alert the user that the limit was exceeded.

The ICM performs the period measurement by counting cycles of an on-board reference frequency during the input time interval (waveform). This on-board reference frequency can be thought of as a sampling frequency: the higher the on-board reference frequency, the more often the input waveform will be sampled, resulting in a higher resolution period measurement.

The Period Measurement function can operate in one of four modes: Single-interval labeled Position Measurement I (PMI), Multiple-interval labeled Position Measurement II (PMII), Multiple-interval labeled Position Measurement III (PMIII), or Single-interval labeled Period/Pulse Width Measurement.

The PM I and Period/Pulse Width modes compute the period after evaluating one time interval (waveform) of the input signal. In addition, the Period/Pulse Width mode will return the pulse width measurement for the signal in question.

The PM II and PM III modes compute the period of the time interval after measuring a user specified number of time intervals. The measurement is then averaged to find the time period of one time interval.

The PM II and PM III modes are a more accurate method of measurement because they allow the readings to be averaged over several time intervals. However, when using the PM I and Period/Pulse Width Measurement modes, several readings can be taken and averaged by the user, resulting in similar accuracy.

Upon initialization of the Period Measurement function, the user is given the option of selecting whether the measured period will be returned in integer or floating point format (explained in Chapter 4, Section 4.11). The integer format allows the user to specify different units upon initialization. The floating point format expresses the measurements in units of seconds.

The operands that will be specified upon initialization will depend upon which mode, Single (PM I, Period/Pulse Width Measurement) or Multiple (PM II, PM III), the period measurement function is operating in.

Upon initialization of the PM I or the Period/Pulse Width Measurement mode, the user specifies three and possibly five operands:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- the on-board reference frequency to be used during the period measurement,
- data format,
- units for integer data,
- optional high limit, and
- optional low limit if the user chooses to specify a range of acceptable input periods.

Upon initialization of the PM II or PM III mode, the user specifies four and possibly six operands:

- the on-board reference frequency to be used during the period measurement,
- data format,
- units for integer data,
- the number of time intervals over which to measure,
- optional high limit, and
- optional low limit that specify a range of acceptable input periods.

The operand selecting the on-board reference frequency is a one-byte value where the number chosen represents one of the five following frequencies:

- 0 specifies 5MHz
- 1 specifies 312.5KHz
- 2 specifies 19531.25Hz
- 3 specifies 1220.7Hz
- 4 specifies 76.29Hz

The frequency operand for period measurement can be selected by using the following formula.

Choose a Frequency "F" from the five possible choices (shown above) such that:

$$F < \frac{\left(\dfrac{65536}{\text{\# of intervals}}\right)}{\text{longest expected period in seconds}}$$

If the data format operand is zero the data format is integer. If it is 1-FFH, the data format is IEEE single precision floating point (refer to Section 4.11).

The units operand for the integer data format selects units in the following way:

0 specifies units of 1 microsecond
1 specifies units of 10 microseconds
2 specifies units of 100 microseconds
3 specifies units of 1 millisecond
4 specifies units of 10 milliseconds
5 specifies units of 100 milliseconds
6 specifies units of 1 second

(* Remember that the floating point data format always has units of 1 second.)

The operand selecting the number of intervals over which to average the measurement is a two-byte integer that can range from 2 to 65535 (0FFFFH), and is in units of input waveforms.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.1 HARDWARE CONFIGURATION

PM I, PM II, PM III, and Period/Pulse Width measurement all require one channel (two counters) to operate. It is important to note however, that Period/Pulse Width measurement and PM II, due to their advanced measuring capabilities, can operate only on even channels (i.e., 0, 2, 4, and 6). PM I and PM III are able to operate on any one of the eight channels. PM III requires the user to make an external connection between two of the pins on the front panel connectors for each channel of FM III to be activated.

The signal to be measured is connected to the CLOCK input of the selected channel's even-numbered counter. For example, if channel 0 is selected, the input signal is connected to ACLOCK0 on connector JK1. The connections for Period Measurement on each channel are shown in Table 8-1 below.

Table 8-1. Period Measurement Connection

Channel			Period/Pulse Width Measurement	Input Pin
PM I	PM II	PM III		
0	0	0	0	ACLOCK0
1		1		ACLOCK2
2	2	2	2	BCLOCK0
3		3		BCLOCK2
4	4	4	4	CCLOCK0
5		5		CCLOCK2
6	6	6	6	DCLOCK0
7		7		DCLOCK2

For each channel of PM III which is used, the output of the first counter of that channel (counter #0 or #2) must be connected to the gate of the second counter of that channel (counter #1 or #3). Table 8-2 shows the connection to be made for each channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 8-2. PM III External Connections

Channel	Connection
0	AOUT0 (pin 18 of JK1) TO AGATE1 (pin 8 of JK1)
1	AOUT2 (pin 22 of JK1) TO AGATE3 (pin 16 of JK1)
2	BOUT0 (pin 42 of JK1) TO BGATE1 (pin 32 of JK1)
3	BOUT2 (pin 46 of JK1) TO BGATE3 (pin 40 of JK1)
4	COUT0 (pin 18 of JK2) TO CGATE1 (pin 8 of JK2)
5	COUT2 (pin 22 of JK2) TO CGATE3 (pin 16 of JK2)
6	DOUT0 (pin 42 of JK2) TO DGATE1 (pin 32 of JK2)
7	DOUT2 (pin 46 of JK2) TO DGATE3 (pin 40 of JK2)

8.2 PROGRAMMING

The Period Measurement function is initialized by sending the command:

60H = Start PM I, with Optional Limit Check

to any of the ICM's eight channels or by sending the command:

61H = Start PM II, with Optional Limit Check

to any of the ICM's even channels or by sending the command:

62H = Start PM III, with Optional Limit Check

to any of the ICM's eight channels.

The Period/Pulse Width measurement function is initialized by sending the command:

80H = Start Period/Pulse Width Measurement, with Optional Limit Check

to any of the ICM's even channels.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2.1 **Start PM I, with Optional Limit Check**

The Start PM I, with Optional Limit Check command, configures the specified channel for Period Measurement in the PM I mode. The channel measures the period of a single interval of the signal on its CLOCK 0 or CLOCK 2 input. When limit checking is used, this command will run continuously and not generate a response until terminated by a STOP command, or until a specified limit is reached. The format is listed below:

Start PM I, with Optional Limit Check Command Format

Command Word: 60H

16 Operand Bytes:

Byte 1: On-board reference frequency selection	0 = 5MHz 1 = 312.5KHz 2 = 19531.25Hz 3 = 1220.7Hz 4 = 76.29Hz
Byte 2: Data format	(0 = integer, 1 - FFH = IEEE single precision floating point format)
Byte 3: Units for integer data	0 = 1 usec 1 = 10 usec 2 = 100 usec 3 = 1 msec 4 = 10 msec 5 = 100 msec 6 = 1 sec
Byte 4: Unused	
Bytes 5-8: Returned Period Measurement	Contains the 4-byte value representing the Period in the format and units selected by the data format and units operands.
Bytes 9-12: Optional High Limit	These four bytes specify the upper period measured in the format selected by the data format operand. If this limit is exceeded, the period measurement will terminate with the response word 0AH (high limit exceeded). If this operand equals zero, then there is no high limit check.
Bytes 13-16: Optional Low Limit	These four bytes specify the lowest period to be measured in the format selected by the data format operand. If the input period falls below this limit, the period measurement will terminate with the response word 0CH (low limit exceeded). If this operand equals zero, then there is no low limit check.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2.2 Start PM II, with Optional Limit Check

The Start PM II, with Optional Limit Check command, configures the specified channel for Period Measurement in the PM II mode. The channel then calculates the average period of a specified number of intervals on its CLOCK 0 input. When limit checking is used, this command will run continuously and not generate a response until terminated by a STOP command, or until a specified limit is reached. The format is listed below:

Start PM II, with Optional Limit Check Command Format

Command Word: 61H

18 Operand Bytes:

Byte 1: Frequency Selection — 0 = 5MHz
1 = 312.5KHz
2 = 19531.25Hz
3 = 1220.7Hz
4 = 76.29Hz

Byte 2: Data format — (0 = integer, 1 - FFH = IEEE single precision floating point format)

Byte 3: Units for integer data — 0 = 1 usec, 1 = 10 usec, 2 = 100 usec, 3 = 1 msec, 4 = 10 msec, 5 = 100 msec, 6 = 1 sec

Byte 4: Unused

Bytes 5-8: Returned Period Measurement — Contains the 4-byte value representing the Period in the format selected by the data format operand.

Bytes 9-12: Optional High Limit — These four bytes specify the upper period to be be measured in the format selected by the data format operand. If this limit is exceeded, the period measurement will terminate with the response word 0AH (high limit exceeded). If this operand equals zero, then there is no high limit check.

Bytes 13-16: Optional Low Limit — These four bytes specify the lowest period to be measured in the format selected by the data format operand. If the input period falls below this limit, the period measurement will terminate with the response 0CH (low limit exceeded). If this operand equals zero, then there is no low limit check.

Bytes 17-18: Number of intervals over which to measure (2-65535) — These two bytes specify the number of input time intervals over which the period measurement will be averaged.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2.3 Start PM III, with Optional Limit Check

The Start PMIII, with Optional Limit Check command configures the specified channel for Period Measurement in the PMIII mode. The PMIII function operates identically and provides the same options as the PMII function. PMIII differs from PMII in that PMIII can be implemented in a single channel, allowing eight simultaneous channels of PMIII, whereas each PMII operation requires two channels, limiting it to four simultaneous measurements. The price paid for the use of all eight channels is that an additional external connection must be made when implementing PM III (refer to Table 8-2). When limit checking is used, this command will run continuously and not generate a response until terminated by a STOP command, or until a specified limit is reached. The format for the Start PM III command is listed below:

Start PM III, with Optional Limit Check Command Format

Command Word: 62H

18 Operand Bytes:

Byte 1: Frequency Selection	0 = 5MHz 1 = 312.5KHz 2 = 19531.25Hz 3 = 1220.7Hz 4 = 76.29Hz
Byte 2: Data format	(0 = integer, 1 - FFH = IEEE single precision floating point format)
Byte 3: Units for integer data	0 = 1 usec 1 = 10 usec 2 = 100 usec 3 = 1 msec 4 = 10 msec 5 = 100 msec 6 = 1 sec
Byte 4: Unused	
Bytes 5-8: Returned Period Measurement	Contains the 4-byte value representing the Period in the format selected by the data format operand.
Bytes 9-12: Optional High Limit	These four bytes specify the upper period to be measured in the format selected by the data format operand. If this limit is exceeded, the period measurement will terminate with the response word 0AH (high limit exceeded). If this operand equals zero, then there is no high limit check.
Bytes 13-16: Optional Low Limit	These four bytes specify the lowest period to be measured in the format selected by the data format operand. If the input period falls below this limit, the period measurement will terminate with the response word 0CH (low limit exceeded). If this operand equals zero, then there is no low limit check.
Bytes 17-18: Number of intervals over which to measure (2-65535)	These two bytes specify the number of input time intervals over which the period measurement will be averaged.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2.4 PM I - PM III Measurement Delays

When using the period measurement functions PM I, PM II, and PM III, there is a delay associated with the time each function takes to return a measurement. The period measurement functions take a finite amount of time to return a measurement. With short periods this is not a problem, but with longer periods (greater than 1 second) it could become noticeable. The maximum (worst case) delays are:

- PM I Delay = 415 us + 2 * input period
- PM II Delay = 505 us + 2 * (#intervals * input period)
- PM III Delay = 450 us + 3 * input period + #intervals * input period

Note that the formula above which specifies the delay for PMI implies that for input frequencies of less than 2.4KHz the measurements will be taken on every other period.

Note that the formula above which specifies the delay for PMII implies that for input frequencies of less than 1.9KHz the measurements will be taken on every other interval.

Note that the formula above which specifies the delay for PMIII implies that for input frequencies of less than 2.3KHz three periods will be skipped before the interval measurement will be taken.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2.5 Start Period/Pulse Width Measurement, with Optional Limit Check

The Start Period/Pulse Width measurement, with Optional Limit Check command configures the specified channel to measure both the period and the pulse width of an input signal. This command, like PM II, can only be used on the even-numbered channels, and while it is active the next higher odd channel will be disabled. The signal to be measured must be connected to the GATE0 input of the specified channel. Optional limit checking only applies to pulse width measurement in this function. When limit checking is used, this command will run continuously and not generate a response wuntil terminated by a STOP command, or until a specified limit is reached. The format for the command is shown below:

Start Period/Pulse Width Measurement,
with Optional Limit Check Command Format

Command Word: 80H

20 Operand Bytes:

Byte 1: On-board reference frequency selection ——— 0 = 5MHz
1 = 312.5KHz
2 = 19531.25Hz
3 = 1220.7Hz
4 = 76.29Hz

Byte 2: Data format ——— (0 = integer, 1 - FFH = IEEE single precision floating point format)

Byte 3: Units for integer data ——— 0 = 1 usec, 1 = 10 usec, 2 = 100 usec, 3 = 1 msec, 4 = 10 msec, 5 = 100 msec, 6 = 1 sec

Byte 4: Unused

Bytes 5-8: Returned Pulse Width Measurement ——— Contains the 4-byte value representing the pulse width in the format and units selected by the data format and units operand.

Bytes 9-12: Optional High Limit ——— These four bytes specify the upper pulse width to be measured in the format selected by the data format operand. If this limit is exceeded, the period/pulse width measurement will terminate with the response word 0AH (high limit exceeded). If this operand equals zero, then there is no high limit check.

Bytes 13-16: Optional Low Limit ——— These four bytes specify the lowest pulse width to be measured in the format selected by the data format operand. If the input pulse width falls below this limit, the period/pulse width measurement will terminate with the response word 0CH (low limit exceeded). If this operand equals zero, then there is no low limit check.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bytes 17-20: Returned Period Measurement — Contains the four-byte value representing the period measured, in the format and units selected by the data format and units operand.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 9

FREQUENCY MEASUREMENT

9.1 INTRODUCTION

The Frequency Measurement function measures the frequency of an input signal and optionally generates an interrupt if it is not within a user-specified range of frequencies. A typical application for this function would be monitoring the speed of a rotating shaft where a shaft encoder is generating an output frequency directly proportional to the shaft's rotation speed.

The Frequency Measurement function is similar but opposite to the Period Measurement function, because it measures an input frequency based on the number of pulses of the signal received during a known measurement period.

Upon initialization of the function, the user will specify the measurement period, which is labeled the Period operand. The Period operand should be selected based on the expected input frequency, and the constraints on time and accuracy presented by the application.

The Frequency Measurement function operates in one of three modes: Frequency Measurement I (FM I), Frequency Measurement II (FM II), and Frequency Measurement III (FM III). The main difference among modes is the selection range for the measurement period, and the fact that PM I and PM III can operate on all eight channels, while PM II is designed to operate on the even-numbered channels only.

For the Frequency Measurement I mode the choices for the period operand (Byte 1) are:

0 = 51.2 microseconds

1 = 819.2 microseconds

2 = 13.1 milliseconds

For the Frequency Measurement II and Frequency Measurement III modes the choices for the period operand (Byte 1) are:

0 = 100 microseconds

1 = 1 millisecond

2 = 10 milliseconds

3 = 100 milliseconds

4 = 1 second

5 = 10 seconds

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The period operand for frequency measurement is selected by using the following formula:

Choose a Period "P" such that:

$$P < \frac{65536}{\text{highest expected frequency in hertz}}$$

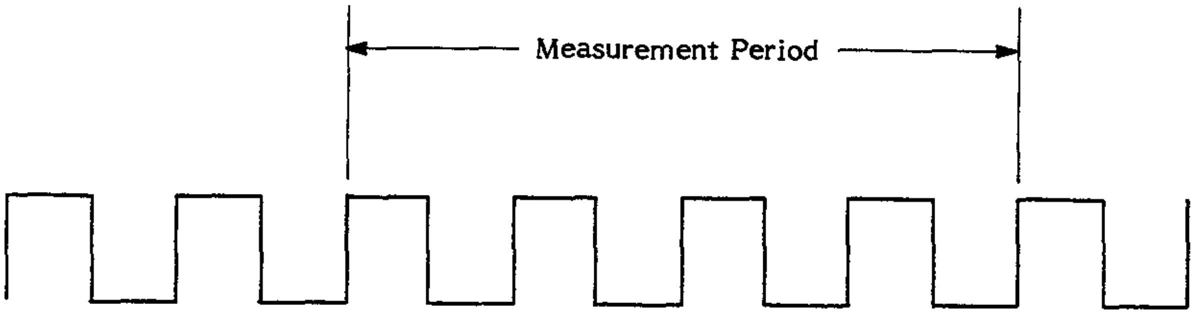


Figure 9-1. Typical Measurement Period/Input Frequency Relationship

The remaining operands to be specified upon initialization are described below.

The data format operand (Byte 2) selects either Integer or IEEE single precision floating point format (see Section 4.11 for a description of the IEEE format). The selection is made by specifying the operand as:

- zero (0) for an integer data format
- 1-FFH for the floating point format

The units operand applies only in the Integer data mode. The integer quantity can be represented in units as follows:

0 = 1 Hz

1 = .1 Hz

2 = .01 Hz

3 = .001 Hz

4 = .0001 Hz

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.2 HARDWARE CONFIGURATION

As previously mentioned, FM I and FM III can be used on any of the eight available channels, while FM II is designed to be used on even-numbered channels only (i.e., FM II use is restricted to four possible channels). FM II and FM III allow more flexibility and range than FMI I, but FM II is of course limited to four channels, and FM III requires the user to make an external connection between two of the pins on the front panel connectors for each channel of FM III to be activated.

In order to use any of the three frequency measurement modes, a connection must be made at the CLOCK input of the relevant channel's even-numbered counter. See Table 9-1 for configuration.

Table 9-1. Frequency Measurement Connection

Channel			Input Pin
FM I	FM II	FM III	
0	0	0	ACLOCK0
1		1	ACLOCK2
2	2	2	BCLOCK0
3		3	BCLOCK2
4	4	4	CCLOCK0
5		5	CCLOCK2
6	6	6	DCLOCK0
7		7	DCLOCK2

For each channel of FM III which is used, the output of the first counter of that channel (counter #0 or #2) must be connected to the gate of the second counter of that channel (counter #1 or #3). Table 9-2 shows the connection to be made for each channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 9-2. FM III External Connections

Channel	Connection
0	AOUT0 (pin 18 of JK1) TO AGATE1 (pin 8 of JK1)
1	AOUT2 (pin 22 of JK1) TO AGATE3 (pin 16 of JK1)
2	BOUT0 (pin 42 of JK1) TO BGATE1 (pin 32 of JK1)
3	BOUT2 (pin 46 of JK1) TO BGATE3 (pin 40 of JK1)
4	COUT0 (pin 18 of JK2) TO CGATE1 (pin 8 of JK2)
5	COUT2 (pin 22 of JK2) TO CGATE3 (pin 16 of JK2)
6	DOUT0 (pin 42 of JK2) TO DGATE1 (pin 32 of JK2)
7	DOUT2 (pin 46 of JK2) TO DGATE3 (pin 40 of JK2)

9.3 PROGRAMMING

The Frequency Measurement function is initialized by sending the command:

50H = Start FM I, with Optional Limit Check

to any of the ICM's eight channels, or by sending the command:

51H = Start FM II, with Optional Limit Check

to any of the ICM's even channels, or by sending the command:.

52H = Start FM III, with Optional Limit Check

to any of the ICM's eight channels.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3.1 Start FM I, with Optional Limit Check

The Start FM I, with Optional Limit Check command, configures the specified channel for frequency measurement in the FM I mode. The channel measures the frequency of the signal on the CLOCK input of its lower counter. When limit checking is used, this command will run continuously and not generate a completion response until terminated by a STOP command or until a limit is reached. The format is listed below:

Start FM I, with Optional Limit Check Command Format

Command Word: 50H

16 Operand Bytes:

Byte 1: Period selection (Refer to formula on page 9-2 for period operand selection)	0 = 51.2 usec 1 = 819.2 usec 2 = 13.1 msec
Byte 2: Data format	0 = integer data format 1 - FFH = IEEE single precision floating point format
Byte 3: Units for integer data format	0 = 1 Hertz 1 = .1 Hertz 2 = .01 Hertz 3 = .001 Hertz 4 = .0001 Hertz
Byte 4: Unused	
Bytes 5-8: Returned Frequency Measurement	The value of these four bytes is the frequency calculated by the function, expressed in the data format and units specified by the data format and units operands.
Bytes 9-12: Optional High Limit	The value of these four bytes specifies the upper frequency to be measured in the data format specified by the data format operand. If this limit is exceeded, the frequency measurement will terminate with the response word 0AH (high limit exceeded). If this operand equals zero, then there is no high limit check.
Bytes 13-16: Optional Low Limit	The value of these four bytes specifies the lowest frequency to be measured in the data format specified by the data format operand. If this limit is exceeded, the frequency measurement will terminate with the response word 0CH (low limit exceeded). If this operand equals zero, then there is not low limit check.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3.2 Start FM II, with Optional Limit Check

The Start FM II, with Optional Limit Check command, configures the specified channel (must be an even channel) to measure the frequency of the signal on the CLOCK input of its lower counter. When limit checking is used, this command will run continuously, not generating a Completion response until terminated by a STOP command or until a limit is reached. The format is listed below:

Start FM II, with Optional Limit Check Command Format

Command Word: 51H

16 Operand Bytes:

Byte 1: Period selection	0 = 100 usec; 1 = 1 msec; 2 = 10 msec; 3 = 100 msec; 4 = 1 sec; 5 = 10 sec
Byte 2: Data format	0 = integer data format 1 - FFH = IEEE single precision floating point format
Byte 3: Units for integer data format	0 = 1 Hertz 1 = .1 Hertz 2 = .01 Hertz 3 = .001 Hertz 4 = .0001 Hertz
Byte 4: Unused	
Bytes 5-8: Returned Frequency Measurement	The value of these four bytes is the frequency calculated by the function, expressed in the data format and units specified by the data format and units operands.
Bytes 9-12: Optional High Limit	The value of these four bytes specifies the upper frequency to be measured in the data format and units specified by the data format and units operands. If this limit is exceeded, the frequency measurement will terminate with the response word 0AH (high limit exceeded). If this operand equals zero, then there is no high limit check.
Bytes 13-16: Optional Low Limit	The value of these four bytes specifies the lowest frequency to be measured in the data format and units specified by the data format and units operands. If this limit is exceeded, the frequency measurement will terminate with the response word 0CH (low limit exceeded). If this operand equals zero, then there is no low limit check.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3.3 Start FM III, with Optional Limit Check

The Start FM III, with Optional Limit Check command, configures the specified channel for frequency measurement based on the time period selected in the first Operand byte of the command block. The channel measures the signal on the CLOCK input of its lower counter. When limit checking is used, this command will run continuously and will not generate a response until it is terminated by a STOP command, or until a specified limit is reached. The format is listed below:

Start FM III, with Optional Limit Check Command Format

Command Word: 52H

16 Operand Bytes:

Byte 1: Period selection (Refer to formula on page 9-2 for period operand selection)	0 = 100 usec 1 = 1 msec 2 = 10 msec 3 = 100 msec 4 = 1 sec 5 = 10 sec
Byte 2: Data format	0 = integer data format 1 - FFH = IEEE single precision floating point format
Byte 3: Units for integer data format	0 = 1 Hertz 1 = .1 Hertz 2 = .01 Hertz 3 = .001 Hertz 4 = .0001 Hertz
Byte 4: Unused	
Bytes 5-8: Returned Frequency Measurement	The value of these four bytes is the frequency calculated by the function, expressed in the data format and in units specified by the data format and units operands.
Bytes 9-12: Optional High Limit	The value of these four bytes specifies the highest frequency that the function can measure in the data format and units specified by the data format and units operands.
Bytes 13-16: Optional Low Limit	The value of these four bytes specifies the lowest frequency that the function can measure in the data format and units specified by the data format and units operands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 10

POSITION MEASUREMENT

10.1 INTRODUCTION

The Position Measurement function works in conjunction with a Quadrature Position Transducer to determine the exact position of a rotating device (e.g., stepper motor).

A Quadrature Position Transducer generates three signals: SIN, COS and INDEX. The Position Measurement function uses the SIN and COS signals to calculate the number of steps the rotating device makes in the clockwise or counter-clockwise direction (see Figure 10-1 below).

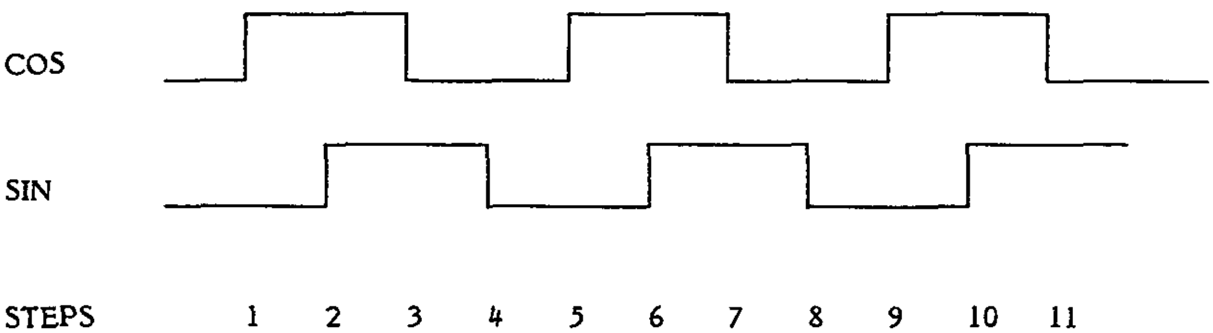


Figure 10-1. SIN, COS, and Step Relationship

Position Measurement can be performed in either of two modes: Position Measurement I or Position Measurement II.

The Position Measurement I mode requires only the SIN and COS signals to operate. It can be implemented on any of the ICM's eight channels.

The Position Measurement I function provides the following information:

- The Direction of Steps from Reference Position is a one-byte operand that indicates whether the Number of Steps from Reference Position is clockwise (byte=0) or counter-clockwise (byte≠0) with respect to the position at the time the command was first started (i.e., the Reference Position).

- The Number of Steps from Reference Position is a two-byte operand that indicates the number of steps the rotating device being monitored has traveled from the Reference Position.

The Position Measurement II function requires the SIN, COS and INDEX signals to operate. It uses the INDEX pulse as a reference signal to count the number of full rotations of the device, so the INDEX signal must be provided. (Note that not all transducers produce an INDEX signal.) The Position Measurement II function requires two channels to operate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Position Measurement II function provides the following information:

- The Direction of Steps from Index Position is a one-byte operand that indicates whether the Number of Steps from Index Position represents a position that is clockwise (byte=0) or counter-clockwise (byte≠0) with respect to the Index Position (i.e., the position at the time the last index pulse was received).

- The Direction of Index Pulses from Reference Position is a one-byte operand that indicates whether the Number of Index Pulses from Reference Position represents a position that is clockwise (byte=0) or counter-clockwise (byte≠0) with respect to the Reference Position (i.e., the position at the time the command was first started).

- The Number of Steps from Index Position is a two-byte operand that indicates the number of steps the rotating device being monitored has traveled from the last index pulse (i.e., the Index Position).

- The Number of Index Pulses from Reference Position is a two-byte operand that indicates the number of Index Positions the device being monitored has passed since the Position Measurement function was initialized (i.e., since the Reference Position). This operand can increase or decrease as the rotating device reverses direction and passes the Index Position. It is therefore a measure of net rotation.

Figure 10-2 below shows a rotating device being tracked by an eight-step transducer. This illustrates how the position parameters described above represent a typical 4 state movement. Table 10-1 shows the value of the parameters as calculated by both the Position Measurement I and Position Measurement II functions according to each state in Figure 10-2.

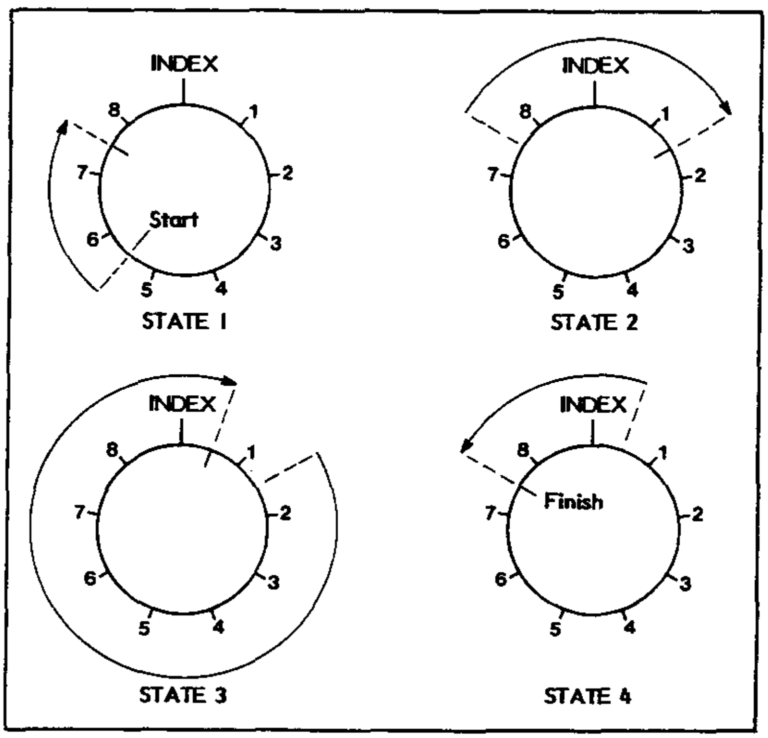


Figure 10-2. Four Movements of a Rotating Device With Eight Step Resolution

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 10-1. Parameter Values

STATE	Direction of Steps from Reference Position	Direction of Steps from Index Position	Direction of Index Pulses from Reference Position	Number of Steps from Reference Position	Number of Steps from Index Position	Number of Index Pulses from Reference Position
	PM I	PM II	PM II	PM I	PM II	PM II
1	0 CW	0 CW	0 CW	2	2	0
2	0 CW	0 CW	0 CW	4	1	1
3	0 CW	0 CW	0 CW	11	0	2
4	0 CW	FF CCW	0 CW	10	1	1

While examining both Figure 10-2 showing the device rotation and Table 10-1 showing the effect of this rotation on the Position Measurement parameters, key points in each state are now discussed.

In State 1, position 5½ is established as the Reference Position because it is the place where rotation is first detected after the Position Measurement function is initialized.

In State 2 the first index pulse is generated as the device rotates past the Index Position. In Position Measurement II this resets the step count to zero and counting starts over, while the Position Measurement I function, which ignores the index pulse, continues to count all steps that have been taken from the initial Reference Position. The first index pulse is also recorded if the mode is Position Measurement II (see Table 10-1).

In State 3 the Position Measurement I mode adds 7 steps to its total whereas the Position Measurement II mode has its step total reset to zero as the last part of the rotation corsses the index position.

Up until State 4 all three direction indicators have shown clockwise rotation (see Table 10-1). This now changes as the device reverses rotation moving one step backward. The rotation from reference parameters still show clockwise rotation because the device's net rotation is still clockwise. However, the Direction of Index pulses from Reference Position parameter now indicates a counter-clockwise rotation because the index pulse was generated while traveling in a counter-clockwise direction.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10.2 HARDWARE CONFIGURATION

The channel selected for the Position Measurement I function is connected to the Quadrature Position transducer in the following manner:

1. The SIN input is connected to the CLOCK input of the even-numbered counter.

2. The COS input is connected to the CLOCK input of the odd-numbered counter.

The Position Measurement I configuration for each channel is shown in Table 10-2 below.

Table 10-2. Position Measurement I Connection

Channel	SIN	COS
0	ACLOCK0	ACLOCK1
1	ACLOCK2	ACLOCK3
2	BCLOCK0	BCLOCK1
3	BCLOCK2	BCLOCK3
4	CCLOCK0	CCLOCK1
5	CCLOCK2	CCLOCK3
6	DCLOCK0	DCLOCK1
7	DCLOCK2	DCLOCK3

The Position Measurement II function requires an entire Function Block (two channels) to operate. The channel pair selected to be used is connected to the Quadrature Position transducer in the following manner:

1. The SIN input is connected to the CLOCK0 input.

2. The COS input is connected to the CLOCK1 input.

3. The INDEX input is connected to the CLOCK2 input.

The Position Measurement II configuration for each channel pair is shown in Table 10-3 below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 10-3. Position Measurement II Connection

Channel	SIN	COS	INDEX
0	ACLOCK0	ACLOCK1	ACLOCK2
2	BCLOCK0	BCLOCK1	BCLOCK2
4	CCLOCK0	CCLOCK1	CCLOCK2
6	DCLOCK0	DCLOCK1	DCLOCK2

10.3 PROGRAMMING

The Position Measurement I function is initialized by sending the following command to any of the ICM's eight channels.

72H = Start Position Measurement I

The position calculated is read by sending the following command to the channel being used for Position Measurement I.

73H = Read Position I

The Position Measurement II function is initialized by sending the following command to any of the ICM's four even channels (0, 2, 4, or 6).

70H = Start Position Measurement II

The position calculated is read by sending the following command to the channel being used for position measurement.

71H = Read Position II

10.3.1 Start Position Measurement I

The Start Position Measurement I command configures the specified channel to compute the position of the rotating device, based on the SIN and COS inputs received from a transducer attached to the device. This command returns a response immediately.

The position of the rotating device at this point is recorded within the ICM as the Reference Position.

Start Position Measurement I Command Format

Command Word: 72H

no operands required

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10.3.2 Read Position I

The Read Position I command returns the following two operands (described in the beginning of Section 10.1) back to the Command Block where they are read by the host: Direction from Reference and Number of Steps from Reference Position. This command returns a response immediately. The format is listed below:

Read Position I Command Format

Command Word: 73H

Four Operand Bytes:

Byte 1: Direction from Reference Position ——— (0 = CW, Ø = CCW relative to Reference Position)

Byte 2: Unused

Bytes 3-4: Number of Steps from Reference Position

10.3.3 Start Position Measurement II

The Start Position Measurement II command configures the specified channel pair to compute the position of the rotating device, based on the inputs from a transducer attached to the device. This command returns a response immediately.

The position at this point is recorded within the ICM as the Reference Position. When an index pulse is received, the current position is recorded within the ICM as the Index Position.

The channel pairs for Position Measurement are selected by writing to the even channel of the Channel pair (e.g., to select Channel pair 0,1 the Command Block is sent to Channel 0). Note that this means the corresponding odd channel is no longer available to the user.

Start Position Measurement II Command Format

Command Word: 70H

no operands required

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10.3.4 Read Position II

The Read Position II command returns the following four operands (described in the beginning of Section 10.1) back to the Command Block where they are read by the host: Direction from Reference, Direction from Index Position, Number of Steps from Index Position, and Number of Index Pulses from Reference Position. This command returns a response immediately. The format is listed below:

Read Position II Command Format

Command Word: 71H

Six Operand Bytes:

Byte 1: Direction of Steps from Index Position ———— (0 = CW, ∅ = CCW relative to Index Position)

Byte 2: Direction of Index Pulses from Reference Position ———— (0 = CW, ∅ = CCW relative to Reference Position)

Bytes 3-4: Number of Steps from Index Position

Bytes 5-6: Number of Index Pulses from Reference Position

Note that between the time that the Start Position Measurement command is given and the time that the first index pulse is received, the Number of Steps from Index Position operand will actually indicate the number of steps from the Reference Position. In addition, during this time the Direction of Index Pulses from Reference Position will be zero (i.e., CW). (Before the index pulse is received, the Direction of Steps from Index Position and Number of Steps from Index Position operands behave identically to the Position Measurement I Direction of Steps from Reference Position and Number of Steps from Reference Position operands, respectively.) It is apparent, then, that the Position Measurement II function could be used in an application where an index signal is not connected, but the user would be better off using the Position Measure I function as it requires only one ICM I/O channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 11

STEPPER MOTOR CONTROL

The ICM Stepper Motor Control function enables the user to easily control the motion of a stepper motor. The user may, for example, move the stepper motor a specified number of steps and then stop, or he may allow it to run continuously.

The Stepper Motor Control function can be programmed to drive either of two types of stepper motor translators: the type requiring a separate pulse train for each direction, or the type requiring a pulse train for stepping rate and a direction signal to specify direction. The type is specified upon initialization (see Section 11.3).

Upon first initializing the Stepper Motor Control function, the user must specify four operands: time for ramp up or down, number of ramp steps, minimum motor rate and maximum motor rate. The Stepper Motor Control function requires two channels to operate.

When dealing with stepper motors, one important operand that must be predefined is the stepper motor ramp time. This is the time it takes for the stepper motor to accelerate to the chosen maximum speed, therefore determining the rate of acceleration. The same rate also is used to decelerate the stepper motor, either to a slower speed or to a stop. The ramp time operand is required upon initialization, so it must be defined if the user intends to specify a fixed number of motor steps or continuous motion. The ramp time is a two-byte operand that is expressed in integer form and specified in units of milliseconds.

The number of ramp steps operand (see Figure 11-1) is a one-byte operand with a range of 1 to 32 that specifies the number of "steps" taken as the stepper motor is accelerated to maximum speed.

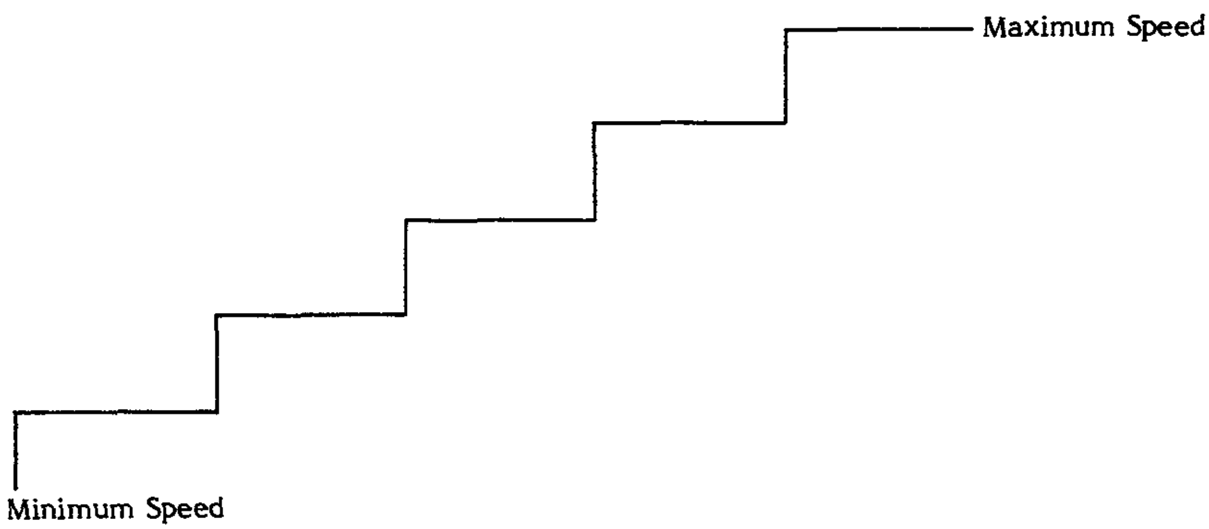


Figure 11-1. Motor Acceleration for Ramp Steps = 5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The minimum motor rate operand specifies the slowest speed the motor will move during the performance of the command. This operand is defined in pulses per second and has a range of 1 - 65534.

The maximum motor rate operand specifies the maximum speed the motor will move during the performance of the command. This operand is defined in pulses per second and has a range of 2 - 65535.

After the function has been initialized, the user must specify three operands in each motion command: the direction, the stepping rate, and the number of pulses the motor is to move.

The direction operand specifies a clockwise movement when equal to zero (0), and a counter-clockwise movement when not equal to zero.

The stepping rate operand is the rate the user desires the stepper motor to move while performing its move or spin operation. This operand must be greater than or equal to the minimum motor rate specified upon initialization, and less than or equal to the maximum motor rate specified upon initialization. The stepping rate is a two-byte operand with a range of zero to 65535.

The number of pulses operand is a two-byte operand with a range of zero to 65535 that is used in the Move Motor command to specify the number of pulses the stepper motor will move.

11.1 HARDWARE CONFIGURATION

The Stepper Motor Control function requires two channels (one Function Block) to operate. The channel pairs are: 0,1; 2,3; 4,5; and 6,7. They are written to as 0, 2, 3, and 6 respectively. For implementation of the dual and single pulse train stepper motor control on each channel pair, see Tables 11-1 and 11-2.

Table 11-1. Connection for Dual Pulse Train
Stepper Motor Translator

Channel	CW Output	CCW Output
0	AOUT0	AOUT2
2	BOUT0	BOUT2
4	COUT0	COUT2
6	DOUT0	DOUT2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 11-2. Connection for Pulse Train with Direction Bit Stepper Motor Translator

Channel	Stepping Rate	Direction
0	AOUT0	AOUT2
2	BOUT0	BOUT2
4	COUT0	COUT2
6	DOUT0	DOUT2

11.2 PROGRAMMING

The Stepper Motor Control function is serviced by the following commands:

40H = Configure Stepping Motor Control as a Single Pulse Train With a Direction Signal

41H = Configure Stepping Motor Control as a Dual Directional Pulse Train

42H = Move Motor

43H = Spin Motor

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2.1 Configure Stepping Motor as a Single Pulse Train with a Direction Signal

The Configure Stepping Motor Control as a Single Pulse Train With a Direction Signal command configures the channel pair written to for single pulse train stepping motor control. The channel pair provides the proper signals for a single pulse train stepper motor translator. A channel running this function will not be "released" until a Stop command is issued. This command returns a response immediately. The format is listed below:

Configure Single Pulse Train Command Format

Command Word: 40H

Eight Operand Bytes

Bytes 1-2: Minimum motor rate ——— (range of 0-65534 pulses/sec)
Byte 1: MSB
Byte 2: LSB

Bytes 3-4: Maximum motor rate ——— (range of 1-65535 pulses/sec)
Byte 3: MSB
Byte 4: LSB

Bytes 5-6: Time for ramp up or down ——— (units are in milliseconds)
Byte 5: MSB
Byte 6: LSB

Bytes 7-8: Number of ramp steps ——— (range of 1-32)
Byte 7: MSB
Byte 8: LSB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2.2 Configure Stepping Motor Control as Dual Pulse Train

The Configure Stepping Motor Control as a Dual Pulse Train command configures the channel pair written for dual pulse train stepping motor control. The channel pair provides the stepper motor translator with a clockwise and counter-clockwise pulse train. A channel running this function will not be "released" until a Stop command is issued. This command returns a response immediately. The format is listed below:

Configure Dual Pulse Train Command Format

Command Word: 41H

Eight Operand Bytes

Bytes 1-2: Minimum motor rate ——— (range of 0-65534 pulses/sec)
  Byte 1: MSB
  Byte 2: LSB

Bytes 3-4: Maximum motor rate ——— (range of 1-65535 pulses/sec)
  Byte 3: MSB
  Byte 4: LSB

Bytes 5-6: Time for ramp up or down ——— (units are in milliseconds)
  Byte 5: MSB
  Byte 6: LSB

Bytes 7-8: Number of ramp steps ——— (range of 1-32)
  Byte 7: MSB
  Byte 8: LSB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2.3 Move Motor

The Move Motor command programs the specified channel pair (previously initialized in one of the two stepper motor modes) to move the stepper motor the specified number of steps in the specified direction and then stop. To stop this command in "mid-move" issue the channel a Stop command. This command returns a response when the movement is completed. The format is listed below:

Move Motor Command Format

Command Word: 42H

Six Operand Bytes

Byte 1: Direction ———————— 0 = CW, 1 = CCW

Byte 2: Unused

Bytes 3-4: Stepping Rate ———— Range of 0-65535 pulses/seconds
Byte 3: MSB — The stepping rate must be > the minimum
Byte 4: LSB — rate and < the maximum rate.

Bytes 5-6: Number of pulses ——— Range of 0-65535
Byte 5: MSB
Byte 6: LSB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2.4 Spin Motor

The Spin Motor command programs the channel pair (previously initialized in one of the two stepper motor modes) to spin the motor until a Stop command is issued or a spin rate of 0 is specified. The Stop command will cause an immediate stop with no ramp down. It returns a response immediately. The format is listed below:

Spin Motor Command Format

Command Word: 43H

Four Operand Bytes

Byte 1: Direction ——————— 0 = CW, 1 = CCW

Byte 2: Unused

Bytes 3-4: Stepping Rate ——————— Range of 0-65535 pulses/seconds
Byte 3: MSB
Byte 4: LSB

The stepping rate must be $\geq$ the minimum motor rate and $\leq$ the maximum motor rate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP