Mux488/64

Analog Multiplexer

For the Mux488/64 and Mux/64

USER'S MANUAL

ioTech

Copyright Notice

Warranty

IOtech warrants each of our products to be free from defects in material or workmanship. Our obligation under this warranty is to repair or replace. at our option, any product or part thereof during the time period indicated on the product's warranty card. No other warranty is expressed or implied. IOtech does not warrant that Mux488/64 and Mux/64 will operate as described in this manual in every hardware and software environment.

Limitation of Liability

Mux488/64 and Mux/64 have been thoroughly tested and the documentation reviewed. However, IOtech does not warrant the performance of the product for any particular purpose. In no event is IOtech liable for any damages resulting, directly or indirectly. from the use of this product.

Trademark Notices

IOtech, Inc. 25971 Cannon Road. Cleveland. Ohio 44146

Printed in the United States of America.

Mux488/64
User ' s Manual

Part #188-0920

Revision 1 .O

January 1992

Table of Contents

Table of Contents

Introduction

1.1 General Description

Mux488/64 and Mux/64 are 64 single-ended or 32 differential signal analog multiplexers. They expand the input channel capacity of analog data acquisition systems and can provide front-end signal multiplexing for analog to digital (A/D) converters such as the IOtech ADC488. The multiplexers can be switched directly under program control or can be switched automatically through a programmable sequence of settings.

For application flexibility, the Mux488/64 and Mux/64 can be easily configured for the following switching capabilities:

64 single-ended inputs: One single-ended output.
32 differential inputs: One differential output.
Two banks of 32 single-ended inputs: 2 single-ended outputs.
Two banks of 16 differential inputs: 2 differential outputs.
Four banks of 16 single-ended inputs: 4 single-ended outputs.
Eight banks of 8 single-ended inputs: 8 single-ended outputs.
Four banks of 8 differential inputs: 4 differential outputs.

A Mux system can be set up in one of several configurations: master/slave with RS-232C or IEEE 488 interfacing, LPT port peripheral, plug-in card port peripheral or digital I/O port peripheral. A Mux488/64 can be set for master, slave, LPT port, digital I/O port or plug-in card port peripheral operation. It is a master unit that can be operated independently or as the control unit for up to fifteen slave Mux/64s. A Mux/64 can only be set for slave, LPT port, digital I/O port or plug-in card port peripheral operation. Mux/64 is not capable of master operation. Mux488/64 programs slave units through its digital control port, a parallel TTL interface. Digital control can be driven by a Mux488/64 acting as a master, any parallel Centronics port, or a digital I/O interface such as the IOtech Power488, the MetraByte PIO-12 or DAS-16 or the digital I/O port of the Data Translation DT2801.

Mux488/64 and Mux/64 are packaged in a standard 1U (1.75") high, 19-inch rack enclosures. An IEEE 488 standard connector is provided for bus communication and an RS-232C interface is available through a DB-9 connector. Analog inputs are accommodated by four DB-25 plug connectors on the rear panel of the instrument; analog outputs are accessed through two DB-25 sockets. The digital control for slave devices is available on a DB-15 socket connector.

1.2 Available Accessories

Additional accessories that can be ordered for the Mux488/64 and Mux/64 include:

CA-7-3	6 foot shielded IEEE 488 cable
CA-81	Master to slave interface cable
CA-82	Slave to slave interface cable
CA-83	PC LPT port to slave interface cable
CA-84	Power488 digital output to slave interface cable
CA-85	PC serial port to master interface cable
CA-86	PC/AT serial to master interface cable
CA-91	20 pin ADC488 series edge card connector to DB-15 slave control cable
CA-96	ADC488/16A or ADC488/8SA to Mux488/64 or Mux/64 interface cable
Mux488/64-901	Additional user's manual

1.3 Specifications

Power: 90-125/210-240 V ac 50/60 Hz 60 VA

Dimensions: 425 mm wide x 45 mm high x 203 mm deep (16.75" x 1.75" x 8")

Mux488/64 Interfaces:

IEEE 488 Interface: SH1, AH1, T6, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT1, C0, E1

RS-232C Interface (EIA RS-232C): AB, BE, BB, CA, CB
Baud Rates: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400

Data Bits: 8, Stop Bits: 1

Parity: even, odd, none

Terminator: CR or LF

Mux/64 Interface: 7 data, 6 control/status lines, TTL levels.

Slave Control Output: Proprietary 15-pin TTL

Electrical Isolation Between Analog Channels and Control Interfaces: 500V (peak)

Temperature Rating: 0-50°C, 0-95% RH, non-condensing

Weight: 5.3 lb.

Input Channels: 64 SE/32 DE

Output Channels: 1, 2, 4, or 8, user selectable

Analog Signal Level: ±10 V maximum, protected for signals up to 25 volts.

Switch ON Resistance: 2 kOhm

Maximum Disconnected Output Leakage: 20 nA max

Analog Signal Bandwidth: 100 kHz

Commutation Rate: 4 kHz

Channel to Channel Isolation: 70 dB @ 50 kHz @ 1 kOhm

Getting Started

The Mux488/64 manual includes information about both the Mux488/64 and the Mux/64. "Mux" refers to either the Mux488/64 or the Mux/64. Where information is specific to the Mux488/64 or the Mux/64, the full name is used.

2.1 Inspection

The Mux was carefully inspected mechanically and electrically prior to shipment. When you receive the interface, carefully unpack all items from the shipping carton and check for any obvious signs of physical damage that may have occurred during shipment. Report any such damage found to the shipping agent immediately. Retain all shipping materials in the event that shipment back to the factory is necessary.

Every Mux is shipped with the following items:

Mux488/64 or Mux/64	64 channel analog multiplexer with RS-232C and IEEE 488 interfaces. 64 channel analog multiplexer with 8-bit digital I/O control interface.	
Mux488/64-901	Mux488-901	User's Manual
188-0600	Sample program disk in IBM Format	
000-0800	Accessory kit, including:	
	CA-1	Power cable
	FE-1	Rubber Feet (4)
	EN-6	Rack Ears (2)
	HA-41-6	Screws (4)
	FU-1-.5	1/2 Amp Replacement Fuse
	FU-1-.25	1/4 Amp Replacement Fuse

WARNING

The Muxes are intended for INDOOR USE ONLY. Failure to observe this warning could result in equipment failure or personal injury.

2.2 Internal Configuration

The internal configuration of a Mux consists of setting the line voltage and replacing fuses. Line voltage must be set for 110 or 220 V ac to match the power being supplied to the Mux. If the line voltage is changed, the fuse must also be changed. See Figure 2.1 for line voltage switch and fuse locations.

WARNING

Disconnect the power cord from the power line and from the Mux prior to disassembly.

Never open the Mux case while it is connected to the power line. Internal voltage potentials exist that could cause personal injury.

The Mux is designed to switch signals that may carry dangerous voltages. De-energize circuits connected to the Mux before changing the configuration of the Mux or its wiring.

To avoid injury, de-energize attached circuits, then disconnect the Mux from other equipment and the power line before changing the internal configuration of the Mux.

To open the unit, place the Mux on a flat surface. Remove the four screws on the top cover of the case. Lift the top cover off. Reverse this procedure to reassemble the unit.

2.2.1 Line Voltage Selection

The Mux can be operated from 90-125 or 210-250 V ac, 50-60 Hz. The interface was shipped from the factory set for the operating voltage marked on the label placed over the rear panel line cord jack. If this setting is not appropriate for the power to be supplied to the unit, the setting of the internal voltage switch (S3) and the power fuse must be changed to avoid damage to the unit. The locations of S3 and the fuse are shown in Figure 2.1.

CAUTION

A fuse with a rating higher than that specified may cause damage to the instrument and may pose a fire hazard. If the instrument repeatedly blows fuses, locate and correct the cause of the trouble before replacing the fuse.

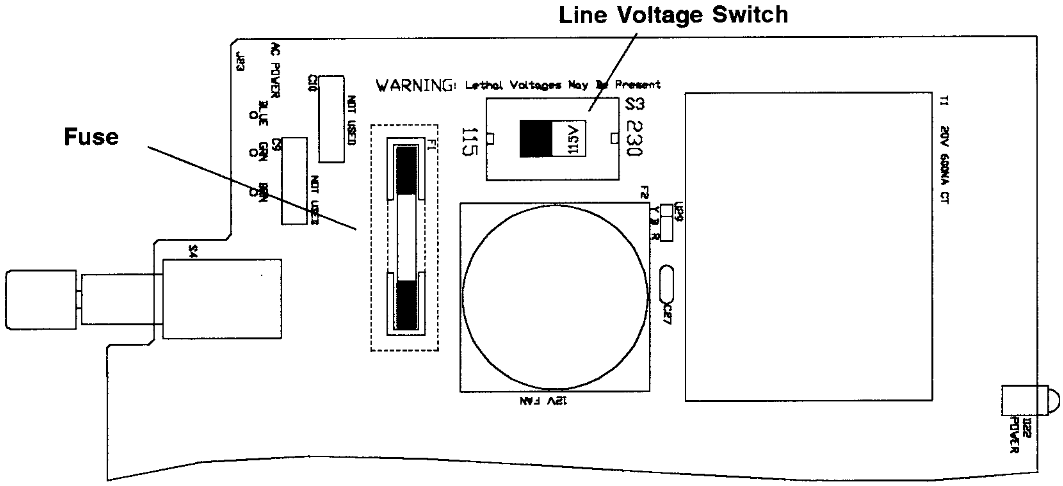

Figure 2.1: Line Voltage Switch and Fuse Location

1. The line voltage selection switch (S3) is located next to the fan and the power supply transformer. Insert the tip of a small screwdriver into the slot of the switch and slide the switch until it clicks into place with the desired line voltage visible.

2. Install a power line fuse appropriate for the line voltage. The fuse is located next to the internal line voltage switch (S3). Pull upward on the plastic fuse housing to remove the entire housing with the fuse inside. Select a fuse with the proper rating (see table below).

Line Voltage	Fuse Type
90-125V	½ A 250V, Slo Blo, 3AG
210-250V	¼ A 250V, Slo Blo, 3AG

3. Open the fuse housing by pushing up on the tab on the bottom of the housing.

4. Replace the fuse and close the housing. Replace the fuse housing into the fuse holder. Make sure the fuse snaps into place.

5. Make note of the new voltage setting for later reference and carefully re-assemble the unit.

2.3 Mounting

The Mux includes accessories for rack or bench use.

2.3.1 Rack Mount

If rack mount installation is required, remove the two plastic screws from the predrilled holes on each side of the unit. Only remove the screws from the set of holes that will be toward the front of the rack (the unit can be mounted with the front or rear panel facing the front of the rack fixture).

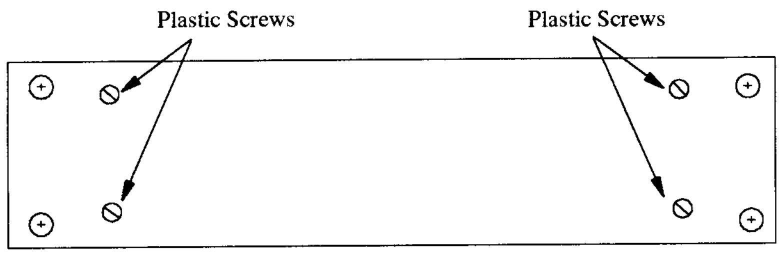

Figure 2.2: Rack Installation (Side View)

Install the two rack ears using the enclosed screws as shown in Figure 2.3.

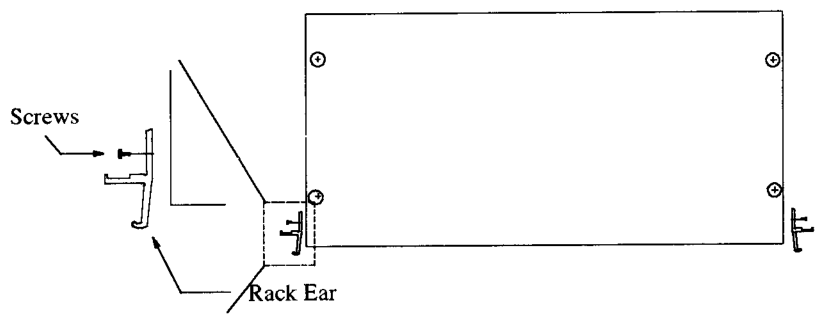

Figure 2.3: Installing Rack Ears (Top View)

2.3.2 Bench Top

If bench top installation is required, install the self-adhesive rubber feet on the bottom of the unit approximately one inch from each corner.

2.4 System Configurations

A Mux can be set up in one of several configurations: master/slave with RS-232C or IEEE 488 interfacing, LPT port peripheral, plug-in card port peripheral, or digital I/O port peripheral. A Mux488/64 can be set for master, slave, LPT port, plug-in card peripheral, or digital I/O port peripheral operation. A Mux/64 can only be set for slave, LPT port, plug-in card peripheral or digital I/O port peripheral operation. Mux/64 is not capable of master operation. The following sections describe the system configurations.

2.4.1 Master/Slave Configuration

In master/slave configuration, a Mux488/64 is configured as a master and is connected to a host computer. The master Mux488/64 can be connected to as many as fifteen slave Mux/64s or Mux488/64s through their Master/Slave ports. Master/slave configuration expands a single IEEE 488 bus address or RS-232C port to control a total of sixteen devices with as many as 1024 signals. To set up a system for IEEE 488 master/slave operation, see Section 2.6. To set up a system for RS-232C master/slave operation, see Section 2.7. See Section 3 for operation.

2.4.2 LPT Port Peripheral Configuration

The LPT port peripheral mode allows any IBM-PC compatible printer port to control a single Mux/64 or Mux488/64. To set up a system for LPT Port Peripheral operation, see Section 2.8. See Section 4 for operation.

2.4.3 Digital I/O Port Peripheral Configuration

Digital I/O port peripheral mode is almost identical to LPT port peripheral mode except that it is designed to allow the individual control of up to fifteen slave units (depending on electronic drive capability) from a single host computer. To set up a system for Digital I/O Port Peripheral operation, see Section 2.9. See Section 5 for operation.

2.4.4 Plug-In Card Port Peripheral Configuration

The plug-in-card port peripheral mode allows any digital I/O port with seven outputs and one input to quickly and simply control a single Mux. To set up a system for Plug-in Card Port Peripheral operation, see Section 2.10. See Section 6 for operation.

2.5 Default External Switch Settings

The Mux system configuration is set by switches accessible from its rear panel. The Mux488/64 has two eight position switches that determine the command set used, the configuration mode and the settings for each configuration. The Mux/64 has two eight position switches that set the configuration mode and the settings for that configuration. The switches labeled W, X, Y and Z are not used on the Mux488/64 and should be set down (0) for future compatibility.

The rear panel switches are read only during power on or reset and should be set before applying power. Figures 2.4 and 2.5 show the factory default settings for the rear panel switches. The tables following the diagrams show the options for the switches. Switch settings are shown graphically in later sections describing specific setups for IEEE 488, RS-232C, LPT Port, Plug-in Card Port and Digital I/O Port operation.

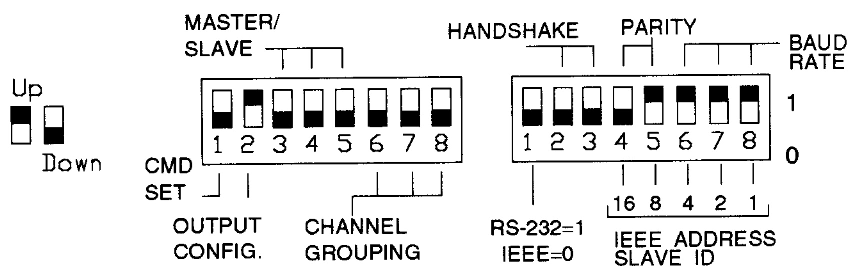

Figure 2.4: Mux488/64 Default Switch Settings

Left Mux488/64 Rear Panel Switches		
Switch #	**Label**	**Settings**
1	CMD SET	1 (up): Reserved for future enhancements 0 (down): Register-based Command Set (default)
2	OUTPUT CONFIG.	1: Separate output (default) 0: Shared output.
3, 4, 5	MASTER/SLAVE	000: Master (default) 101: LPT Port 100: Slave 111: Digital I/O Port 110: Plug-in Card Port
6, 7, 8	CHANNEL GROUPING	000: 1 single-ended (SE) (default) 001: 2 SE 010: 4 SE, 2 Differential (DE) 011: 8 SE, 4 DE 100: 16 SE, 8 DE

Right Mux488/64 Rear Panel Switches		
Switch #	**Label**	**Settings**
1	RS-232=1 IEEE=0	0: IEEE 488 Operation (default) 1: RS-232C Operation
4, 5, 6, 7, 8 IEEE 488 Operation	IEEE ADDRESS	IEEE 488 bus address. Default is 15.
2, 3 RS-232C Operation	HANDSHAKE	00: No handshaking 01: XON/XOFF handshaking 10: DTR/CTS handshaking 11: XON/XOFF and DTR/CTS handshaking
4, 5 RS-232C Operation	PARITY	00: No parity 01: Odd parity 10: Even parity
6, 7, 8 RS-232C Operation	BAUD RATE	000: 300 baud 001: 600 baud 010: 1200 baud 011: 2400 baud 100: 4800 baud 101: 9600 baud 110: 19200 baud 111: 38400 baud
4, 5, 6, 7, 8 LPT Port, Digital I/O Port or Slave Operation	SLAVE ID	Slave identification number.

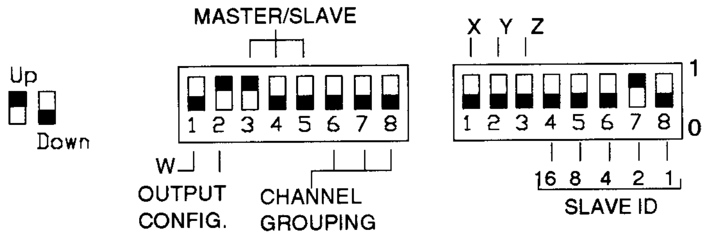

Figure 2.5: Mux/64 Default Switch Settings

Left Mux/64 Rear Panel Switches		
Switch #	**Label**	**Settings**
1	W	Not Used; Set to 0 (default).
2	OUTPUT CONFIG.	1: Separate output (default) 0: Shared output.
3, 4, 5	MASTER/SLAVE	000: Master 100: Slave (default) 110: Plug-in Card Port 101: LPT Port 111: Digital I/O Port
6, 7, 8	CHANNEL GROUPING	000: 1 single-ended (SE) (default) 001: 2 SE 010: 4 SE, 2 Differential (DE) 011: 8 SE, 4 DE 100: 16 SE, 8 DE

Right Mux/64 Rear Panel Switches		
Switch #	**Label**	**Settings**
1, 2, 3	X, Y, Z	Not Used; Set to 0 (default).
4, 5, 6, 7, 8	SLAVE ID	Slave identification number. Default is 2.

2.6 Master/Slave IEEE 488 Configuration

This section describes the configuration for Muxes to be used in master/slave operation commanded through the IEEE 488 bus. When switch setup is completed, refer to page 2.28 for wiring instructions.

The system setup with IOtech cable options is shown in Figure 2.6.

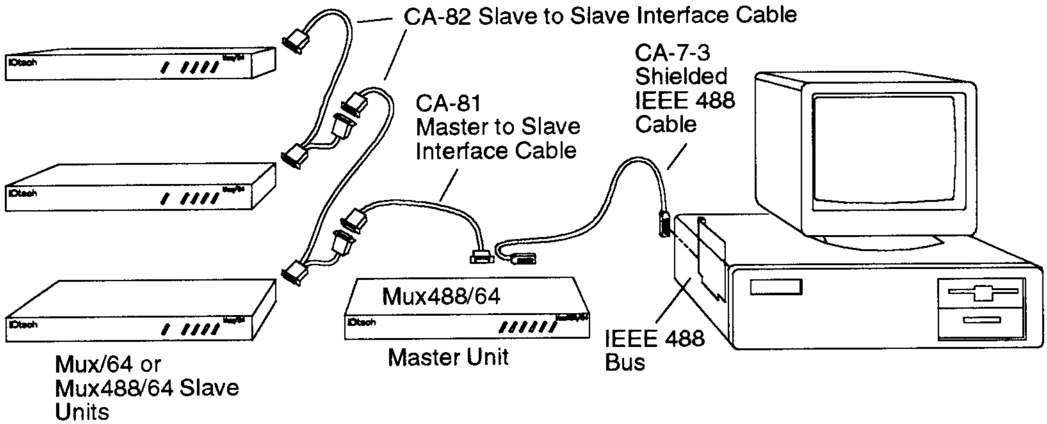

Figure 2.6: IEEE 488 Master/Slave System Configuration

2.6.1 Mux488/64 Master Unit Switch Settings

The master unit in a master/slave system configuration must be a Mux488/64 (it cannot be a Mux/64). The switch settings required for IEEE 488 operation are as follows.

The switch labeled CMD SET must be down (0). The up (1) position is reserved for future enhancements. This sets the unit for the register-based command set.

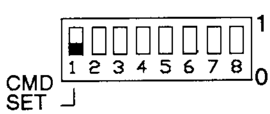

Figure 2.7: Set for Register-Based Command Set

The OUTPUT CONFIG. switch selects how outputs from a master and its slaves are combined. It can select shared outputs, in which outputs from the individual units are combined, or separate outputs, in which individual units act independently. See Section 2.11, Wiring, for more information on these modes.

Figure 2.8 shows the switch settings for shared and separate output modes.

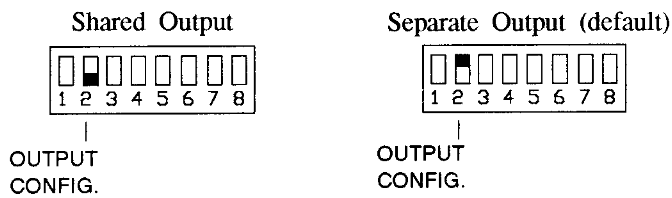

Figure 2.8: Output Configuration Settings

To operate the Mux488/64 through its IEEE 488 interface, the unit must be configured as a Master unit. This is done by setting the switches labeled MASTER/SLAVE as shown in Figure 2.9.

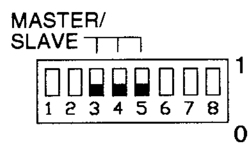

Figure 2.9: Mux488/64 Set as Master

Channel grouping is set by the switches labeled CHANNEL GROUPING. They select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. See Section 2.11, Wiring, for more information on channel grouping.

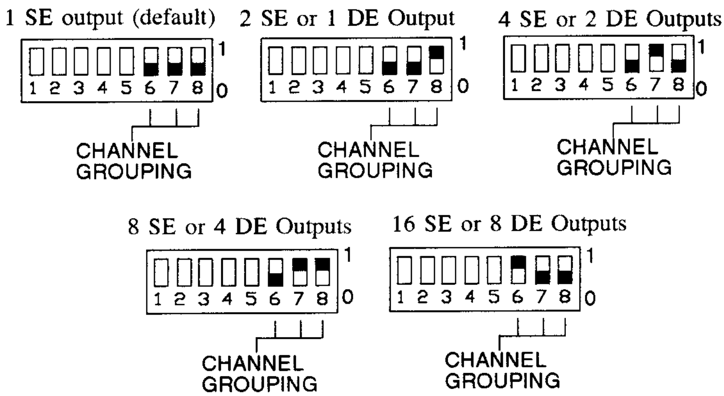

Figure 2.10: Mux488/64 Channel Groupings

The switch labeled RS-232=1/IEEE=0 must be set for the IEEE 488 bus, as shown in Figure 2.11.

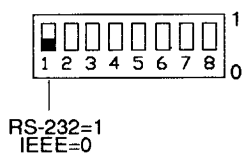

Figure 2.11: Mux488/64 Set for IEEE 488 Operation

All IEEE 488 bus devices, including the Mux488/64, must have an IEEE 488 bus address. The switches labeled IEEE ADDRESS are used for this purpose. The bus address can be set from 0 through 30 and is read only at power on or reset. The address is selected by simple binary weighting. The switch labeled 1 is the least significant bit; 16 is the most significant bit. The factory default is address 15, as shown in Figure 2.12. If the Mux's bus address switches are set to 31, a bus address of 30 is used.

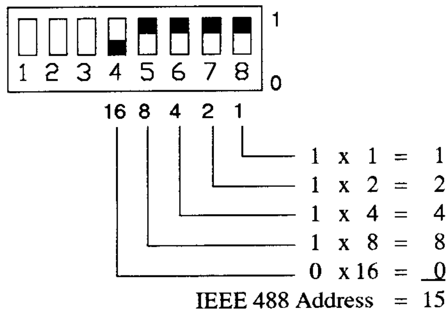

Figure 2.12: IEEE 488 Bus Address Setting (default)

2.6.2 Mux488/64 or Mux/64 Slave Unit Switch Settings

Both the Mux488/64 and the Mux/64 can be slaved to a Mux488/64. The switches that need to be set when a Mux is used as a slave are MASTER/SLAVE, CHANNEL GROUPING and SLAVE ID. Any unused switches should be set down (0).

When a Mux is to be used as a slave, set the switches labeled MASTER/SLAVE as shown in Figure 2.13.

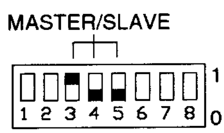

Figure 2.13: Mux Set as Slave

The channel grouping is set by the switches labeled CHANNEL GROUPING. If shared output configuration is being used, the slave units ignore their channel grouping setting and use the master's channel grouping setting.

The CHANNEL GROUPING switches select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. Figure 2.14 shows the channel grouping settings. See Section 2.11, Wiring, for more information on channel grouping.

Each slave unit must be assigned a unique slave identification number. The slave identification

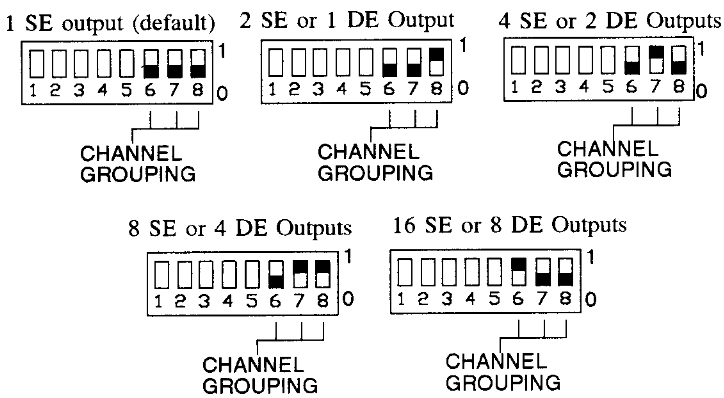

Figure 2.14: Mux Channel Grouping Settings

number can be set from 2 through 16 and is read only at power on or reset. The address is selected by simple binary weighting. The switch labeled 1 is the least significant bit; 16 is the most significant bit. The factory default for the Mux/64 is identification number 2, as shown in Figure 2.15.

> **If two or more slave units with the same slave identification settings are connected together, their operation, and the operation of other attached units, is unspecified and may cause errors during operation. Slave identification number 1 is not allowed because it is reserved for the master unit. Identification numbers 17 through 31 are interpreted as 1 through 16 by the Mux. 0 is interpreted as 16.**

When switch setup is completed, refer to page 2.28 for wiring instructions.

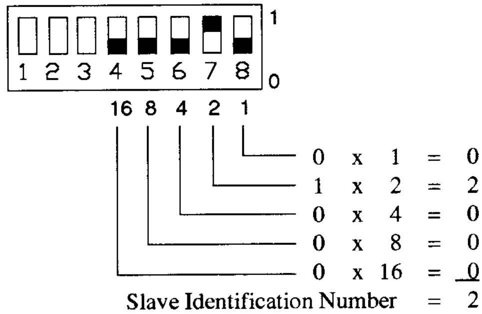

Figure 2.15: Slave Identification Setting

2.7 Master/Slave RS-232C Configuration

This section describes the configuration for Muxes to be used in master/slave operation using RS-232C communication. When setup is completed, refer to page 2.28 for wiring instructions.

The system setup with IOtech cable options is shown in Figure 2.6.

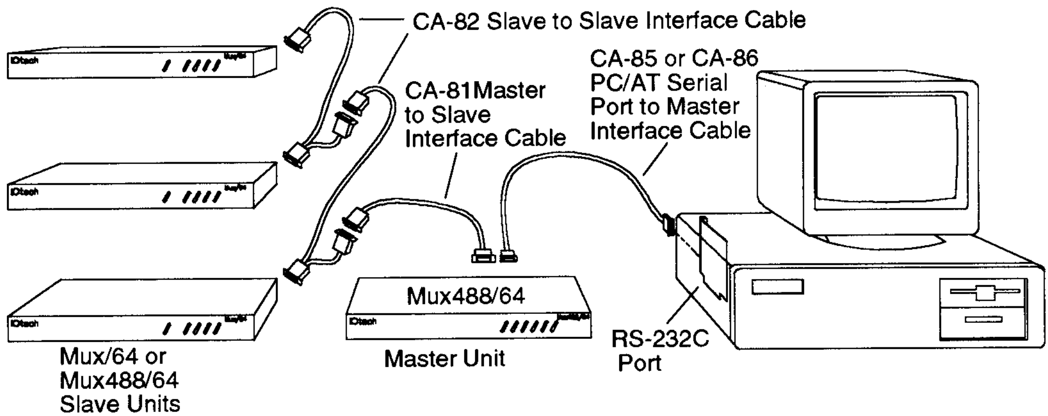

Figure 2.16: RS-232C Master/Slave System Configuration

2.7.1 Mux488/64 Master Unit Switch Settings

The master unit in a master/slave system configuration must be a Mux488/64 (it cannot be a Mux/64). The switch settings required for IEEE 488 operation are described in the following.

The switch labeled CMD SET must be down (0). The up (1) position is reserved for future enhancements. This sets the unit for the register-based command set.

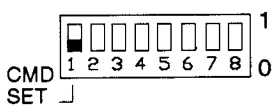

Figure 2.17: Set for Register-Based Command Set

The OUTPUT CONFIG. switch selects how outputs from a master and its slaves are combined. It can select shared outputs, in which outputs from the individual units are combined, or separate outputs, in which individual units act independently. See Section 2.11, Wiring, for more information on these modes.

Figure 2.18 shows the switch settings for shared and separate output modes.

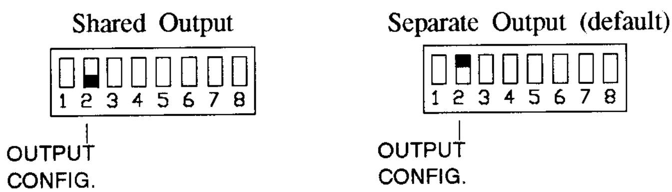

Figure 2.18: Output Configuration Settings

To operate the Mux488/64 through its RS-232C interface, the unit must be configured as a master unit. This is done by setting the switches labeled MASTER/SLAVE as shown in Figure 2.19.

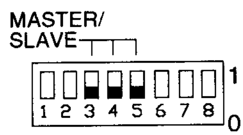

Figure 2.19: Mux488/64 Set as Master

Channel grouping is set by the switches labeled CHANNEL GROUPING. They select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. See Section 2.11, Wiring, for more information on channel grouping.

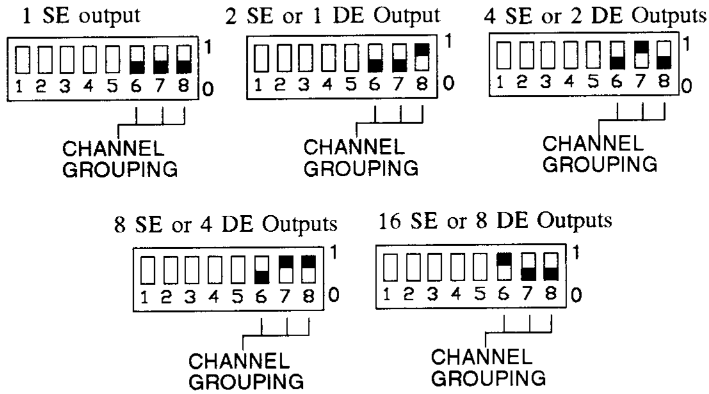

Figure 2.20: Mux488/64 Master Channel Group Settings

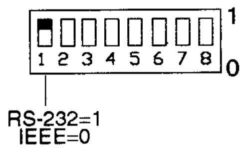

Figure 2.21: Mux488/64 Set for RS-232C Operation

The switch labeled RS-232=1/IEEE=0 must be set for RS-232, as shown in Figure 2.21.

When the RS-232C port is used, the type of handshaking must be selected by the switches labeled HANDSHAKE. Mux488/64 offers the options of no handshaking, XON/XOFF, DTR/CTS or both XON/XOFF and DTR/CTS handshaking. Figure 2.22 shows the switch settings for each option.

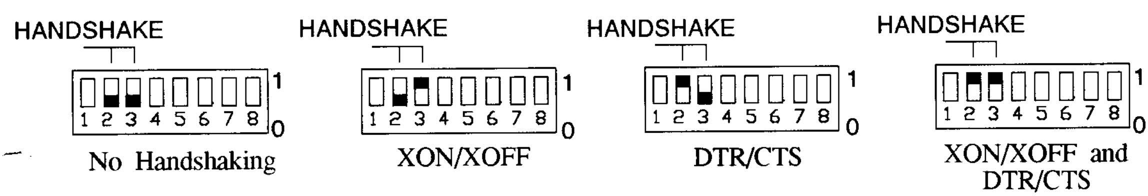

Figure 2.22: RS-232C Handshake Settings

The parity must be selected using the switches labeled PARITY. Mux488/64 provides for odd, even or no parity. Figure 2.23 shows the switch settings for each option.

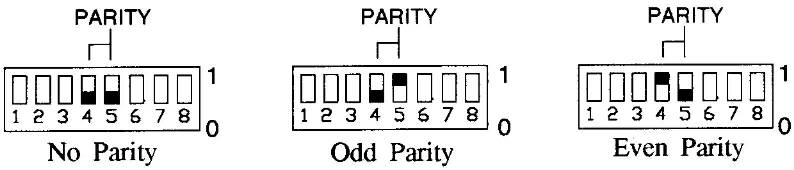

Figure 2.23: RS-232C Parity Settings

The baud rate is selected using the switches labeled BAUD RATE. The available baud rates are 300, 600, 1200, 2400, 4800, 9600, 19200 and 38400. The settings for each are shown in Figure 2.24.

Note: If sequenced operation is used, baud rates above 9600 may not be reliable.

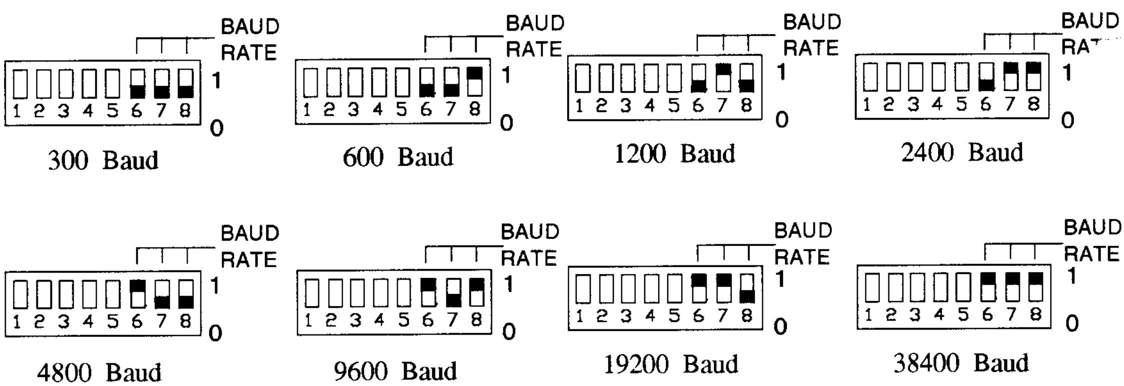

Figure 2.24: RS-232C Baud Rate Settings

2.7.2 Mux488/64 or Mux/64 Slave Unit Switch Settings

Both the Mux488/64 and the Mux/64 can be slaved to a Mux488/64. The switches that need to be set when a Mux is used as a slave are MASTER/SLAVE, CHANNEL GROUPING and SLAVE ID. Any unused switches should be set down (0).

When a Mux is used as a slave, set the switches labeled MASTER/SLAVE as shown in Figure 2.25.

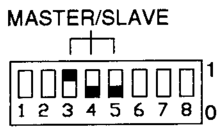

Figure 2.25: Mux Set as Slave

The channel grouping is set by the switches labeled CHANNEL GROUPING. If shared output configuration is being used, the slave units ignore their channel grouping setting and use the master's channel grouping setting.

The CHANNEL GROUPING switches select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. See Section 2.11, Wiring, for more information on channel grouping.

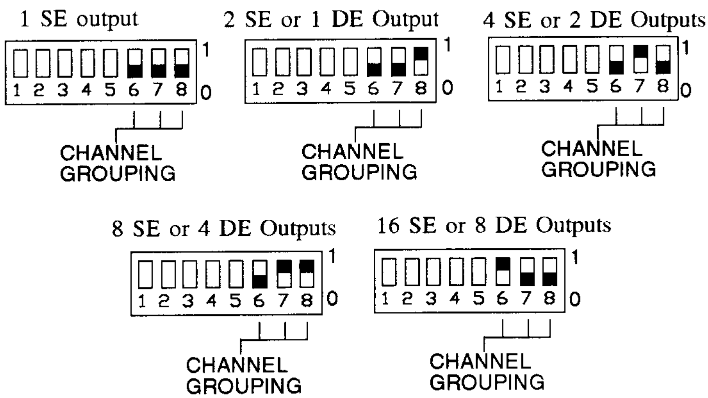

Figure 2.27: Mux Channel Grouping Settings

Each slave unit must be assigned a unique slave identification number. The slave identification number can be set from 2 through 16 and is read only at power on or reset. The address is selected by simple binary weighting. The switch labeled 1 is the least significant bit; 16 is the most significant bit. The factory default is identification number 2, as shown in Figure 2.26.

Caution

If two or more slave units with the same slave identifications are connected together, their operation, and the operation of other attached units, is unspecified and may cause errors during operation. Slave identification number 1 is not allowed because it is reserved for the master unit. Identification numbers 17 through 31 are interpreted as 1 through 16 by the Mux. 0 is interpreted as 16.

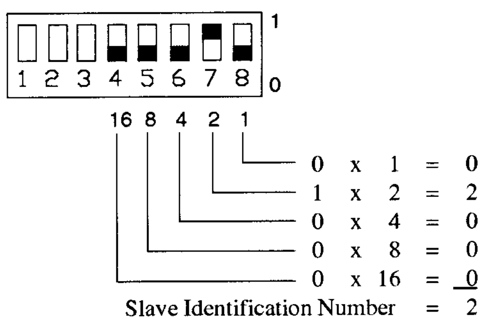

Figure 2.26: Slave Identification Setting (Default)

When switch setup is completed, refer to page 2.28 for wiring instructions.

2.8 LPT Port Peripheral Configuration

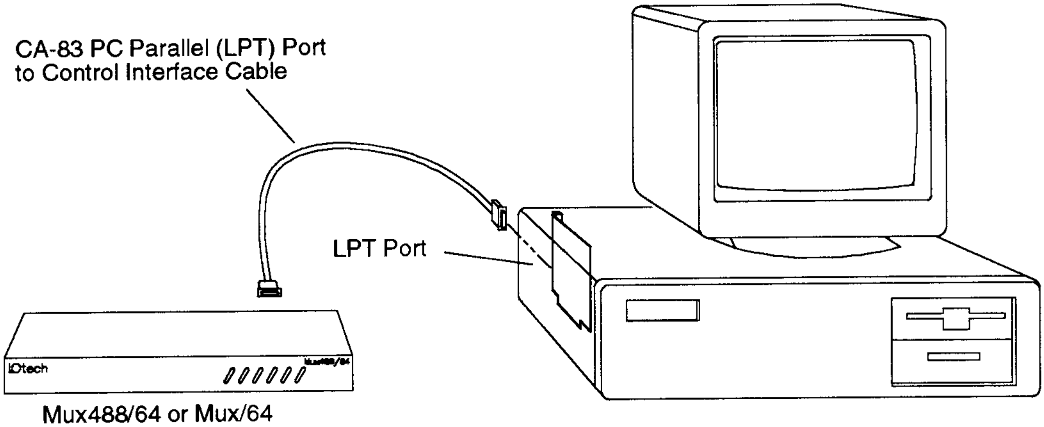

Figure 2.28: LPT Port System Configuration

The switches that need to be set for LPT Port Peripheral operation are the ones labeled MASTER/SLAVE and CHANNEL GROUPING. All other switches should be set down (0). Settings for the Mux488/64 and Mux/64 are the same for LPT Port Peripheral mode.

A Mux connected to an LPT printer port must have the switches labeled MASTER/SLAVE set as shown in Figure 2.29.

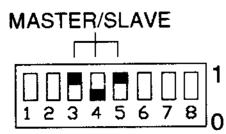

Figure 2.29: Mux Set for LPT Port Mode

Channel grouping is set by the switches labeled CHANNEL GROUPING. They select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. See Section 2.11, Wiring, for more information on channel grouping.

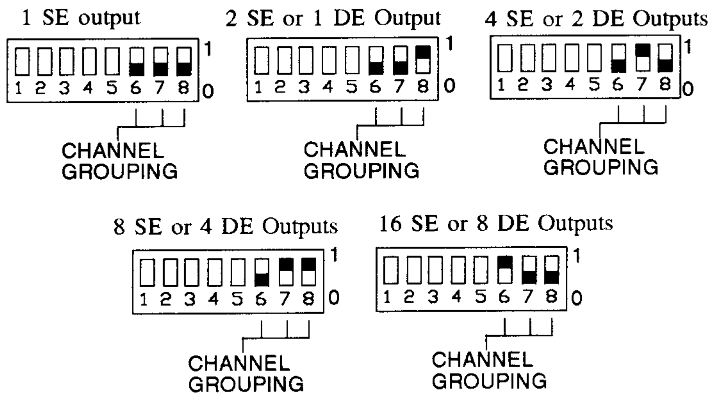

Figure 2.30: Channel Grouping Settings

When switch setup is completed, refer to page 2.28 for wiring instructions.

2.9 Digital I/O Port Peripheral Configuration Settings

This section describes the configuration for Muxes to be used in digital I/O port peripheral operation.

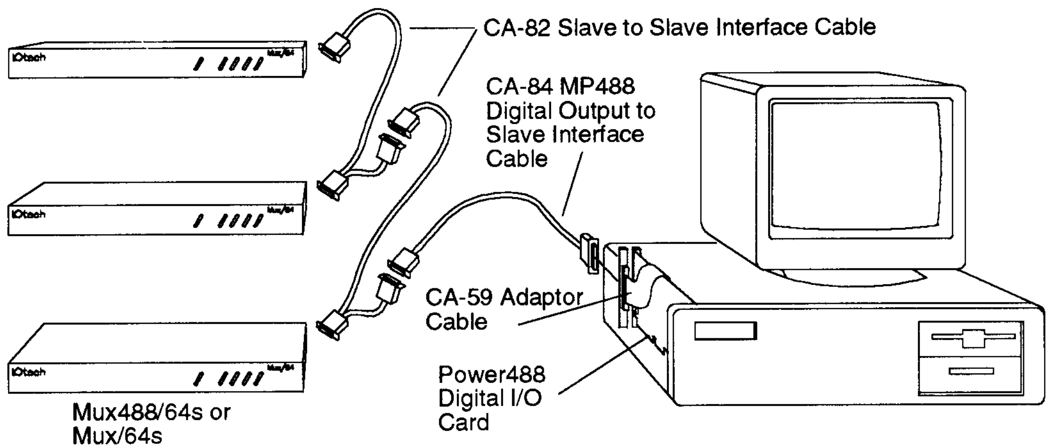

Figure 2.31: Digital I/O Port System Configuration

The number of units allowed in digital I/O port mode may be limited by the output current capabilities of the digital I/O port. Digital I/O must be able to sink at least 1.6 mA at 0.4 V for each attached unit. For example, when using a digital output port such as IOtech's Power488, which uses an 8255 output port that can sink 2.5 mA, only one unit can be attached. When driven by a 74LS244 bus buffer, which can sink 12 mA, a maximum of eight Muxes can be attached. When setup is completed, refer to page 2.28 for wiring instructions.

2.9.1 Mux Digital I/O Port Mode Switch Settings

When Digital I/O Peripheral Port mode is used, the switches labeled MASTER/SLAVE, CHANNEL GROUPING and SLAVEID must be set. All other switches should be set down (0).

When a Mux is connected to a digital output port, the switches labeled MASTER/SLAVE must be set as shown in Figure 2.32.

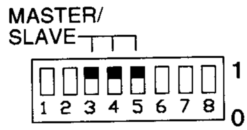

Figure 2.32: Mux Set to Digital I/O Port Mode

Channel grouping is set by the switches labeled CHANNEL GROUPING. They select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. See Section 2.11, Wiring, for more information on channel grouping.

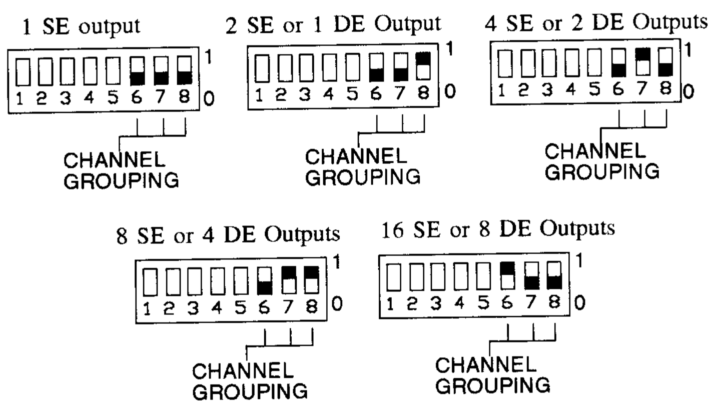

Figure 2.33: Channel Group Settings

Each Mux in a digital I/O port system must be assigned a unique slave identification number. The slave identification can be set from 2 through 16 and is read only at power on or reset. The address is selected by simple binary weighting. The switch labeled 1 is the least significant bit; 16 is the most significant bit. The factory default is identification number 15, as shown Figure 2.34.

Caution

If two or more units with the same slave identifications are connected together, their operation, and the operation of other attached units, is unspecified and may cause errors during operation. Identification numbers 17 through 31 are interpreted as 1 through 16 by the Mux. 0 is interpreted as 16. 1 is not allowed.

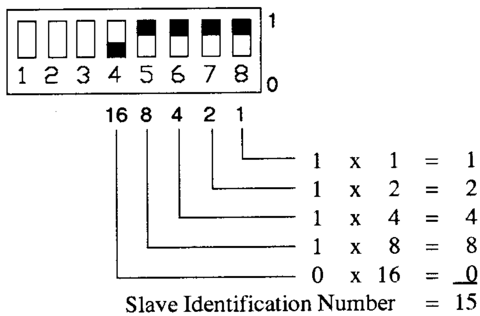

Figure 2.34: Slave Identification Setting

When switch setup is completed, refer to page 2.28 for wiring instructions.

2.10 Plug-In Card Port Peripheral Configuration

The switches that need to be set for Plug-in Card Port Peripheral operation are the ones labeled MASTER/SLAVE and CHANNEL GROUPING. All other switches should be set down (0). Settings for the Mux488/64 and Mux/64 are the same for Plug-in Card Port Peripheral mode.

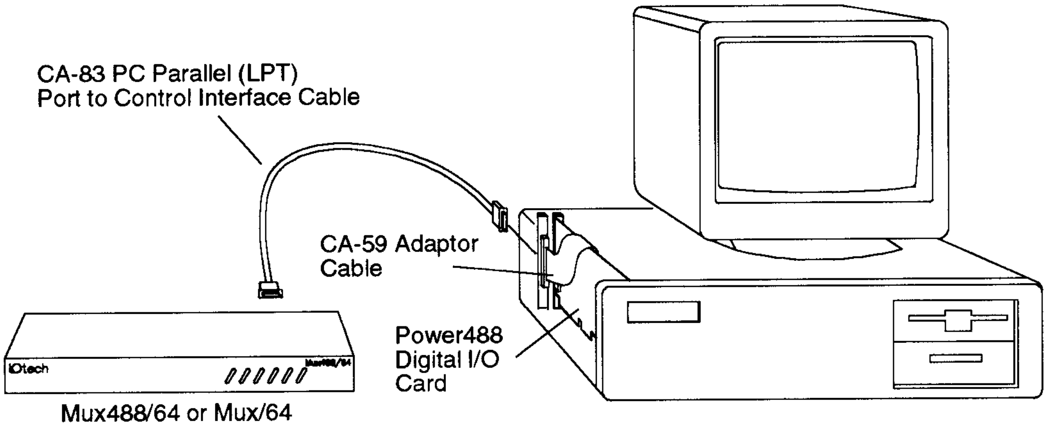

Figure 2.35: Plug-in Card Port System Configuration

When a Mux is used in a plug-in card configuration, the switches labeled MASTER/SLAVE must be set as shown in Figure 2.36.

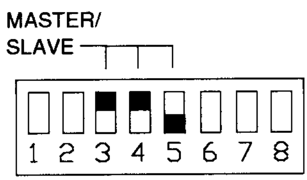

Figure 2.36: Mux Set for Plug-in Card Mode

Channel grouping is set by the switches labeled CHANNEL GROUPING. They select the number of active outputs, as well as the grouping of the input signals. There can be one, two, four, eight or sixteen active outputs. The 64 inputs are divided into groups of consecutive signals. Each group of inputs contains the same number of inputs as there are active outputs. For example, if four outputs have been selected, the 64 inputs are divided into sixteen groups of four consecutive signals each. The first group consists of inputs 1, 2, 3, and 4; the next 5, 6, 7, 8; and so on, with the last group consisting of inputs 61, 62, 63, and 64.

During operation, the outputs are either all disconnected or each one is connected to its corresponding signal from the specified input group. See Section 2.11, Wiring, for more information on channel grouping.

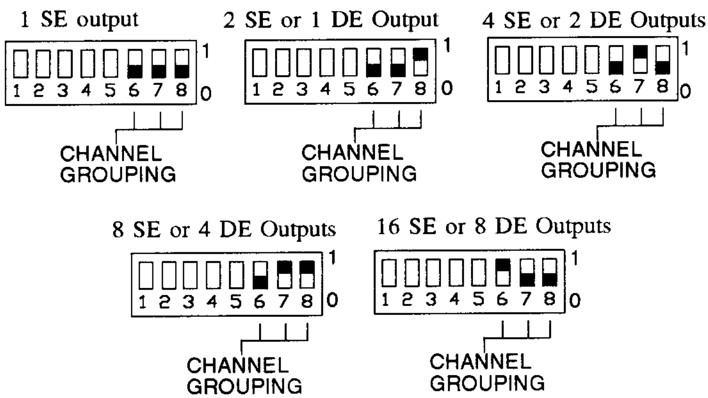

Figure 2.37: Channel Group Settings

When setup is completed, refer to page 2.28 for wiring instructions.

2.11 Wiring

Wiring of the Mux consists of wiring the DB-25 connectors. Four plug connectors accept the 64 inputs and two socket connectors provide the outputs. All of these connectors are located on the right hand side of the rear panel, as shown in Figure 2.38.

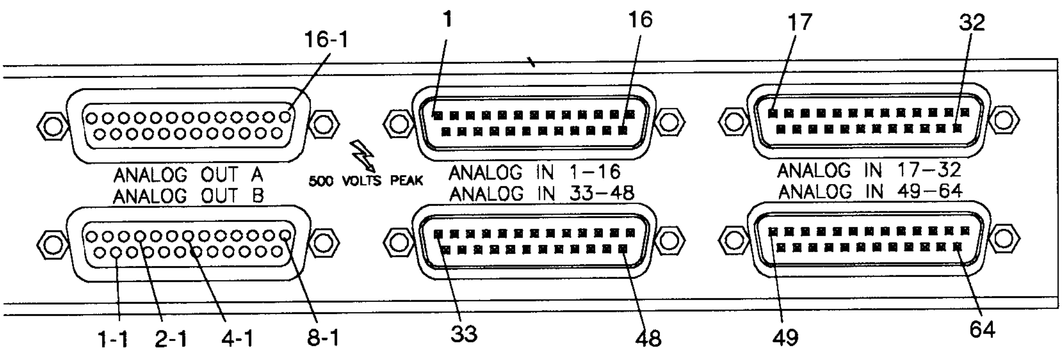

Figure 2.38: Rear Panel DB Connectors

WARNING

The Mux is designed to switch signals that may carry dangerous voltages. De-energize circuits connected to the Mux before changing the configuration of the Mux or its wiring.

The inputs are numbered 1 to 64 and are connected as shown in the following pinout tables.

ANALOG IN 1-16		ANALOG IN 17-32		ANALOG IN 33-48		ANALOG IN 49-64	
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	1	1	17	1	33	1	49
2	2	2	18	2	34	2	50
3	Ground	3	Ground	3	Ground	3	Ground
4	5	4	21	4	37	4	53
5	6	5	22	5	38	5	54
6	Ground	6	Ground	6	Ground	6	Ground
7	9	7	25	7	41	7	57
8	10	8	26	8	42	8	58
9	Ground	9	Ground	9	Ground	9	Ground
10	13	10	29	10	45	10	61
11	14	11	30	11	46	11	62
12	Ground	12	Ground	12	Ground	12	Ground
13	Shield	13	Shield	13	Shield	13	Shield
14	Ground	14	Ground	14	Ground	14	Ground
15	3	15	19	15	35	15	51
16	4	16	20	16	36	16	52
17	Ground	17	Ground	17	Ground	17	Ground
18	7	18	23	18	39	18	55
19	8	19	24	19	40	19	56
20	Ground	20	Ground	20	Ground	20	Ground
21	11	21	27	21	43	21	59
22	12	22	28	22	44	22	60
23	Ground	23	Ground	23	Ground	23	Ground
24	15	24	31	24	47	24	63
25	16	25	32	25	48	25	64

One to sixteen outputs may be active at a time. When sixteen outputs are active, they are carried on the pins listed as 16-1 through 16-16 on analog outputs connector A. When eight outputs are active, they are carried on the pins listed as 8-1 through 8-8. Similar numbering is used when four, two or just one outputs are active. The output signal connections are shown in the following pinout table.

ANALOG OUT A		ANALOG OUT B	
Pin	**Connection**	**Pin**	**Connection**
1	Signal 16-1	1	Signal 8-1
2	Signal 16-2	2	Signal 8-2
3	Ground	3	Ground
4	Signal 16-5	4	Signal 8-5
5	Signal 16-6	5	Signal 8-6
6	Ground	6	Ground
7	Signal 16-9	7	Signal 4-1
8	Signal 16-10	8	Signal 4-2
9	Ground	9	Ground
10	Signal 16-13	10	Signal 2-1
11	Signal 16-14	11	Signal 2-2
12	Ground	12	Ground
13	Shield	13	Shield
14	Ground	14	Ground
15	Signal 16-3	15	Signal 8-3
16	Signal 16-4	16	Signal 8-4
17	Ground	17	Ground
18	Signal 16-7	18	Signal 8-7
19	Signal 16-8	19	Signal 8-8
20	Ground	20	Ground
21	Signal 16-11	21	Signal 4-3
22	Signal 16-12	22	Signal 4-4
23	Ground	23	Ground
24	Signal 16-15	24	Signal 1-1
25	Signal 16-16	25	Ground

Each unit may be viewed as a switch that connects one of several groups of inputs to the outputs. For example, in the 16-output channel grouping mode, the unit switches one of four groups of sixteen inputs to the sixteen outputs. All of the outputs are always connected to their corresponding inputs from the same group: they may all be off, or they may all connect to the first input group, or to the second, or the third, or the fourth. Note that these five output connections (off, one, two, three or four) are the only possible connections. It is not possible to control the individual output signal connections. The 16-output channel grouping capabilities are represented by Figure 2.39, showing the input-output connections.

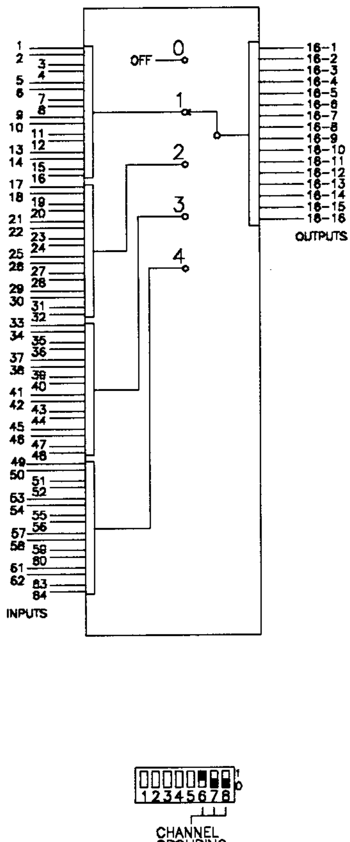

Figure 2.39: 16-Output Mode

This channel grouping can also be used to switch differential signals by using consecutive odd- and even-numbered signal connections to carry 32 differential input signals which are switched to generate eight differential output signals.

In the 4-output channel grouping mode (two differential outputs), the unit switches one of sixteen groups of four inputs to four outputs. In the 8-output channel grouping mode (four differential outputs), the unit switches one of eight groups of eight inputs to the eight outputs. Figure 2.40 shows the eight and four output channel grouping modes.

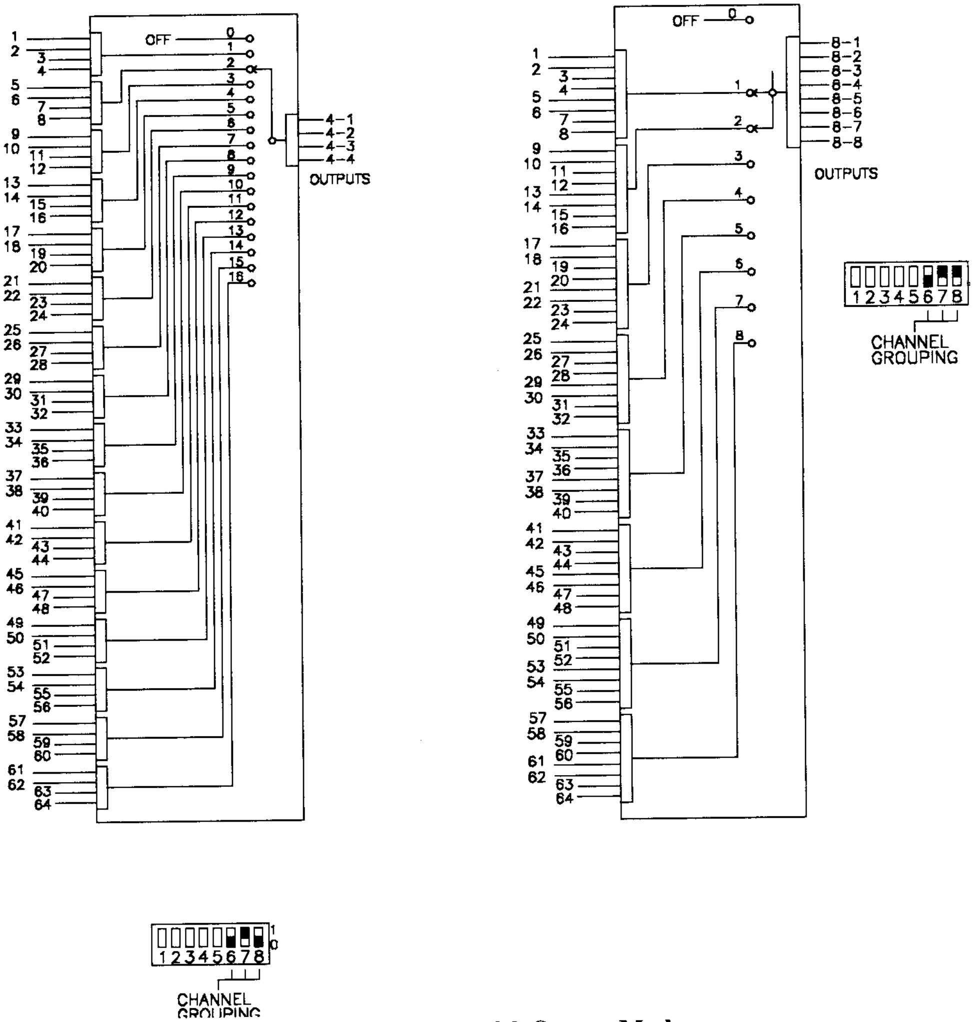

Figure 2.40: 4-Output and 8-Output Modes

Figure 2.41 shows single output and the two (single differential) output channel grouping modes.

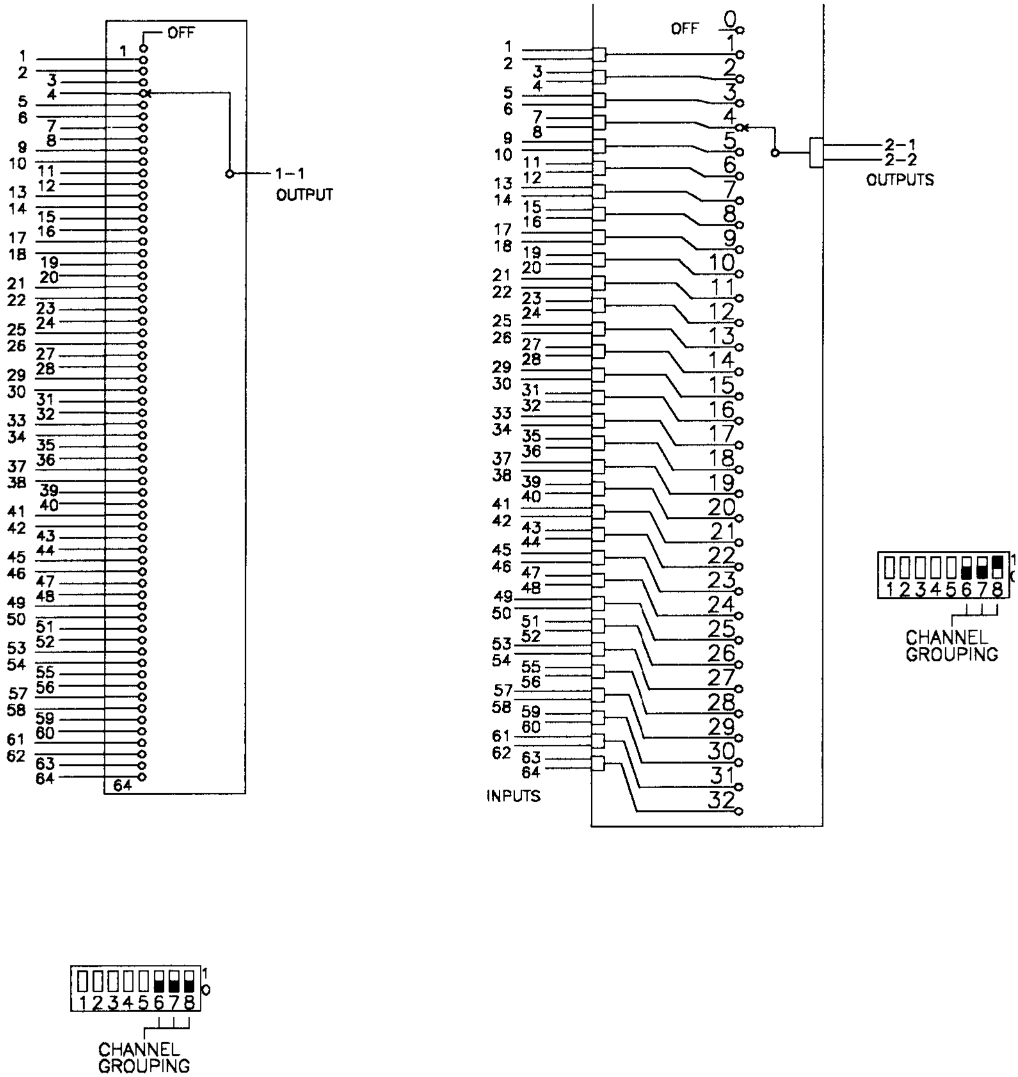

Figure 2.41: One Output and 2-Output Modes

To demonstrate these modes, assume there is a master unit with one attached slave. In separate mode, the two units operate independently. One might be configured to switch 64 inputs to one output, and the other switch its 64 inputs to 16 outputs. There is no relationship between their switch settings or their output connections.

This mode is shown in Figure 2.42.

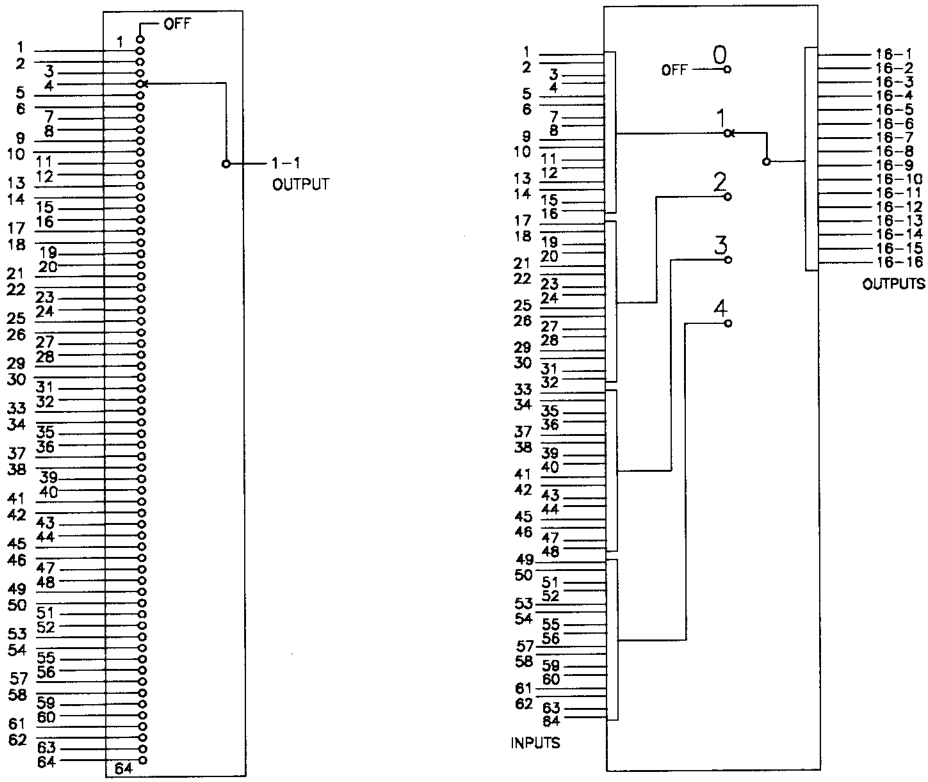

Figure 2.42: Separate Output Operation

In shared mode, the multiplexers act together to increase the number of inputs that can be switched to the outputs. The master unit sets the number of outputs that will be active on all units and the corresponding outputs from each unit are to be electrically wired together. For example, if the master is configured to switch 64 inputs to one output, the slave, regardless of its switch settings, also switches 64 inputs to one output. The two outputs, one from the master and one from the slave, should be wired together to form a single common output line switched among all 128 inputs.

This mode is not recommended unless more than 64 single-ended (32 differential) signals must be switched into a single output.

This mode is illustrated in Figure 2.43.

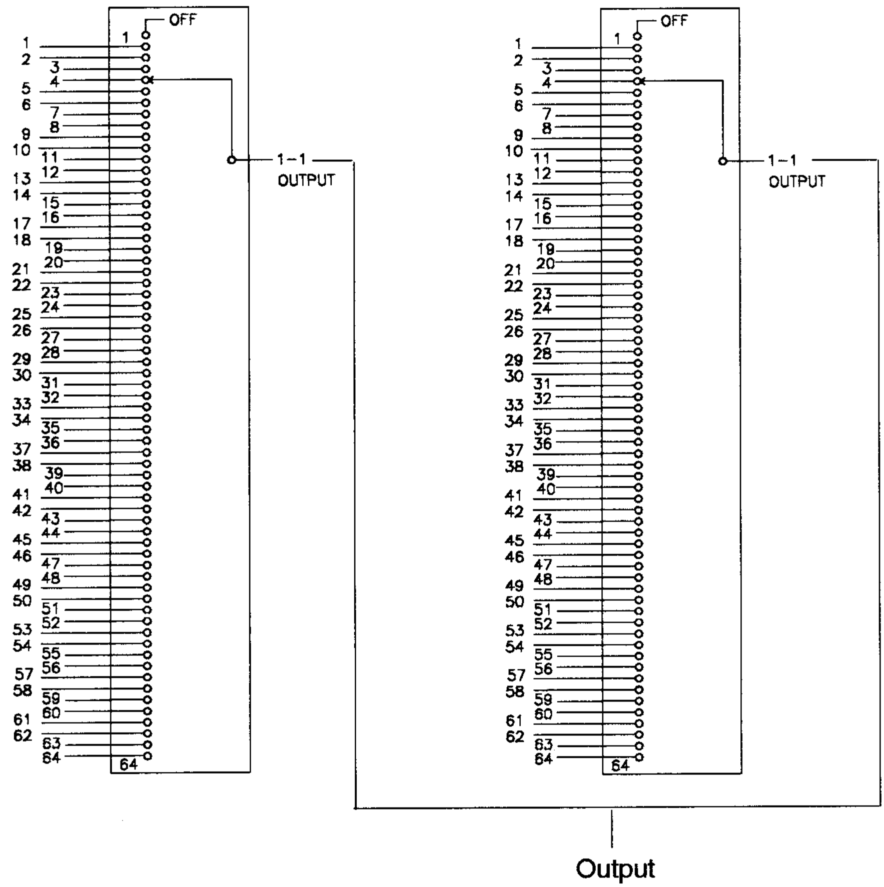

Figure 2.43: Shared Output Operation

2.12 Front Panel Indicators

Indicator lights on the Mux front panel display the status of the interface. The function of each indicator is described below.

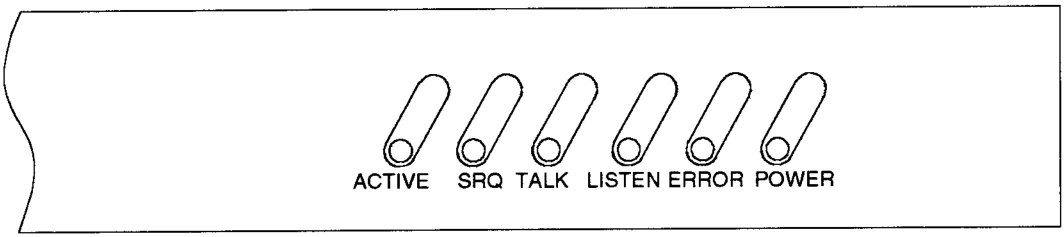

Figure 2.44: Mux488/64 Front Panel Indicator Lights

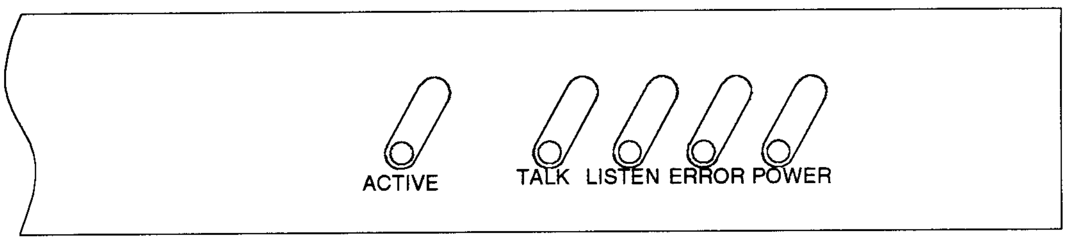

Figure 2.45: Mux/64 Front Panel Indicator Lights

ACTIVE — ACTIVE is lit when the outputs are connected to any of the input groups. It is not lit when the outputs are disconnected from all inputs.

SRQ — (Mux488/64 only) SRQ (Service Request) is lit when the Mux is set as a master and requires the attention of the RS-232C or IEEE 488 controller. The SRQ light follows the internal Master Summary Status (MSS), which is maintained by the status reporting functions of the interface firmware (see Section 6.7.4 for more discussion).

TALK
If the unit is an IEEE 488 master, TALK is lit when Mux488/64 has been addressed to talk, and is not lit when it is not. It does not reflect actual data transmission.
If the unit is an RS-232C master, TALK is lit while characters are being transmitted out the RS-232C port.
On a slave device, TALK is lit while characters are being transmitted out the master/slave port.
TALK is not used in LPT

LISTEN
If the system configuration is IEEE 488 master, LISTEN is lit when Mux has been addressed to listen, and is not lit when it is not. It does not reflect actual data transmission.
If the system configuration is RS-232C master, LISTEN is lit while characters are being received from the RS-232C port.
When LPT port

ERROR
On when an error has occurred, off when no error condition exists. ERROR is also used, in combination with other indicators, to display various self test results as described in Section 2.13.

POWER
On when power is applied to the Mux and the power switch on the rear panel is in the on position (depressed). Off if power is not present. POWER indicates that the digital 5 volt supply is operating.

2.13 Power-Up

At initial power-up or on Reset (*R command), the Mux performs several automatic self-tests to ensure that it is fully functional. The indicator lights on the Mux front panel show any errors if they occur. Possible error conditions and their corresponding indicator light patterns are shown in the following table. Any pattern not shown is an internal error that is not field-serviceable – contact the factory. If ERROR is on by itself, there is a master/slave communication error. Check the error using the Error Query (E?) command (see the Command Description Section, 6 for more information).

Failure	SRQ	Talk	Listen	Error	Power
General Hardware Failure	○	○	○	○	○
Communication or NVRAM Error	●	●	●	○	○
RAM Error	●	○	●	☼	○
ROM Checksum Error	●	●	○	☼	○
Invalid Switch Setting	●	○	○	☼	○

● Indicator light off. ○ Indicator light on. ☼ Indicator light flashing.

To ensure initialization of the slave interfaces, the first command sent to the master unit should be a *R. This causes a reset and establishes communication with the slave units. This is not necessary if the master unit was powered on after the slaves were.

Master/Slave Operation

This section describes Mux master/slave operation. In master/slave operation, a master unit and any attached slave units are controlled using an IEEE 488 interface or an RS-232C port. The master unit in the system must be a Mux488/64. The optional slave units can be either Mux/64s or Mux488/64s configured as slave units. See Section 2.6 for IEEE 488 operation setup or Section 2.7 for RS-232C operation setup to configure the units properly for master/slave operation.

By providing communication between a single master unit and up to fifteen slave units, the master/slave configuration expands a single IEEE 488 bus address or RS-232C port to control a total of sixteen units with up to 1024 signals.

Each Mux in a master/slave system is identified by a unique identification number. Master units are always unit 1. Slave units' identification numbers can be any number from 2 to 16, as set by the rear panel switches (see Section 2 for switch settings).

3.1 System Reset

Before using the Muxes, it is important to reset the system to ensure that the master unit and any attached slave units are reset and functioning. To reset the system, send the reset command:

```
PRINT#1,"OUTPUT 15;*RX"
```

and then wait five seconds for the reset to take effect.

Once reset, the slave units' ERROR indicator lights should be off and their TALK and LISTEN indicators should light as they communicate with the master.

The master unit's ERROR indicator should be off. However, it is on if an error occurred during reset. The Error Query (E?) and Query Attached Units (U4) commands are used to diagnose the source of the error. For example:

```
OPEN "IEEEIN" FOR INPUT AS #1
IOCTL#1,"BREAK"
PRINT#1,"RESET"                      Attach and reset the IEEE 488 interface.
OPEN "IEEEOUT" FOR OUTPUT AS #2
PRINT#1,"TERM LF EOI"     Use line-feed and EOI for terminator.
PRINT#1,"OUTPUT 15;E? X"
PRINT#1,"ENTER 15"        Retrieve response.
INPUT#2,E
```

`PRINT E`	Displays 64 if communication error occurred.
`PRINT#1,"OUTPUT 15;U4X"`	Query attached units.
`PRINT#1,"ENTER 15"`	Retrieve response.
`LINE INPUT#2,A$`	
`PRINT A$`	Displays 01,nn,nn,nn... showing which units are and are not recognized.

The above example assumes the Mux is attached to an IOtech IEEE 488 interface and is programmed in BASIC. If an RS-232C interface is used to communicate with the Mux, the equivalent of the above example in BASIC is:

```
OPEN "COM1:9600,N,8,1,CS,DS" FOR RANDOM AS #1
```

	Attach the RS-232C interface.
`PRINT#1,"E?X"`	Query attached units.
`INPUT#1,E`	Retrieve response.
`PRINT E`	Displays 64 if communication error occurred.
`PRINT#1,"U4X"`	Query attached units.
`LINE INPUT #2,A$`	Retrieve response.
`PRINT A$`	Displays 01,nn,nn,nn... showing which units are and are not recognized.

3.2 Control of Switch Settings

Once the Muxes have been reset, their outputs can be switched among their inputs. The Set Switch Setting (Cn) and Query Switch Setting (C?) commands control the input/output connections of the Muxes. The Select Unit (Pn) command chooses which unit's (master or a specific slave) switch setting is to be affected or interrogated.

If the Select Unit (`Pn`) command has not been specified since power-on or Reset (`*R` command), the master unit's switches are affected. If a slave unit has been specified (with `Pn`, where `n` is the slave identification number), that slave will be affected.

The Set Switch Setting (`Cn`) command connects the outputs of the current unit to the input signals. The `n` in the `Cn` command is a number that specifies the input signal group that is to be connected to the outputs. If `n` is zero, the outputs are disconnected from any of the inputs. Otherwise, it specifies which group of inputs are to be connected to the outputs. For example, if the channel grouping rear-panel switches are set for 4-output mode, the inputs are divided into sixteen groups. Group 1 consists of inputs 1, 2, 3, and 4. Group 2 consists of inputs 5, 6, 7, and 8. Each higher-numbered group consists of the next four consecutive channels up through the last group, group 16, which consists of channels 61, 62, 63, and 64. If `C2` is executed in this configuration, group 2 is connected to the outputs, connecting input 5 to output 4-1, input 6 to output 4-2, input 7 to output 4-3, and input 8 to output 4-4.

`Cn` commands do not take effect until the Execute (`X`) command is interpreted, so `Cn` is called a deferred command. If a syntax error is detected before the `X` is reached, `Cn` commands have no effect. In contrast, Select Unit (`Pn`) is an immediate command that immediately affects all subsequent commands, including subsequent deferred commands such as `Cn`. This allows several units to be referred to in a single command string (as in `P1 C4 P2 C5 X`). See Section 6.3 of the Command Description section for more information on immediate and deferred commands.

`PRINT#1,"OUTPUT 15;C0 X"`	Disconnect outputs.
`PRINT#1,"OUTPUT 15;C1 X"`	Connect outputs to input group 1.
`PRINT#1,"OUTPUT 15;C5 X"`	Connect outputs to input group 5.
`PRINT#1,"OUTPUT 15;C1 C0 X"`	
	Disconnect all outputs, because the `C0` command overrides the preceding `Cn` settings for the selected unit.
`PRINT#1,"OUTPUT 15;P1 C1 P2 C2 X"`	
	Connect master outputs to master input group 1 and slave 2 outputs to slave 2 input group 2.

The Set Switch Setting query (`C?`) command interrogates the output setting of the current unit.

`PRINT#1,"OUTPUT 15;C0 X C? X"`	
	Disconnect outputs.
`PRINT#1,"ENTER 15"`	Read the response.
`INPUT#2,A$`	
`PRINT A$`	Display response of `C00000`.
`PRINT#1,"OUTPUT 15;C13 C? X"`	

Response is still C000000 because the Set Switch Setting command is not executed until the X is interpreted, while the C?, like all other queries, is executed immediately upon interpretation.

`PRINT#1,"OUTPUT 15;C? X"` Response is now C00013.

Because the Cn command takes some time to execute, it may be necessary to confirm that the outputs have switched before continuing with other operations such as digitizing the switched signals. To do this, perform a query command, typically C? or E?, after executing the Cn. The query response is not generated until after the switches have been set and so other operations may proceed.

`PRINT#1,"OUTPUT 15;C0 X C? X"`	
	Disconnect outputs.
`PRINT#1,"ENTER 15"`	Read the response.
`INPUT#2,A$`	
`PRINT A$`	Display response of C00000.

3.3 Sequenced Operation

In addition to the ability to connect and disconnect the outputs, the Mux can automatically step through a list of switch settings. This is called sequenced operation. During sequenced operation, the master unit senses trigger events, then steps itself and the slaves through their sequence settings tables. Each unit's sequence settings table consists of 1100 elements numbered from 1 to 1100. Each element contains the switch settings for that unit for that point in sequenced operation.

In addition to its sequence settings table, the master unit has a sequence duration table. This table contains the duration (the number of triggers) that each corresponding element in the units' sequence settings tables is held. The sequence duration table consists of 1100 elements, numbered 1 to 1100.

To illustrate the use of sequenced operation, assume that the master unit and slave unit 2 are both configured for sixteen outputs and are configured for separate (not shared) operation. The outputs are each to be connected in turn to various input signal groups, repeated, and then all of the outputs are to be deactivated, stepping to the next switch setting every ¼ second (250 ms). The following table illustrates the sequence:

Time (sec)	Master Switch Setting	Slave Switch Setting
0.000	Input Group 1	Input Group 4
0.250	Input Group 2	Input Group 3
0.500	Input Group 3	Input Group 2
0.750	Input Group 4	Input Group 1
1.000	Input Group 1	Input Group 4
1.250	Input Group 2	Input Group 3
1.500	Input Group 3	Input Group 2
1.750	Input Group 4	Input Group 1
2.000	Off	Off

The Muxes accomplish this sequence by stepping through their sequence settings table at intervals generated by their internal timebase generators. The first step in preparing for this sequence is setting up the sequence tables.

3.3.1 Setting the Sequence Tables

Elements in the sequence settings tables can be set with the Set Sequence Switch Setting (`In`) command. Elements in the master unit's sequence duration table can be set with the Sequence Duration (`Jn`) command. The Sequence Table Location (`Wn`) command is used to specify which sequence table element is to be affected or interrogated. The Select Unit (`Pn`) command chooses which unit's (master or a specific slave) sequence settings table is to be affected or interrogated. The Sequence Duration (`Jn`) command and table is shared among all units, and is independent of the sequence settings table and Select Unit (`Pn`) commands.

To set or interrogate an element in the sequence setting or duration tables, the Sequence Table Location (`Wn`) command must be used to set the element to be interrogated or set. Once the sequence table location is defined, `W$` can be used to increment the sequence table location by one location. If the location is incremented past the maximum value of 1100, an execution error occurs. If this occurs, reset the location to a lower value with `Wn`.

The Set Sequence Switch Setting (`In`) specifies which input group, if any, is to be connected to the outputs of the current unit (as specified by `Pn`) at a location in the sequence settings table determined by `Wn`. The `n` in the `In` command is a number that specifies the input signal group to be connected to the outputs. If `n` is zero, the outputs are disconnected from any of the inputs. Otherwise, it specifies the group of inputs to be connected to the outputs.

To set the sequence settings table for the example, use the following commands:

`PRINT#1,"OUTPUT 15;W1 X"` Specify the first element in the sequence settings and duration tables.

`PRINT#1,"OUTPUT 15;P1 I1 X"`

Connect master outputs to master input group 1.

`PRINT#1,"OUTPUT 15;P2 I4 X"`

Connect slave 2 outputs to slave 2 input group 4.

```
PRINT#1,"OUTPUT 15;W$ X"  Increment the sequence table location.
PRINT#1,"OUTPUT 15;P1 I2 P2 I3 X"
                          Connect master outputs to input group 2, slave outputs
                          to group 3.
PRINT#1,"OUTPUT 15;W$ P1 I3 P2 I2 X"
                          Connect master outputs to input group 2, slave outputs
                          to group 3.
PRINT#1,"OUTPUT 15;W$ P1 I4 P2 I1 X"
                          Connect master outputs to input group 2, slave outputs
                          to group 3.
PRINT#1,"OUTPUT 15;W$ P1 I0 P2 I0 X"
                          Disconnect all outputs in both master and slave units.
```

Now sequence settings table elements 1 through 5 contain the desired switch settings. Elements 1 through 4 specify the switching among the input groups, which are repeated twice, and element 5 is set to the final setting, all outputs disconnected. This is shown in Figure 3.1.

Element (location) selected in common by Wn.

Sequence Settings Table			Sequence Duration Table
Element	**Master Unit Channels**	**Slave Unit Channels**	**Common to Both Units**
1	Input Group 1 (I1)	Input Group 4 (I4)	1 (J1)
2	Input Group 2 (I2)	Input Group 3 (I3)	1 (J1)
3	Input Group 3 (I3)	Input Group 2 (I2)	1 (J1)
4	Input Group 4 (I4)	Input Group 1 (I1)	1 (J1)
5	All Disconnected	All Disconnected	Not Set
6	Not Set	Not Set	Not Set
. . .	. . .	. . .	. . .
1100	Not Set	Not Set	Not Set

s (element 1), e (element 4), f (element 5): Qs,e,f Sequence Range

Figure 3.1: Sequence Settings & Duration Tables

The Set Sequence Outputs query (I?) command interrogates the output setting of the current unit.

```
PRINT#1,"OUTPUT 15;W1 P1 I? P2 I? X"
```

Interrogate master and slave sequence settings table element 1.

```
PRINT#1,"ENTER 15"
INPUT#2,A$
PRINT A$
```

Response is I00001I00004, showing that the master outputs are connected to input group 1 and that the slave outputs are connected to input group 4.

```
PRINT#1,"OUTPUT15;W20 I3 I? X"
```

Response is I00003 because the Set Sequence Outputs command is executed immediately, before the I?.

```
PRINT#1,"OUTPUT15;W20 I? X"
```

Response is still I00003.

After setting the channels for a sequence table location, set the number of triggers that must be received before moving to the next sequence table location. The Sequence Duration (Jn) command sets this, where n is the number of triggers from 1 to 65,535. The Mux continues to output from that location until n numbers of triggers are received. In the example, each step takes the same time and so, assuming that the trigger source will be set at the desired rate of four per second, all of the durations can be set to one trigger interval.

```
PRINT#1,"OUTPUT 15;W1 J1 W$ J1 W$ J1 W$ J1 X"
```

No duration is specified for the final switch setting (which is all outputs disconnected at sequence table position 5). This is because the final output does not use its duration: it lasts forever (or until the output settings are explicitly changed).

Once the sequence settings and durations are set, they can be examined. Several commands can be used to look at the sequence table settings.

The Sequence Table Location query (W?) command responds with Wnnnnn, where nnnnn is the present sequence table location.

The sequence duration settings of a unit in the system are queried with U3. The response is in the form:

```
W00001Jnnnnn XW$Jnnnnn X   .   .   .   W$Jnnnnn X
```

This response is the entire sequence table for a unit, in one long string of thousands of characters.

To query the sequence duration settings for a section of the sequence duration table, use U7. This responds with the same response as U3, except the start and end of the response are defined by the start and end parameters of the Sequence Range (Qs,e,f) command.

To query the duration at a single location in the sequence duration table, use J?. The response is in the form Jnnnnn.

The sequence settings table of a unit in the system are queried with U6. The response is in the form:

```
Pn X   W00000Innnnn XW$Innnnn X   .   .   .   W$Innnnn X
```

This response is the entire sequence table for a unit, in one long string of thousands of characters.

To query a section of the sequence settings table of a unit in the system, use U8. This responds with the same response as U6, except the start and end of the response are defined by the Qs,e,f command.

3.3.2 Setting the Sequence Range

Once the sequence table has been initialized, the sequence range must be set. The sequence range is the portion of the sequence table used for sequenced operation. It consists of two parts, a range of settings, and a final setting. The range is the portion of the sequence table to be repeated, one or more times, during sequenced operation. The final setting is the sequence table element that takes effect after the range is complete. In the example, the range, which consists of sequence table elements 1 through 4, is to be repeated twice, and the final setting is element 5. This sequence range is specified with the Sequence Range (`Qs,e,f`) command:

```
PRINT#1,"OUTPUT 15;Q1,4,5 X"
```

The Sequence Range (`Qs,e,f`) command defines which part of the sequence settings tables is used for sequenced operation. `s` is the start of the sequence range (element 1 in the example). `e` is the end of the range (4 in the example) and must be greater than or equal to `s`. `f` is the final element (5 in the example), which takes effect when sequenced operation is complete.

The ranges for `s`, `e` and `f` are 1 to 1100.

If the final element is the same as the end of the range (`f` is the same as `e`), sequenced operation stops at the end of the range and does not proceed to the final element and repeat the ending setting. For example, `Q1,3,5` produces the sequence 1,2,3,5. But `Q1,3,3` does not produce the sequence 1,2,3,3. It stops at the ending element and produces 1,2,3.

If a sequence range has not been specified with the `Qs,e,f` command, a default range is used for sequenced operation. The default range is based on the sequence table location, which is set or incremented with the `Wn` command. Mux keeps track of the greatest sequence table location that has been referenced and uses that location as the sequence range end and final elements. For example, at power-on or Reset (`*R`), the default range is `Q1,1,1`. In the example, elements up through 5 have been referenced so the default range is `Q1,5,5`.

For the example, the sequence range is:

```
PRINT#1,"OUTPUT 15;Q1,4,5
```

3.3.3 Setting the Sequence Mode and Count

The Sequence Mode (`Ln`) command specifies how and if the sequence range is to be repeated. `L0` disables any sequenced operation. `L1` specifies that the sequence range is stepped through exactly once. `L2` specifies that the sequence range will be repeated the number of times specified by the Sequence Repetition (`Kn`) command. `L3` specifies that the sequence range is repeated indefinitely until it is stopped by switching to another `Ln` command mode or until triggering is disabled.

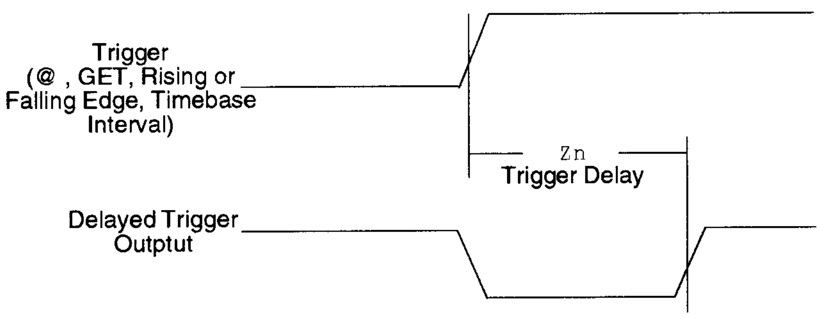

Figure 3.2: Delayed Trigger Output Timing Diagram

The Sequence Repetition (Kn) command specifies the repetition count (from 1 to 65535) for the L2 mode.

In the example, the sequence range (from 1 to 4) is to be repeated twice:

```
PRINT#1,"OUTPUT 15;K2 L2 X"
```

Now that the sequence settings, durations, range, mode and repetition count have been specified, sequenced operation begins as soon as triggers are detected.

3.3.4 Setting the Trigger Source

During sequenced operation, the Mux steps through the sequence table as trigger occur. Triggers can be generated by the IEEE 488 Group Execute Trigger (GET), rising and/or falling edges on the trigger input connector, the Command Trigger (@) or by the Mux's internal timebase generator. In this example, the sequence is to be stepped through every ¼ second, so the internal timebase is used, set for a rate of 250 milliseconds with the Interval Timer (Yn) command:

```
PRINT#1,"OUTPUT 15;Y250 X"
```

Now set the trigger source to accept triggers from the internal timebase and sequenced operation commences:

```
PRINT#1,"OUTPUT 15; T6 X"
```

Other possible trigger sources include IEEE 488 bus Group Execute Trigger (GET) commands, rising or falling edges on the trigger input connector, or the Command Trigger (@).

When sequenced operation completes, the trigger source is reset to no triggering (T0). The trigger source can be interrogated with the Trigger Query (T?) command:

`PRINT#1,"OUTPUT 15;T? X"`	Query trigger source.
`PRINT#1,"ENTER 15"`	Retrieve response.
`INPUT#2,A$`	
`PRINT A$`	Response is T00 if sequenced operation is complete.

The rear-panel BNC TRIG OUT connector carries an active low TTL signal which goes active (low) when a new sequence setting element takes effect. This signal goes inactive (high) after the amount of time specified by the Trigger Delay (Zn) command. The trigger out signal can be used to indicate to other equipment that a new sequence setting has taken effect. In the example, trigger out pulses the default 100 microseconds and there are a total of seventeen pulses. Nine of the pulses are for the two repetitions of eight settings in the sequence, plus one more pulse for the final setting.

3.4 Sequenced Operation Example Recap

This is a summary of the commands used in the example of sequenced operation used in the preceding sections.

`PRINT#1,"OUTPUT 15;W1 X"`	Specify the first element in the sequence settings and duration tables.
`PRINT#1,"OUTPUT 15;P1 I1 X"`	Connect master outputs to master input group 1.
`PRINT#1,"OUTPUT 15;P2 I4 X"`	Connect slave 2 outputs to slave 2 input group 4.
`PRINT#1,"OUTPUT 15;W$ X"`	Increment the sequence table location.
`PRINT#1,"OUTPUT 15;P1 I2 P2 I3 X"`	Connect master outputs to input group 2, slave outputs to group 3.
`PRINT#1,"OUTPUT 15;W$ P1 I3 P2 I2 X"`	Connect master outputs to input group 2, slave outputs to group 3.
`PRINT#1,"OUTPUT 15;W$ P1 I4 P2 I1 X"`	Connect master outputs to input group 2, slave outputs to group 3.
`PRINT#1,"OUTPUT 15;W$ P1 I0 P2 I0 X"`	Disconnect all outputs in both master and slave units.
`PRINT#1,"OUTPUT 15;W1 P1 I? P2 I? X"`	Interrogate master and slave sequence settings table element 1.
`PRINT#1,"ENTER 15"`	
`INPUT#2,A$`	
`PRINT A$`	Response is `I00001I00004`, showing that the master outputs are connected to input group 1 and that the slave outputs are connected to input group 4.
`PRINT#1,"OUTPUT15;W20 I3 I? X"`	Response is `I00003` because the Set Sequence Outputs command is executed immediately, before the `I?`.
`PRINT#1,"OUTPUT15;W20 I? X"`	Response is still `I00003`.
`PRINT#1,"OUTPUT 15;W1 J1 W$ J1 W$ J1 W$ J1 X"`	Set all of the durations to one trigger interval.
`PRINT#1,"OUTPUT 15;Q1,4,5 X"`	Set sequence range.

`PRINT#1,"OUTPUT 15;K2 L2 X"`	Set sequence mode to be repeated, sequence repetition to two.
`PRINT#1,"OUTPUT 15;Y250 X"`	Set trigger delay to 250 milliseconds (¼ second).
`PRINT#1,"OUTPUT 15; T6 X"`	Set trigger source to internal timebase. Sequenced operation begins.

3.5 Error Handling

When the Mux detects an error, it lights the ERROR indicator, sets the appropriate error code in the error status register, discards any deferred commands that were to be executed, and skips all further commands until it has interpreted an `X` command.

The Error Query (`E?`) command responds with the present error condition and clears it. The possible error conditions are:

`E000`	No error has occurred.
`E001`	Invalid device dependent command (IDDC).
`E002`	Invalid device dependent command option (IDDCO).
`E004`	No such unit.
`E008`	Self-test failure.
`E016`	Trigger overrun.
`E064`	Master/slave communication error.
`E128`	Conflict error.
`Ennn`	If two or more errors occurred, `nnn` is the sum of those errors.

These errors also set bits in the Standard Event Status Register. The IDDC error sets the Command Error bit, the Conflict error sets the Device Dependent Error bit, and the other errors all set the Execution Error bit. The Standard Event and Status Byte Registers can be used to generate service requests (SRQs) and are described in detail in Section 5.9.

3.6 Power-on and Reset

The Reset (`*R`) command restores the master unit and any attached slave units to their initial power-up states.

This command has the same effect as removing and re-applying power and is not processed until an Execute (`X`) command is received. Resetting includes initializing the sequence settings and durations tables as well as switching all units to disconnect their outputs. The initial power-up state is described in detail in a later section.

When the master unit is reset, either as a result of a power cycle or a reset command, all attached slave units are also reset. This keeps the master and slaves in a consistent state.

The IEEE 488 bus Device Clear commands (DCL and SDC) do not perform a reset. These commands clear any pending commands and responses and prepare the master to receive new commands.

3.7 IEEE 488 Interface

The IEEE 488 interface provides IEEE 488 peripheral-only functions. It cannot act as a bus controller and cannot accept control.

3.7.1 IEEE 488 Addressing

Mux488/64 uses a single IEEE 488 bus primary address in the range from 0 through 30. Secondary addressing is not used. If the rear-panel switches are set for IEEE 488 bus address of 31, address 30 is used.

3.7.2 IEEE 488 Bus Implementation

The Mux488/64 implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following sections. The Mux488/64 does not support or respond to the bus uniline and multiline commands:

Remote Enable (REN)	Parallel Poll (PP)
Go to Local (GTL)	Parallel Poll Configure (PPC)
Local Lockout (LLO)	Parallel Poll Unconfigure (PPU)
Take Control (TCT)	Parallel Poll Disable (PPD)

3.7.3 My Talk Address (MTA)

When the Mux488/64 is addressed to talk, it replies with whatever response strings have been requested by previous commands. If no requests for response strings have been issued, no data are returned. This may cause a time out error on the IEEE 488 bus controller.

If a serial poll enable (SPE) has been received, it responds with the serial poll status byte, rather than the response strings.

3.7.4 My Listen Address (MLA)

When the Mux488/64 is addressed to listen, it accepts characters from the active talker and interprets these characters as commands and command parameters. These commands are explained in Section 6.

3.7.5 Device Clear (DCL and SDC)

Device clear clears the command input buffer, the command response queue and any pending commands.

3.7.6 Group Execute Trigger (GET)

When the Mux488/64 recognizes a GET, and the Trigger Source command is set for GET (`T1` or `T7`), it performs the specified sequenced operation, if any.

3.7.7 Interface Clear (IFC)

IFC places the Mux488/64 in the Talker/Listener Idle state.

3.7.8 Serial Poll Enable (SPE)

When serial poll is enabled, the Mux488/64 sets itself to respond to a serial poll with its serial poll status byte if addressed to talk. When the serial poll status byte is accepted by the controller, any pending Service Requests (SRQs) are cleared. The Mux488/64 continues to try to output its serial poll response until it is serial poll disabled by the controller.

3.7.9 Serial Poll Disable (SPD)

Disables the Mux488/64 from responding to serial polls by the controller.

3.7.10 Unlisten (UNL)

UNL places the Mux488/64 in the Listener Idle state.

3.7.11 Untalk (UNT)

UNT places the Mux488/64 in the Talker Idle state.

3.7.12 Serial Poll Response

Whenever the Mux488/64 generates a service request (SRQ), a serial poll responds with a serial poll status byte of at least 64 (decimal), showing that the SRQ was generated by the Mux488/64. For complete details on SRQ generation, see Section 6.7, the Serial Poll model.

Note: Service Requests are only generated for those conditions for which their corresponding enable mask bits have been set. See Section 6.7 for details.

3.7.13 IEEE 488 Bus Terminators

Responses from a Mux488/64's IEEE 488 port are terminated with a line-feed with EOI (End-Or-Identify) asserted. Line-feed is ASCII 10, also referred to as new-line. The Response Terminator (Dn) command can be used to change the terminator.

Commands to a Mux488/64's IEEE 488 port must be terminated with the X command. Mux488/64 generally treats control characters, such as carriage-return and line-feed as white space and ignores them unless they are within a command or command option.

3.8 RS-232C Implementation

3.8.1 RS-232C Pinout

The RS-232C interface is connected through a DB-9 plug connector with the following pin-out. Refer to Appendix C for PC-compatible cable pin-outs.

Pin	Description
1	NC
2	Receive Data
3	Transmit Data
4	Ready To Receive Output (DTR)
5	Ground
6	NC
7	Request To Send (RTS) (Held High)
8	Clear To Send Input (CTS)
9	NC

3.8.2 RS-232C Data Format

The RS-232C interface supports the following data formats:

300, 600, 1200, 2400, 4800, 9600, 19200, or 38400 baud.
Odd, Even, or None Parity

Sequenced operation reduces the maximum recommended baud rate to 9600. 19200 and 38400 baud are not recommended if sequenced operation is used.

3.8.3 RS-232C Handshaking

Mux supports both hardware (DTR/CTS) and/or software (XON/XOFF) handshaking.

Hardware handshaking uses the DTR and CTS lines. The DTR output is true when the serial interface is ready to receive data and is false when the serial interface cannot accept more data. The Mux does not send data unless the CTS input is asserted.

Software handshaking uses the XON and XOFF characters to control data flow. The Mux sends an XOFF (ASCII 19) when its input buffer has space for fewer than 1100 characters and sends XON (ASCII 17) when space for more than 2500 characters becomes available. The Mux stops sending data when it receives an XOFF and does not resume until it receives an XON.

3.8.4 RS-232C Terminators

Responses from a Mux488/64's RS-232C port are terminated with a carriage-return followed by a line-feed. Carriage-return is ASCII 13, and line-feed is ASCII 10, also referred to as new-line. The Response Terminator (`Dn`) command can be used to change the terminator.

Commands to a Mux488/64's RS-232C port must be terminated with the `X` command. Mux488/64 generally treats control characters, such as carriage-return and line-feed as white space and ignores them unless they are within a command or command option.

LPT Port Peripheral Mode Operation

The LPT port peripheral mode allows any IBM-PC compatible LPT (printer) port to control the output channels of a single Mux unit. See Section 2.8 for information on configuring a Mux for this mode.

In this mode, the Mux appears to the computer as a standard printer with a Centronics-style parallel interface. It uses the standard DATA, STROBE, BUSY, and ACK signals to transfer data from the computer to the Mux. See Appendix C for the IBM PC LPT port to Mux Master/Slave port (CA-83) wiring information.

The only function supported in LPT mode is switching the outputs among the input groups.

Each switch setting consists of a decimal number from 0 to the number of input groups (as set by the channel-grouping rear-panel switches) followed by white space. White space is all ASCII values of 32 and below, including the space, tab, new-line (line-feed) and carriage-return characters. The Mux interprets the decimal value and connects the outputs to the specified input group; if the setting is 0, the outputs are disconnected.

For example, to disconnect the outputs, and then connect them to input group 5, use the BASIC statements:

`LPRINT 0;" "`	Disconnect all outputs.
`LPRINT 5;" "`	Connect outputs to input group 5.

If the specified switch setting is too large (greater than the number of input groups), or too small (less than 0), the Mux does not change its switch setting and it lights the ERROR indicator. The ERROR indicator goes off when a valid switch setting has been received.

In many applications, it is necessary to be sure that the Mux has finished switching its outputs before the application continues processing. To do so, the application must send at least two white-space characters after each setting. The first white-space character marks the end of the setting and allows the Mux to switch its outputs. While it is switching, the Mux does not accept the second white-space character. When the outputs have been switched, the Mux accepts and discards the extra white-space, allowing the application to continue, knowing the Mux has acted upon the new switch setting.

In the examples, a space (" ") has been appended to the switch settings to be sure that at least two white-space characters are being sent after each setting. The first is the space, and the second is the carriage-return that is automatically sent by the `LPRINT` command.

4.1 Programming Example

This BASIC program shows the Mux attached to the LPT1 port of a PC-compatible computer. In this configuration, the Mux receives data as a printer would.

```
10 'Disconnect all outputs
20 LPRINT 0;" "
30 '
40 'Connect the outputs to input group 1
50 LPRINT 1;" "
60 '
70 'Connect the outputs to input groups 1 through 16 in succes-
sion.
80 FOR I=1 TO 16
90 LPRINT I;" "
100 NEXT I
```

Digital I/O Peripheral Mode Operation

The Digital I/O peripheral mode allows any digital I/O port with eight outputs (seven data plus one strobe) and one input to control up to fifteen Mux units. See Section 2.4.3 for information on configuring Mux units for this mode.

In this mode, each Mux's master/slave port has nine active lines: seven data inputs, one strobe input, and one ready output. The data lines are active high and carry 7-bit ASCII characters to the units. The strobe line is active low. It should be asserted (low) for at least one microsecond after the data lines have been set with the ASCII character, and then unasserted (high) to latch the character into the units. The ready line is active high. It goes inactive (low) when a character has been latched into the units and will not go active (high) until all of the units have processed that character.

This type of interface is very similar to a Centronics-style printer interface with two major exceptions: an active-high ready line is used instead of an active-high busy line, and the ACK line is not used. This similarity allows an IBM PC LPT port to command multiple Mux units in this mode, although special software is required (see the example below).

The Digital I/O peripheral mode can also be driven by digital I/O interfaces such as those on the IOtech Power488, the MetraByte PIO-12 or DAS-16, or the Data Translation DT2801.

The number of Muxes that can be connected simultaneously to a single digital I/O interface is limited by the current drive capability of the interface. Each Mux has an input current of 1.6 mA at 0.4 V. A standard LPT port, using 74LS374 drivers, or their equivalent, can sink 12 mA and so can drive about eight Mux units. An IOtech Power488 uses an 8255 I/O driver which can only sink 2.5 mA and so can drive one or at most two Mux units.

The digital I/O peripheral mode sets the units' switches by directing commands to individual Mux units, referred to by their slave identification numbers (numbers from 2 to 16).

Each switch setting consists of a decimal number from 0 to the number of input groups (as set by the channel grouping rear panel switches), followed by white space. White space is all ASCII values of 32 and below, including the space, tab, new-line (line-feed) and carriage-return characters. The Mux interprets the decimal value and connects the outputs to the specified input group; if the setting is zero, the outputs are disconnected.

The following protocol is used to send output settings to one or more digital I/O peripheral mode units:

1. Send the addresses of the units that are to receive the command. An address is a single character with an ASCII value equal to the slave identification plus 64. In BASIC, this is `CHR$(64+slaveID)`.
2. Send the new output setting as a string of ASCII characters. In BASIC, this is `STR$(setting)`.
3. Send carriage-return (`CHR$(13)`), line-feed (`CHR$(10)`) or white space.
4. Optional – send additional white space to wait for the Mux to act on the new setting.

If the specified switch setting is too large (greater than the number of input groups), or too small (less than zero), the Mux does not change its switch setting and it lights the ERROR indicator. The ERROR indicator turns off when a valid switch setting has been received.

In many applications, it is necessary to be sure that the Mux has finished switching its outputs before the application continues processing. To do so, the application must send at least two white-space characters after each setting (steps 3 and 4 above). The first white-space character marks the end of the setting and allows the Mux to switch its outputs. While it is switching, the Mux will not accept the second white-space character. When the outputs have been switched, the Mux accepts and discards the extra white-space allowing the application to continue, knowing the Mux has acted upon the new switch setting.

To implement shared operation in Digital I/O Peripheral Mode, the following steps must be taken:

1. Each unit must be configured with the same channel grouping so that Muxes in the system all have the same number of active outputs. The output configuration (shared/separate) switch has no affect in Digital I/O Peripheral Mode.
2. Disconnect the current inputs (if any) by sending a switch setting of zero to the appropriate unit.
3. Connect the new inputs by sending a new switch setting to the appropriate unit.
4. Repeat steps 2 and 3, disconnecting the outputs before reconnecting them, each time the outputs are to be switched. This guarantees break-before-make operation.

5.1 Digital I/O Peripheral Programming Examples

The following BASIC program sends commands to Mux units in Digital I/O peripheral mode connected to the LPT1 port of a PC-compatible computer. As mentioned above, the similarity of the Digital I/O peripheral mode to a standard Centronics-style interface allows the standard PC LPT port hardware to be used to drive the Digital I/O mode. However, the differences do require that different software be used. In this example program, the subroutine starting at line 1000 directly manipulates the LPT port control registers to communicate with the control units. See Appendix C for the IBM PC LPT port to Mux Master/Slave port cable (CA-83) wiring diagram.

```
100 'Find the base port address of LPT1
110 '
115 DEF SEG = 0
120 DATAPORT = (PEEK(&H409) * 256) + PEEK(&H408)
130 STATUSPORT = DATAPORT + 1
140 CONTROLPORT = DATAPORT + 2
145 DEF SEG
150 '
160 'The unit's address plus the address command
170 '
180 UNITADDR = 5 + 64
```

```
185 '
200 'Disconnect all outputs
210 C$ = CHR$(UNITADDR) + "0" : GOSUB 1000
230 '
240 'Connect outputs to input group 1
250 C$ = CHR$(UNITADDR) + "1" : GOSUB 1000
260 '
300 'Connect the outputs to input groups 1 through 16 in succession.
310 FOR I=1 TO 16
320 C$ = CHR$(UNITADDR) + STR$(I) : GOSUB 1000
330 NEXT I
340 '
350 'Connect slave unit 2 outputs to input group 3
360 C$ = CHR$(64 + 2) + "3" : GOSUB 1000
370 '
999 END
1000 'Send the character string in C$ to the Mux unit
1001 'specified by the UNITADDR variable appending white-space.
1002 '
1004 C$ = C$ + SPACE$(2)
1005 FOR C = 1 TO LEN(C$)
1010 IF (INP(STATUSPORT) AND &H80) <> 0 THEN PRINT "Busy";: GOTO
1010
1020 OUT DATAPORT,ASC(MID$(C$,C))
1030 OUT CONTROLPORT,&HD 'pulse strobe
1040 OUT CONTROLPORT,&HC
1050 NEXT C
1060 RETURN
```

The following BASIC program sends commands to Mux units in Digital I/O peripheral mode connected to the digital I/O port of an IOtech Power488 or Power488/CT interface. In this example program, the set-up code at line 100 and the subroutine starting at line 1000 directly manipulate the Power488 8255 I/O chip to communicate with the Mux units. See Appendix C for the Power488 Digital I/O Port Adapter (CA-59) to Mux Master/Slave port cable (CA-84) wiring diagram.

```
100 ' Set the 8255 base address to 02F0
110 '
112 OUT &H8EE1, &HBC
113 '
114 ' Make 8255 port A output, ports B and C input
```

```
115 OUT &H2F3, &H8B
117 '
120 DATAPORT = &H2F0
130 STATUSPORT = &H2F2
150 '
160 'The unit's address plus the address command
170 '
180 UNITADDR = 5 + 64
185 '
200 'Disconnect all outputs
210 C$ = CHR$(UNITADDR) + "0" : GOSUB 1000
230 '
240 'Connect outputs to input group 1
250 C$ = CHR$(UNITADDR) + "1" : GOSUB 1000
260 '
300 'Connect the outputs to input groups 1 through 16 in succession.
310 FOR I=1 TO 16
320 C$ = CHR$(UNITADDR) + STR$(I) : GOSUB 1000
330 NEXT I
340 '
350 'Connect slave unit 2 outputs to input group 3
360 C$ = CHR$(64 + 2) + "3" : GOSUB 1000
370 '
999 END
1000 'Send the character string in C$ to the Mux unit
1001 'specified by the UNITADDR variable appending white-space
1002 '
1004 C$ = C$ + SPACE$(2)
1005 FOR C = 1 TO LEN(C$)
1010 IF (INP(STATUSPORT) AND &H80) = 1 THEN PRINT "Busy";: GOTO 1010
1020 OUT DATAPORT,ASC(MID$(C$,C))
1030 OUT DATAPORT,ASC(MID$(C$,C)) + 128
1050 NEXT C
1060 RETURN
```

Plug-In-Card Port Peripheral Mode Operation

The plug-in-card port peripheral mode allows any digital I/O port with seven outputs and one input to quickly and simply control a single Mux. To set up a system for Plug-in Card Port Peripheral operation, see Section 2.10.

In plug-in card peripheral mode, the Mux is controlled with seven data lines, D0 through D6, and responds on the /ACK line. The Mux constantly monitors the data lines looking for them to change. When any of them change, the Mux waits for them to stop changing and then connects the outputs to the input group specified by the data lines. Then, the Mux toggles the /ACK line; if it was low, it is set high; if it was high, it is set low. When the computer has sensed the transition of the /ACK line, it knows that the outputs have been switched.

Data lines D6 through D0 are interpreted as a binary number with D0 carrying the least-significant bit. If all of the data lines are low, the switch setting is zero and the outputs are disconnected from all of the inputs. Other values select among the allowable switch settings, which depend on the present channel grouping as set by the rear-panel switches. If the switch setting is out of range, the ERROR indicator light is lit.

This type of interface can be driven by almost any digital I/O interface, including the hardware of an IBM PC LPT port and the digital I/O ports of such plug-in data-acquisition cards as the IOtech Power488, the MetraByte PIO-12 or DAS-16, or the Data Translation DT2801.

The following protocol is used to set the switch settings of a Mux in plug-in-card peripheral mode:

1. Note the current level of the /ACK line.
2. Write the new switch setting as a binary number to the data lines.
3. Wait for the /ACK line to change from its noted value.

6.1 Plug-in Card Peripheral Programming Examples

The following BASIC program switches a Mux unit in plug-in-card peripheral mode connected to the LPT1 port of a PC-compatible computer. As mentioned above, the simplicity of the plug-in-card mode allows the use of the standard PC LPT port hardware. However, it does require that different software be used. In this example program, the subroutine starting at line 1000 directly manipulates the LPT port control registers to communicate with the Mux unit. See Appendix C for the IBM PC LPT port to Mux Master/Slave port cable (CA-83) wiring diagram.

```
10 'This program will control the Mux unit when configured
12 'in the PIC interface mode from the PC's LPT1 port.
15 '
100 'Find the base port address of LPT1
```

```
110 '
115 DEF SEG = 0
120 DATAPORT = (PEEK(&H409) * 256) + PEEK(&H408)
130 STATUSPORT = DATAPORT + 1
140 CONTROLPORT = DATAPORT + 2
145 DEF SEG
150 '
200 'Turn off all channels
210 C = 0: GOSUB 1000
230 '
300 'Switch input groups 1 through 16 in succession
305 FOR REPEAT = 1 TO 10
310 FOR I=1 TO 16
320 C = I: GOSUB 1000
330 NEXT I
335 NEXT REPEAT
340 '
999 END
1000 'Send c out the parallel port, looking for /ACK to toggle
1005 IF INP(DATAPORT) = C THEN RETURN 'Do nothing if no change
1010 C1 = INP(STATUSPORT) AND &H40 'Current status value
1020 OUT DATAPORT,C
1030 IF (INP(STATUSPORT) AND &H40) =|C1 THEN PRINT "Busy";: GOTO 1030
1040 RETURN
```

The following BASIC program switches a Mux unit in plug-in-card peripheral mode connected to the digital I/O port of an IOtech Power488 or Power488/CT interface. In this example program, the set-up code at line 100 and the subroutine starting at line 1000 directly manipulate the Power488 8255 I/O chip to communicate with the control units. See Appendix C for the Power488 Digital I/O Port Adapter (CA-59) to Mux Master/Slave port cable (CA-84) wiring diagram.

```
10 'This program will control the Mux unit when configured
12 'in the PIC interface mode from the MP488 digital port.
15 '
100 ' Set the 8255 base address to 02F0
110 '
112 OUT &H8EE1, &HBC
113 '
114 ' Make 8255 port A output, ports B and C input
115 OUT &H2F3, &H8B
117 '
```

```
120 DATAPORT = &H2F0
130 STATUSPORT = &H2F2
150 '
200 'Turn off all channels
210 C = 0: GOSUB 1000
230 '
300 'Switch input groups 1 through 16 in succession
305 FOR REPEAT = 1 TO 10
310 FOR I=1 TO 16
320 C = I: GOSUB 1000
330 NEXT I
335 NEXT REPEAT
340 '
999 END
1000 'Send c out the parallel port, looking for /ACK to toggle
1005 IF INP(DATAPORT) = C THEN RETURN 'Do nothing if no change
1010 C1 = INP(STATUSPORT) AND &H40 'Current status value
1020 OUT DATAPORT,C
1030 IF (INP(STATUSPORT) AND &H40) = C1 THEN PRINT "Busy";:
GOTO 1030
1040 RETURN
```

Command Descriptions

6.1 Overview

The Mux488/64 is controlled by modifying the contents of its internal registers through commands. The relationship between the contents of the registers and the actions taken by Mux488/64 are described in the command descriptions that follow in this section.

There are two types of register-based commands. System commands affect the entire Mux system (including any slave units). Unit commands affect only the unit specified by the Select Unit (`Pn`) command (the current unit).

The System commands are:

Command Trigger (`@`)
Reset (`*R`)
Response Terminator (`Dn`)
Error Query (`E?`)
Sequence Duration (`Jn`)
Sequence Repetition (`Kn`)
Sequence Mode (`Ln`)
SRQ Mask (`Mn`)
Event Mask (`Nn`)
Select Unit (`Pn`)
Sequence Range (`Qs,e,f`)
Trigger Source (`Tn`)
Query Event Status (`U0`)
Query Status Byte (`U1`)
Query System Settings (`U2`)
Query System Buffers (`U3`)
Query Connected Units (`U4`)
Query System Buffer Range (`U7`)
Sequence Table Location (`Wn`)
Execute (`X`)
Timebase Interval (`Yn`)
Trigger Delay (`Zn`)

The unit is selected using the Pn command. The Unit commands are:

Set Switch Setting (Cn)
Set Sequence Switch Setting (In)
Query Unit Settings (U5)
Query Unit Buffers (U6)
Query Unit Buffer Range (U8)
Query Product Name (U9)
Query Output Configuration(U10)

Most commands consist of one alphabetic character followed by one or more numbers. The alphabetic character is the command and the number(s) are the command parameters.

The examples in this section use a personal computer functioning as an IEEE 488 bus controller, using the IOtech Personal488 PC/IEEE 488 board and associated driver software. All examples are given using BASIC. The Mux488/64 bus address is set to 10 for all examples.

In order to establish communication with DRVR488 from BASIC, the following sequence must be used:

```
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
IOCTL#1,"BREAK"
PRINT#1,"RESET"
OPEN "\DEV\IEEEIN" FOR INPUT AS #2
PRINT#1,"TERM IN LF EOI"
```

All of the command examples assume the driver has been properly opened and reset by the above sequence.

6.2 Terminators

Responses from a Mux488/64's IEEE 488 port are terminated by default with a line-feed with EOI (End Or Identify) asserted. Responses from a Mux488/64's RS-232C port are, by default, terminated with a carriage-return followed by a line-feed. The Response Terminator (Dn) command is used to change the terminator(s). Commands to a Mux488/64's IEEE 488 port must be terminated with the X command. Mux488/64 generally treats control characters, such as carriage-return and line-feed as white space and ignores them unless they are within a command or command option.

6.3 Command Interpretation

As commands are received by Mux488/64, they are interpreted in the order in which they are received. Some commands are immediate, which means they immediately take effect. Other commands are deferred, and have no effect on device operation until the execute command (X) is interpreted.

An example of an immediate command is Select Unit (Pn), which immediately chooses which unit is being referred to. The immediate commands are Dn, In, Jn, Pn, Wn and all queries, including Un.

An example of a deferred command is Set Switch Setting (Cn) which connects the outputs to the specified input group when X is interpreted. As deferred commands are interpreted, their desired effects are recorded in internal temporary registers. As additional deferred command are interpreted, their effects are added to these registers, possibly overwriting earlier command's effects. Finally, when X is interpreted, the temporary registers are examined in the execution order described below. If two deferred commands that do not affect the same function are received before the execute command, they take effect in the execution order described below. If two deferred commands are sent that affect the same function, the earlier command is overridden. For example, if C1 C0 X is sent, the outputs are disconnected as specified by the C0 command. The C1 command is overridden and never takes effect.

If an error is detected during command processing, commands are ignored up through and including the next execute command. Thus any immediate commands after the error, as well as all deferred commands, are ignored. For example, the command line C1 W2 AA C3 P4 X containing the error AA only executes the W2, because it is an immediate command that occurred before the error. The deferred commands C1 and C3, and the immediate command P4 after the error, have no effect.

Deferred commands help reduce the effects of errors and improve synchronization between master and slave units. Because the switch setting command Cn) is deferred, all units' switches are simultaneously switched by the execute command. If an error occurs during the specification of the new state, none of the switch settings are modified, keeping the outputs consistent. This is the primary advantage of deferred commands: they are executed as a group, either all or none. If any errors occur, deferred commands have no effect and the device is left in a consistent state instead of a partially modified, inconsistent state.

The deferred commands are Cn, Kn, Ln, Mn, Nn, Qs,e,f, Tn, Yn, Zn, *R and @.

6.4 Command Execution Order

The immediate commands (Dn, In, Jn, Pn, Wn and all queries including Un) take effect immediately when they are interpreted. Even so, they must be followed by an X command to terminate the command string for correct operation. For example: P1 W0 I5 X.

The deferred commands do not take effect until after the X is interpreted. At that time, they are executed in the following order, regardless of the order they are in the command string: Mn, Nn, Cn, Kn, Qs,e,f, Yn, Zn, Ln, Tn, @, *R.

6.5 Syntax Rules

Most commands are identified by a single letter (A through Z) or an asterisk (*) followed by an single letter. The command trigger is the at-sign (@).

6.5.1 Case Sensitivity

Commands can be entered in upper or lower case. For example, the command `C2 X` acts the same as `c2 x`.

6.5.2 Spaces

White space, which consists of all ASCII values of 32 and below and includes the space, tab, new-line (line-feed) and carriage-return characters, is generally allowed anywhere between commands and command arguments. White space is not allowed in the middle of command options (for instance, `1  2  3` is not the same as `123`).

6.5.3 Multiple parameters

If more than one parameter is used for a command, they must be separated by a comma and can be separated by white space.

Examples:

```
PRINT#1,"OUTPUT 15;Q100,400, 10"
```

is a command with three parameters: 100, 400, and 10.

6.5.4 Command Strings

Commands can be sent individually or in a string with other commands.

For example, these three commands:

```
PRINT#1,"OUTPUT 15;P1"
PRINT#1,"OUTPUT 15;C0 X"
PRINT#1,"OUTPUT 15;K2 X"
```

have the same effect as the single command:

```
PRINT#1,"OUTPUT 15;P1 C0 X K2 X"
```

6.5.5 Query Option

Most commands have a corresponding query command formed by appending a question mark (?) to the command letter. Query commands respond with the present configuration or mode of a previously executed command. When appropriate, the response from a query command is in the form of a command string which, if it were executed, would put the unit into the configuration it was in when the query was executed. For example, the response to a `C? X` is a command of the form `Cnnnnn`, where `nnnnn` is the current switch setting. Query responses are always fixed-length strings in a pre-defined format.

Any number of query commands can be combined into one string to create a specialized status command that responds with only the information of interest for a given application. For example, `P? C? X` responds with the current unit number followed immediately by its current setting, such as `P3C00012`. No spaces separate the responses from consecutive queries.

Query commands are immediate, which means their responses are generated as soon as they are interpreted, before any other commands, including `X`. For example:

`PRINT#1,"OUTPUT 15;C0 X C? X"`	Response is `C00000`.
`PRINT#1,"OUTPUT 15;C1 X C? X"`	Response is `C00001`.
`PRINT#1,"OUTPUT 15;C4 C? X"`	Response is still `C00001`, because the `C?` is an immediate command. The `C4` is a deferred command that takes effect at the `X`.
`PRINT#1,"OUTPUT 15;C? X`	Response is now `C00004`.

Even though query commands generate their response as soon as they are interpreted, they must still be followed by the Execute (`X`) command for proper termination.

6.6 Default Configuration

The factory default configuration, which is restored at power-on or by the execution of the restore defaults (*R) command, is equivalent to executing the following commands:

Step	Command	Effect
1	PnC0	Turns off switches for each unit in the system.
2	D2 (RS-232) D4 (IEEE 488)	Set default response terminator to CR-LF (RS-232) or LF EOI (IEEE 488).
3	L0	Disables sequenced operation.
4	M0	Clear service request enable.
5	N0	Clear event enable.
6	K1	Set sequence repetition to 1.
7	Q1,1,1	Set default sequence range to location 1.
8	T0	Disable triggering.
9	Y1	Set the default timebase interval to 1 millisecond.
10	Z0.1	Set the default trigger delay to 100 microseconds.
11	X	Execute the above commands.

The sequence switch settings and sequence duration registers are initialized according to the following rules:

The sequence duration registers are all set to a duration of one trigger event and the sequence switch setting registers are initially set to outputs disconnected:

```
W1 J1 I0 W$ J1 I0
```

This is repeated 1099 times.

In the separate output mode, the first elements of the sequence switch setting registers of all units are set to settings 1 through the number of input groups of that unit. For example, if a master unit and slave units number 2 and 6 were configured with one, two, and four outputs, respectively, the following sequence switch setting is used:

```
P1 W1I1 W$I2 W$I3 . . . W$I64
P2 W1I1 W$I2 W$I3 . . . W$I32
P6 W1I1 W$I2 W$I3 . . . W$I16
```

In the shared output mode, the sequence switch setting registers are configured to switch first through the master's channels, then through the lowest numbered slave's channels, and then through the next slave's channels until all of the channels have been switched.

For example, if a master unit, and slave units with identifications 2 and 6, were configured for shared outputs in the 64-input, 1-output configuration, the following sequence switch setting is used:

```
P1 W1I1 W$I2 W$I3 . . . W$I64
P2 W$I1 W$I2 . . . W$I64
P6 W$I1 W$I2 . . . W$I64
```

This would set the first 192 (3 times 64) sequence switch setting registers to switch, in sequence, through all 192 input signals.

Finally, the sequence address register is reset to 1 after the sequence settings have been initialized.

```
W1 X
```

6.7 Status Reporting

The Mux488/64 can report errors and other events in three ways: by asserting the IEEE 488 bus Service Request (SRQ) line, by setting the status byte register which can be read using a serial poll, or by setting bits in status registers which can be read with query commands. The first two of these methods apply only to IEEE 488 operation, because RS-232 does not provide either service requests or serial polling. Both IEEE 488 and RS-232 can use the third method, in which status registers indicate the errors and events. The status reporting registers are shown in Figures 6.1A and 6.1B, and the commands to set and query them are:

	Register	Set	Query
STB	Status Byte Register	- -	U1 Serial Poll
SRE	Service Request Enable Register	Mn	M?
ESR	Event Status Register	- -	U0
ESE	Event Status Enable Register	Nn	N?
ESC	Error Source Register	- -	E?

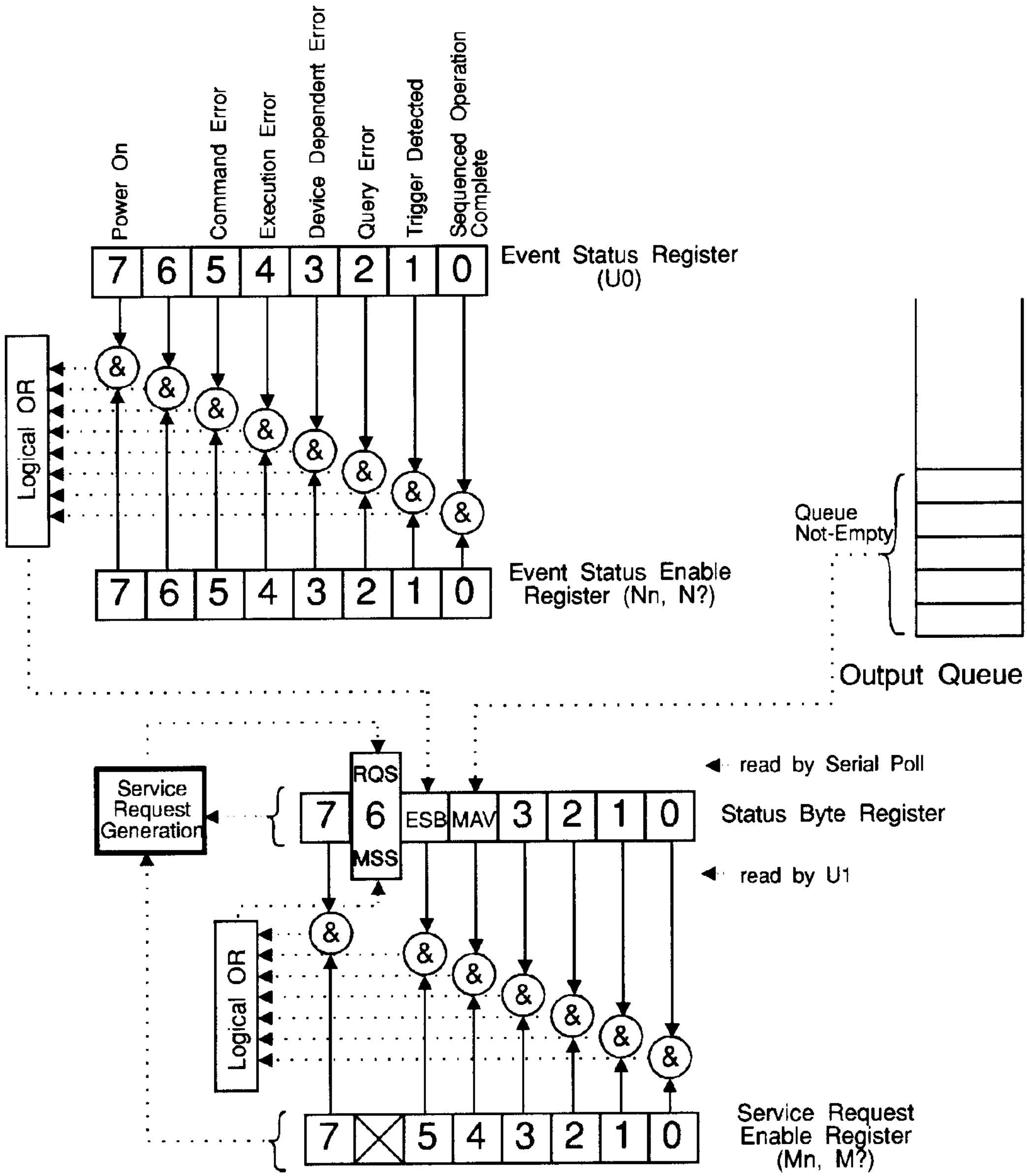

Figure 6.1A: Control488/16 Status Reporting Registers

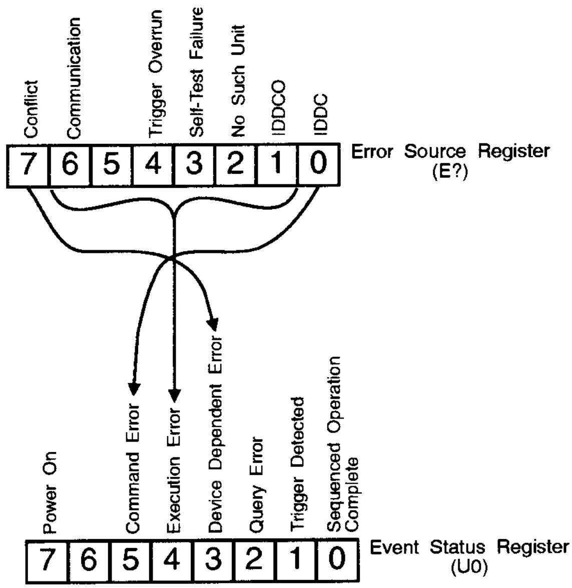

Figure 6.1B: Control488/16 Status Reporting Registers

6.7.1 Service Request Generation

Service requests are generated by the status byte register in combination with the service request enable register. Bits in the status byte register are used to indicate if a response is pending in the Mux488/64 output queue, or if an event from the event status register has been detected. When one of these conditions occurs, they are noted in the status byte register and, if the corresponding bit is set in the service request enable register, a service request is generated.

6.7.2 Event Detection

The status byte register event detected bit (ESB) is set by the event status register in combination with the event status enable register. Bits in the event status register are used to indicate the occurrence of various events such as errors and triggers. When such an event occurs, it is noted in the event status register and, if the corresponding bit is set in the service request enable register, the ESB bit is set in the status byte register. The setting of the ESB bit might, in turn, generate a service request.

6.7.3 Error Detection

The event status register command error (CME), execution error (EXE), and device dependent error (DDE) bits are set by the error source register. Bits in the error source register are used to indicate the occurrence of various errors. When an error occurs, it is noted, and the corresponding bit in the event status register is set. Setting a bit in the event status register might, in turn, cause the ESB bit to be set in the status byte register. This might then cause a service request. Thus an error will generate a service request if both the service request enable register is set to generate a service request on event detection (ESB), and the event status enable register is set to generate an ESB on the appropriate event status bit (CME, EXE, or DDE). For example, to generate a service request on an illegal device dependent command option (IDDCO) error:

`PRINT#1,"OUTPUT 15;N16 X"`	Enable ESB on Execution Error (including IDDCO)
`PRINT#1,"OUTPUT 15;M32 X"`	Enable SRQ on ESB

To check the error status both serial poll and `U1` can be used to read the status byte, and `U0` and `E?` can be used to read the event status and error source, respectively:

`PRINT#1,"SPOLL 15"`	Read status byte with serial poll
`INPUT#2,S`	
`PRINT#1,"OUTPUT 15;U1 X"`	Read status byte with query command
`PRINT#1,"ENTER 15"`	Retrieve the response
`INPUT#2,S`	
`PRINT#1,"OUTPUT 15;U0 X"`	Read event status
`PRINT#1,"ENTER 15"`	Retrieve the response
`INPUT#2,S`	
`PRINT#1,"OUTPUT 15;E? X"`	Read error source
`PRINT#1,"ENTER 15"`	Retrieve the response
`INPUT#2,S`	

6.7.4 Status Byte Register

The Mux488/64 status byte register has only three active bits:

`Bit 4`	Message Available (MAV)	Set when a response is waiting to be read by the controller. Unread responses are stored in an output queue which sets the message available bit when it is not empty.
`Bit 5`	Event Status	Set when the corresponding bits in both the event status register and the event status enable register are set.
`Bit 6`	Request For Service (RQS) Master Summary Status (MSS)	Request for service is returned with the status byte when it is read by the controller using a serial poll. If it is set, then it indicates that the Mux488/64 is requesting service and asserting the IEEE 488 bus Service Request (SRQ) signal. Master summary status is returned with the status byte when it is read with the Read Status Byte `U1` command. If it is set, then it indicates that corresponding bits of the status byte and service request enable registers are set. RQS is cleared when the Mux488/64 is serial polled.

All other bits of the status byte register are unused and are always 0.

6.7.5 Service Request Enable Register

The service request enable register controls which bits of the status byte register are to be reflected in the request for service and master summary status bits of the status byte register. It is set and interrogated with the `Mn` command. As shown in Figure 6.1A, the bits of the status byte register are logically ANDed with the corresponding bits of the service request enable register, and the resulting bits are logically ORed together to form the master summary event status bit (MSS) in the status byte register and to control the request for service (RQS) bit in that register.

6.7.6 Event Status Register

The event status register indicates which events, if any, have occurred. It is read with the `U0` command, which clears the register after responding with its contents. The event status register bits, and the events which set them, are as follows:

`Bit 0`	Sequenced Operation Complete	Set when sequenced operation has been stopped for any reason, including completion of the sequence, execution of an appropriate command (such as `L0`), or an error such as a trigger overrun.
`Bit 1`	Trigger Detected	Set when the Mux488/64 has been configured to accept triggers and the specified trigger has occurred.
`Bit 2`	Query Error	Set when the controller has attempted to read from the Mux488/64 when no response is present or pending, or when a response has been lost because the controller has sent a new query before reading the response to a prior query.
`Bit 3`	Device Dependent Error	Set when a conflict error has occurred. A conflict error is generated when a command cannot execute correctly because it would interfere with other commands or settings.
`Bit 4`	Execution Error	Set when one of several errors have occurred during the execution of a command.
`Bit 5`	Command Error	Set when a command syntax error is detected.
`Bit 6`	Unused	Reserved for future expansion. Always 0.
`Bit 7`	Power On	Set on power-on or system reset (`*R`).

6.7.7 Event Status Enable Register

The event status enable register control which events, if any, are to be reflected in the status byte register. It is set and interrogated with the `Nn` command. As shown in Figure 6.1A, the bits of the event status register are logically ANDed with the corresponding bits of the event status enable register, and the resulting bits are logically ORed together to form the event status bit (ESB) in the status byte register. The event status enable register does not affect the event status register; it only affects the ESB bit of the status byte register.

6.7.8 Error Source Register

The error source register indicates which errors, if any, have occurred. The individual errors are described in the `E?` command.

When an error occurs, it sets the appropriate bit in the error source register. This in turn sets a bit in the event status register as shown in Figure 6.1B.

6.8 Sequenced Operation

The Mux488/64 products include the ability to step through a sequence of switch settings. Each step includes two items: the switch settings that connect the outputs to the inputs, and the number of triggers for which the setting is to persist. The switch settings are stored in the 1100-element sequence settings table elements numbered 1 through 1100 and the number of triggers are stored in the 1100-element sequence duration table elements also numbered 1 through 1100. Each has separate outputs, so the sequence settings registers are present in each individual unit and the select unit command is used to choose among them. Commands are provided to set the sequence switch settings and duration table elements. They directly modify the sequence tables registers and are not deferred.

Command Trigger @

Type: System command, Deferred

@ Enables generation of a command trigger.

The command trigger enables generation of a command trigger. The trigger occurs when an Execute (X) is interpreted. The system responds to command triggers when Trigger Source is set for a command trigger (T5 or T11) and sequenced operation is enabled (Sequence Mode set to L1, L2 or L3). Command trigger has no affect if trigger source is not set to command trigger.

Command trigger is a deferred command; no trigger is generated if an error occurs before the X.

A conflict error occurs if triggering is not enabled (Trigger Source is T0) when the command trigger is interpreted.

When the trigger source is the @ command (T5 or T11), a trigger is enabled when the next Execute command (X) following the @ is encountered in the command line. A trigger is not generated until the entire command line is processed.

```
PRINT#1,"OUTPUT 15;Q1,10,10X"
```

Set the sequence range to include the first ten sequence elements.

```
PRINT#1,"OUTPUT 15;L1 T5X"
```

Execute the sequence once on command triggers.

```
PRINT#1,"OUTPUT 15;@ X"
```

Generate trigger to step through sequence.

```
PRINT#1,"OUTPUT 15;@X@X@X@X@X@X@X@X@X
```

General nine more triggers.

Reset *R

Type: System Command, Deferred

`*R` Restores the master unit and any attached slave units to their initial power-up state.

This command has the same effect as removing and re-applying power. Mux488/64 is returned to its default settings as described in Section 6.6, which includes resetting many of the unit's settings as well as recalling the beginning of the sequence tables from the non-volatile memory.

The reset process takes several seconds, during which the Mux488/64 is unable to receive or process commands. No commands should be addressed to the unit for at least five seconds after issuing the reset command.

When the master unit is reset, either as a result of a power cycle or a reset command, all attached slave units are also reset. This keeps the master and slaves in a consistent state. To ensure initialization of the slave interfaces, the first command sent to the master unit should be a `*R` to establish communication with the slave units. This is not necessary if the master unit was powered on after the slaves were.

The default setting of the Select Unit (`Pn`) command is unit 1, the master unit. All unit commands will refer to the master unit until a different unit is selected.

The IEEE 488 bus Device Clear commands (DCL and SDC) do not have the same effect as `*R`. They do not perform a reset. They do clear any pending commands and responses and prepare the master to receive new commands.

`PRINT#1,"OUTPUT 15;*RX"`	Reset master and attached slaves.
`SLEEP(5)`	Wait for reset to finish.

Set Switch Setting Cn

Type: Unit Command, Deferred (Query is immediate)

`Cn`	Connect the outputs of the unit selected by `Pn` to the input group specified by n.
`C?`	Responds with `Cnnnnn` where `nnnnn` is the present switch setting of the unit specified by the `Pn` command, a five-digit integer.

Set Switch Setting (`Cn`) sets the switches of the unit specified by the `Pn` command to the input group specified by the value of the integer argument n. Switch setting 0 is all outputs disconnected. Settings greater than 0 connect the outputs to the specified input group. The maximum switch setting is the number of input groups set by the channel grouping rear panel switch setting.

In shared operation, only one unit, master or slave, can have its outputs connected to any inputs at a time. To accomplish this, `Cn` first disconnects all units' outputs before making the specified connection. This provides break-before-make switching.

`Cn` is a deferred command. As commands are interpreted, the new switch setting is generated by the `Cn` command. When the `X` is interpreted, the new switch setting is simultaneously set on the master units and all slave units. If an error occurs during the specification of the new state, none of the outputs are modified, keeping the outputs consistent.

A conflict error occurs if triggering is enabled (Trigger Source is not `T0`) when then set switch setting command is interpreted.

Because the `Cn` command takes some time to execute, it may be necessary to confirm that the outputs have switched before continuing with other operations such as digitizing the switched signals. To do this, perform a query command, typically `C?` or `E?`, after executing the `Cn`. The query response is not generated until after the switches have been set and so other operations may proceed.

Examples:

`PRINT#1,"OUTPUT 15;C0 X"`	Disconnect outputs.
`PRINT#1,"OUTPUT 15;C1 X"`	Connect outputs to input group 1.
`PRINT#1,"OUTPUT 15;C5 X"`	Connect outputs to input group 5.
`PRINT#1,"OUTPUT 15;C1 C0 X"`	Disconnect all outputs, because the `C0` command overrides the preceding `Cn` settings for the selected unit.
`PRINT#1,"OUTPUT 15;P1 C1 P2 C2 X"`	Connect master outputs to master input group 1 and slave 2 outputs to slave 2 input group 2.
`PRINT#1,"OUTPUT 15;E? X"`	Query the error source to be sure that no errors have occurred and that the outputs have been switched.
`PRINT#1,"ENTER 15"` `INPUT#2,E`	Retrieve the error source response.
`PRINT#1,"OUTPUT 15;C0 X C? X"`	Disconnect outputs.

`PRINT#1,"ENTER 15"`	Read the response.
`INPUT#2,A$`	
`PRINT A$`	Display response of `C00000`.
`PRINT#1,"OUTPUT 15;C13 C? X"`	Response is still `C000000`, because the Set Switch Setting command is not executed until the `X` is interpreted, while the `C?`, like all other queries, is executed immediately upon interpretation.
`PRINT#1,"OUTPUT 15;C? X"`	Response is now `C00013`.

Response Terminator Dn

Type: System Command, Immediate

Dn	Set output terminator used by RS-232C and IEEE 488.
D0	LF
D1	CR
D2	CR-LF
D3	LF-CR
D4	IEEE 488: LF with EOI; RS-232C: LF
D5	IEEE 488: CR with EOI; RS-232C: CR
D6	IEEE 488: CR-LF with EOI; RS-232C: CR-LF
D7	IEEE 488: LF-CR with EOI; RS-232C: LF-CR
D?	Response is Dn, where n is the present termination setting, a one digit number from 0 to 7.

The Response Terminator command sets the character(s) that will be appended to the end of responses from the Mux488/64. The power-on default terminator is LF with EOI (D4) for the IEEE 488 interface and CR-LF (D2) for the RS-232C interface. The choice of the correct response terminator can make responses easier to receive and interpret. For example, if CR-LF with EOI was preferred for an IEEE 488 system, the following command could be used:

`PRINT#1,"OUTPUT 15;D6 X"` Set response terminator to CR-LF with EOI.

Error Query E?

Type: System Command, Immediate

E?	Reports and clears the current contents of the error source register After execution of the Error Query command, Mux488/64 responds with one of the following error codes:
E000	No error has occurred.
E001	Invalid device dependent command (IDDC). Due to a command syntax error.
E002	Invalid device dependent command option (IDDCO). A command parameter was out of range or missing.
E004	No such unit. The Select Unit (Pn) command specified a slave unit that was not attached.
E008	Self-test failure. Reserved for non-critical internal errors.
E016	Trigger overrun. Triggers occurred faster than they could be processed.
E064	Master/slave communication error. A slave was disconnected or reset during operation, or corrupt data were transferred during master/slave communications.
E128	Conflict error. A command could not be executed because of the setting of another command.
Ennn	If two or more errors occurred, nnn is the sum of the corresponding error codes.

The Error Query command responds with the contents of the error source register and then clears that register. When an error occurs, the appropriate bits are set in the error source and standard event registers, and possibly in the status byte register. The ERROR indicator light on the front panel of the Mux488/64 illuminates.

The error query clears the error source register, the corresponding bits in the standard event register, and possibly bits in the status byte register. Clearing the error source register allows the ERROR indicator light to turn off.

Error query responds with Ennn, where nnn is a three-digit decimal number equal to the sum of the error codes.

The following examples show the use of E? and Query Attached Units (U4) commands at power-on to check for master/slave communication errors.

Example:

`OPEN "IEEEIN" FOR INPUT AS #1`	Attach and reset the IEEE 488 interface.
`IOCTL#1,"BREAK"`	
`PRINT#1,"RESET"`	
`OPEN "IEEEOUT" FOR OUTPUT AS #2`	
`PRINT#1,"TERM LF EOI"`	Use line-feed and EOI for terminator.
`PRINT#1,"OUTPUT 15;E?X"`	Query error source.
`PRINT#1,"ENTER 15"`	Retrieve response.
`INPUT#2,E`	
`PRINT E`	Displays `64` (communication error).
`PRINT#1,"OUTPUT 15;U4 X"`	Query attached units.
`PRINT#1,"ENTER 15"`	Retrieve response.
`LINE INPUT#2,A$`	
`PRINT A$`	Displays `01,nn,nn,nn...` showing which units are and are not recognized.

If more than one type of error has occurred, the response is the sum of the corresponding error codes:

`PRINT#1,"OUTPUT 15;C99 X"`	Oops, 99 is too big, causing IDDCO error.
`PRINT#1,"OUTPUT 15;P3 X"`	Oops, slave unit 3 does not exist, causing `E004` (No Such Unit) error.
`PRINT#1,"@ X"`	Oops, triggering is not enabled. Command trigger causes conflict error.
`PRINT#1,"OUTPUT 15;E? X"`	Retrieve error code.
`PRINT#1,"ENTER 15"`	
`INPUT#2,E`	
`PRINT E`	Displays `134` (128 + 4 + 2)

Set Sequence Switch Setting In

Type: Unit Command, Immediate

`In` Modifies the output settings in the sequence settings table at the location specified by the `Wn` command for the unit selected by the `Pn` command.

`I?` Responds with the present setting of `Innnnn` where `nnnnn` is a five-digit decimal number.

Set Sequence Switch Setting sets the input group the outputs of the unit selected by `Pn`) will be connected at a location in the sequence settings table determined by `Wn`. Switch setting 0 is all outputs disconnected. Settings greater than 0 connect the outputs to the specified input group. The maximum switch setting is the number of input groups, which is set by the channel grouping rear panel switch setting.

In shared operation, only one unit at a time, master or slave, may have its outputs connected to any inputs. To accomplish this, `In` sets all other units sequence settings registers at the current location to outputs disconnected (with setting 0) before setting the present unit's switch setting.

In separate operation, `In` only affects the present unit's switch setting. It does not affect the other units.

The `In` command is similar to the `Cn` command, except that it sets the output settings at a point in the sequence settings table instead of setting the outputs directly.

Example:

`PRINT#1,"OUTPUT 15;W1 X"` Specify the first element in the sequence settings and duration tables.

`PRINT#1,"OUTPUT 15;P1 I1 X"`
Connect master outputs to master input group 1.

`PRINT#1,"OUTPUT 15;P2 I4 X"`
Connect slave 2 outputs to slave 2 input group 4.

`PRINT#1,"OUTPUT 15;W$ X"` Increment the sequence table location.

`PRINT#1,"OUTPUT 15;P1 I2 P2 I3 X"`
Connect master outputs to input group 2, slave outputs to group 3.

`PRINT#1,"OUTPUT 15;W$ P1 I3 P2 I2 X"`
Connect master outputs to input group 2, slave outputs to group 3.

`PRINT#1,"OUTPUT 15;W$ P1 I4 P2 I1 X"`
Connect master outputs to input group 2, slave outputs to group 3.

The Set Sequence Outputs query (`I?`) command interrogates the output setting of the unit selected by `Pn`. The response is in the form `Innnnn`, where the setting is represented as a five-digit decimal number.

Example:

```
PRINT#1,"OUTPUT 15;W1 P1 I? P2 I? X"
```

Interrogate master and slave sequence settings table element 1.

```
PRINT#1,"ENTER 15"
INPUT#2,A$
PRINT A$
```

Response is `I00001I00004`, showing that the master outputs are connected to input group 1 and that the slave outputs are connected to input group 4.

```
PRINT#1,"OUTPUT15;W20 I3 I? X"
```

Response is `I00003`, because the Set Sequence Outputs command is executed immediately, before the `I?`.

```
PRINT#1,"OUTPUT15;W20 I? X"
```

Response is still `I00003`.

Sequence Duration Jn

Type: System Command, Immediate

`Jn` Sets up how many triggers occur for a location in the sequence settings table to `n` (1 to 65,535).

`J?` Responds with the present sequence duration in the format `Jnnnnn`.

This command sets an element in the sequence duration table. This element specifies the number of triggers received before Mux488/64 goes to the next location in the sequence settings table. `n` is the number of triggers from 1 to 65,535. The Mux continues to output from that location until `n` numbers of triggers are received.

The element to be set in the sequence duration table is determined by the Sequence Table Location (`Wn`) command.

If sequencing is active and `Jn` is changed, `Jn` temporarily suspends sequenced operation to assure synchronization of master and slave units. Trigger events that occur during this suspension, if any, are ignored.

The Select Unit (`Pn`) command has no effect on the Sequence Duration (`Jn`) command or table, because a single sequence duration table is used for the entire system.

To query the duration at a single location in the sequence duration table, use `J?`. The response is in the form `Jnnnnn`.

Example:

To make the first sequence element last for one trigger event, the next for two trigger events, the third for three, and the fourth for four, execute:

```
PRINT#1,"OUTPUT 15;W1 J1 W$ J2 W$ J3 W$ J4 X"
```

When using a one millisecond trigger source, make sequence element 250 last for three seconds:

```
PRINT#1,"OUTPUT 15;W250 J3000X"
```

Sequence Repetition Kn

Type: System Command. Deferred (Query is immediate)

`Kn` Sets the number of times the sequence range defined by the start and end parameters of `Qs,e,f` is repeated when sequence mode `L2` is selected. The range for `n` is 1 - 65,535. Default is 1.

`K?` Responds with the number of repetitions specified in the format `Knnnnn`.

The Sequence Repetition command specifies how many times the sequence range is executed before completion when sequenced operation is performed in sequence mode `L2`. The sequence range is that portion of the sequence table specified by the `s` and `e` parameters of the sequence range `Qs,e,f` command.

Sequence repetition is only used if Sequence Mode is set to counted sequenced operation (`L2`). It is ignored if any other sequence mode is used. If `Kn` is set to `K0` (no repetition), an IDDCO error occurs.

Example:

```
PRINT#1,"OUTPUT 15;K2 L2 X"
```

Sequence repetition set to 2, sequence mode set to counted.

```
PRINT#1,"OUTPUT 15;Y250 X"
```

Internal timebase set to 250 ms.

```
PRINT#1,"OUTPUT 15;Q1,10,20X"
```

Sequence range set from 1 through 10, finishing with 20.

```
PRINT#1,"OUTPUT 15; T6 X"
```

Trigger source set to internal timebase. Sequencing begins, stepping through elements 1 through 10 twice and finishing with 20.

Sequence Mode Ln

Type: System Command, Deferred (Query is immediate)

`L0`	Disables sequenced operation
`L1`	Enables once-through operation
`L2`	Enables counted sequenced operation.
`L3`	Enables continuous sequenced operation.
`L?`	Responds with `Ln` where `n` is the present sequence mode setting.

The Sequence Mode command specifies how and if the sequence range specified by `Qs,e,f` is repeated.

`L0` immediately disables any sequencing when it is executed, even if a sequence is currently running.

`L1` specifies that the sequence range defined by `Qs,e,f` is stepped through exactly once. `L1` is stopped immediately by `L0`.

`L2` specifies that the sequence range defined by `Qs,e,f` is stepped through the number of times specified by the Sequence Repetition (`Kn`) command. `L2` is stopped immediately by `L0` or after the end of the current repetition by `L1`.

`L3` specifies that the sequence range defined by `Qs,e,f` is stepped through indefinitely. `L3` is stopped immediately by `L0`, after the end of the current repetition by `L1` or at the end of `Kn` cycles by `L2`. Otherwise, `L3` operation will not finish.

Changing the Sequence Mode in any other way (such as from `L1` to `L3`) causes sequenced operation to stop immediately.

If sequenced operation is active and `Ln` is decreased (from `L3` to `L2` or `L1`; `L2` to `L1`), sequenced operation is temporarily stopped to assure synchronization of the master and slave units. Any triggers received during suspension are ignored.

Example:

```
PRINT#1,"OUTPUT 15;Q1,8,10 X"
```
Sequence range set to start at element 1, end at element 8 and final at element 10.

```
PRINT#1,"OUTPUT 15;K2 L2 X"
```
Sequence repetition set to 2, sequence mode set to counted.

```
PRINT#1,"OUTPUT 15;Y250 X"
```
Internal timebase set to 250 ms.

```
PRINT#1,"OUTPUT 15; T6 X"
```
Trigger source set to internal timebase. Sequenced operation begins.

SRQ Mask Mn

Type: System Command, Deferred (Query is immediate)

M0	Clear Service Request Enable Register (default)
M16	SRQ on Message Available
M32	SRQ on event detected
Mn	Enable SRQ on specified conditions where n is the sum of the corresponding condition codes.
M?	Response is M0Mnnn where nnn is the service request enable mask.

The SRQ Mask command sets or clears the service request enable register, which controls the generation of Master Summary Status, Request for Service, and the IEEE 488 bus SRQ signal.

The SRQ Mask command enables SRQs on one or more of the conditions listed above. Multiple SRQ Mask conditions can be enabled simultaneously by issuing them separately or by combining them in one command string. The programmed SRQ modes remain enabled until the M0 (clear SRQ mask) command is sent, or the controller sends a Reset (*R) command. This command acts directly on the Service Request Enable Register (see Status Reporting, Section 6.7).

Example:

PRINT#1,"OUTPUT 15;M16 M32 X"	Enable SRQ on Message Available or Event Detected.
PRINT#1,"OUTPUT 15;M48 X"	Same effect as previous command.
PRINT#1,"OUTPUT 15;M? X"	Read present Mn setting. Response is M048 (32 + 16)

Event Mask Nn

Type: System Command, Deferred (Query is immediate)

N0	Clear event mask (default)
N1	Sequence operation complete
N2	Trigger detected
N4	Query error
N16	Execution error
N32	Command error
N128	Power-on
Nnnn	Detect the specified conditions, where nnn is the sum of the corresponding condition codes.
N?	Read event status enable register. Response is in the form N0Nnnn, where nnn is the event enable register setting.

This command sets the event status enable register (ESE). ESE determines which conditions in the event status register (ESR) are reported in the Event Status register Bit (ESB) in the status byte register. See the section on the Status Reporting (Section 6.7) for more details.

Multiple ESR bits can be enabled simultaneously by issuing them separately or by combining them in one command string or one command. This enables reporting of multiple events. The programmed event enable masks remain set until a Clear Event Mask (N0) command is sent or the controller sends a Reset (*R) command.

Example:

`PRINT#1,"OUTPUT 15;N0 X"`	Clear ESE
`PRINT#1,"OUTPUT 15;N? X"`	Read ESE back. Response is N0N000
`PRINT#1,"OUTPUT 15;N4 N16 X"`	Enable detection of query error and execution error.
`PRINT#1,"OUTPUT 15;N20 X"`	Also enable detection of query error and execution error.
`PRINT#1,"OUTPUT 15;N? X"`	Query event mask setting. Response is N0N020, the logical OR of these two conditions.

Select Unit Pn

Type: System Command, Immediate

Pn	Select the unit (master or specified slave) used in subsequent operations.
P?	Responds with Pnn, where nn is a two-digit decimal number specifying the presently selected unit.

Select Unit chooses the unit that subsequent commands affect. The n specifies the unit, where 1 is the master and 2 through 16 are the slaves. The default unit number is 1, referring to the master unit, so when no unit has been selected, commands affect the master unit. If the system consists only of a master unit, the unit number does not need to be specified.

The slave unit numbers are the slave identification numbers set by the rear panel **SLAVE ID** switches (see Section 2 for setting instructions).

Select Unit (Pn) immediately affects all subsequent commands, including subsequent deferred commands such as Cn. This allows several units to be referred to in a single command string (as in P1 C4 P2 C5 X). If the specified slave is not attached, then a No Such Unit error occurs.

Example:

PRINT#1,"OUTPUT 15;P1 X"	Specify master unit
PRINT#1,"OUTPUT 15;C3 X"	Connect to input group 3
PRINT#1,"OUTPUT 15;P5 C1 X"	Connect slave 5 to input group 1
PRINT#1,"OUTPUT 15;P1 C4 P2 C5 X"	Connect master outputs to input group 4 and slave to input group 5

Sequence Range Qs,e,f

Type: System Command, Deferred (Query is immediate)

`Qs,e,f` Sets the sequence range. `s` is the starting location of the sequence. `e` is the end location. `f` is the location the unit is left at after sequenced operation is complete.

`Q?` Responds with the sequence range in the format `Qnnnnn,nnnnn,nnnnn`.

Sequence Range chooses a range from the sequence settings and duration tables that is used for sequenced operation and some of the status commands.

The sequence range consists of a range of settings and a final setting. The range (defined by `s` and `e`) is the portion of the sequence table to be repeated during sequenced operation. The final setting (defined by `f`) is the sequence table element that takes effect after sequencing is complete. `s` is the starting sequence settings table location. `e` is the end location and must be greater than or equal to `s`. `f` is the location the unit is left at when sequenced operation is complete.

s,e, and `f` are all decimal integers from 1 to 1100.

If the final element is the same as the end of the range (`f` is the same as `e`), sequenced operation stops at the end of the range and does not proceed to the final element and repeat the ending setting. For example, `Q1,3,5` produces the sequence 1,2,3,5. But `Q1,3,3` does not produce the sequence 1,2,3,3. It stops at the ending element and produces 1,2,3.

If a sequence range has not been specified with the `Qs,e,f` command, a default range is used for sequenced operation. The default range is based on the sequence table location, which is set or incremented with the `Wn` command. Mux keeps track of the greatest sequence table location that has been referenced and uses that location as the sequence range end and final elements. For example, at power-on or Reset (`*R`), the default range is `Q1,1,1`. If `W50 X` is the largest sequence location that has been specified (elements up through element 50 have been referenced), the default range would be `Q1,50,50`.

`Qs,e,f` is a deferred command and is not acted on until an Execute (`X`) is received.

The start and end locations of the sequence range are also used by the `U7` and `U8` commands to specify the portion of the sequence tables that they will query.

```
PRINT#1,"OUTPUT 15;Q1,8,10 X"
```

Set the sequence range to begin at location 1 and end at location 8, with location 10 as the final setting.

Trigger Source Tn

Type: System Command, Deferred (Query is immediate)

`T0`	Disable triggering (default)
`T1`	Trigger on IEEE 488 bus (GET) trigger
`T2`	Trigger on BNC input TTL rising edge
`T3`	Trigger on BNC input TTL falling edge
`T4`	Trigger on BNC input TTL rising edge and on falling edge
`T5`	Trigger on enabled command (@) trigger
`T6`	Trigger every timebase interval starting after the execute (`X`) command.
`T7`	Trigger every timebase interval starting after an IEEE 488 bus (GET) trigger.
`T8`	Trigger every timebase interval starting after a BNC input TTL rising edge.
`T9`	Trigger every timebase interval starting after a BNC input TTL falling edge.
`T10`	Trigger every timebase interval starting after a BNC input TTL rising or falling edge.
`T11`	Trigger every timebase interval starting after a command trigger.
`T?`	Responds with the present trigger setting in the format `Tnn` where `nn` is a two-digit decimal number.

The Trigger Source command chooses which trigger source is used for sequenced operation. The power-up default is `T0`, trigger disabled.

Modifying the trigger source is a deferred operation.

When Trigger Source is set to GET (`T1` or `T7`), a Group Execute Trigger on the IEEE 488 bus constitutes a trigger. When trigger source is set to timebase interval, the Timebase Interval (`Yn`) command controls the time between triggers.

When the trigger source is the @ command (`T5` or `T11`), a trigger is enabled by the @ command, which causes a trigger to occur when the next execute command (`X`) following the @ is encountered in the command line. A trigger is not generated until the entire command string is processed.

`T2`, `T4`, `T8`, `T9`, and `T10` set the trigger source to the rising and/or falling edges of the BNC trigger input. The trigger input signal must be a TTL compatible signal with a minimum width of 200 ns, high or low.

Regardless of the trigger source, the Mux488/64 cannot process more than 4000 triggers per second. At least 250 microseconds must separate consecutive triggers or a trigger overrun error will occur and sequenced operation will immediately stop.

`T6` triggers every timebase interval starting after the execute (`X`) command.

The `T7` through `T11` modes are used to start timebase triggering after a specified trigger event. In these modes, the unit waits for the specified trigger. When it detects it, the unit initializes the timebase and starts sequenced operation. Note that the initial trigger event is not part of sequenced operation; it does not affect the switch settings nor does it generate a trigger delay output pulse.

The timebase interval triggering does not start immediately after the specified trigger events. Instead, there is a variable delay of 100 to 200 microseconds before the timebase is active. After a further delay specified by the Timebase Interval (Yn), a timebase trigger occurs, affecting the switch setting and causing a delayed trigger output pulse. The switch settings are not affected and no delay output pulse is generated by the initial trigger source.

The timebase interval for the T6 through T11 commands is specified by the Timebase Interval (Yn) command.

When triggering is completed, Tn returns to its power-up default setting, T0.

Note that the Jn and Ln commands may temporarily suspend triggering while they execute to ensure synchronization of the master and slaves. Triggers are ignored during this suspension.

Example:

`PRINT#1,"OUTPUT 15;Q1,1100,1100 K10 L2 T6 X"`	
	Execute the entire sequence table 10 times at timebase interval.
`PRINT#1,"OUTPUT 15;T? X"`	Query trigger source.
`PRINT#1,"ENTER 15"`	Retrieve response.
`INPUT#2,A$`	
`PRINT A$`	Response is T00 if sequenced operation is complete.

Status Un

Type: System Commands (U0-U4, U7, U9), Unit Commands (U5, U6, U8, U10), Immediate

U0	Query and clear Event Status Register (ESR)
U1	Query the status byte register (STB)
U2	Query system settings.
U3	Query sequence duration table.
U4	Query connected units
U5	Query unit settings.
U6	Query sequence settings table.
U7	Query range of sequence duration table.
U8	Query range of sequence settings table.
U9	Query product name and revision.
U10	Query output configuration.
U?	Responds with the last Un command executed.

U0 reads and clears the Event Status Register (ESR) (see Section 6.7 for details on status reporting). The event status register is a read-only register whose bits correspond to those of the event enable register and indicates which events have occurred since the event status register has last been read. It is reset immediately after being read.

It responds with:

001	Sequenced Operation Complete.
002	Trigger Detected
004	Query error
016	Execution error
032	Command error
128	Power-on
nnn	A three-digit number equalling the sum of some or all of the above responses.

U1 responds with the Status Byte Register. This is a copy of the same byte returned in response to a serial poll from the IEEE 488 bus except that bit 6 carries the Master Summary Status rather than the Request for Service. The status byte register is a read-only register whose bits correspond to those of the service request enable register with the addition of bit value 64, which responds with the Master Summary Status (MSS). The MSS indicates whether or not this device needs service.

U1 responds with:

016	Message available
032	Event detected.
064	Master Summary Status.

nnn A three-digit number equalling the sum of some or all of the above responses.

`U2` responds with the system settings. This is all the information necessary to reconfigure the Mux system commands to the same state as the existing state when this command was executed, except for the sequence tables and the NVRAM. Its response is equivalent to the response to the following commands:

`D?F?L?M?N?K?Q?T?Y?Z?.`

`U3` responds with the contents of the sequence duration table in the form:

`W00001Jnnnnn XW$Jnnnnn X . . . W$Jnnnnn X`

This response is the entire sequence duration table, in one long executable string of 9904 characters (plus one or two terminators). The string consists of the commands that would be necessary to reload the entire sequence duration table.

`U4` responds with a list of the connected units in the form: `01,02,03,04,00,00,00,00,09,10,11,12,00,00,00,16` showing which units are connected and which are not. If a unit is not attached, it is shown as `00`. `01`, the master unit, is always attached.

`U5` responds with the unit-specific settings of the currently selected unit. The response is equivalent to the response of the following commands: `P?C?`.

The `U6` response is the contents of the sequence settings table of the currently specified unit, in the form:

`Pn X W00000Innnnn XW$Innnnn X . . . W$Innnnn X`

This response is the entire sequence settings table of the specified unit, in one long executable string of 9904 characters (plus one or two terminators).

`U7` responds with the duration settings for a section of the sequence duration table. This response is the same as `U3`, except the start and end of the response are defined by the Sequence Range (`Qs,e,f`) command. For example:

```
PRINT#1,"OUTPUT 15;Q10,12,50 X U7 X"
```

responds with `W00010JnnnnnXW$JnnnnnXW$JnnnnnX`, showing the durations for locations 10, 11, and 12.

The `U8` responds with a section of the sequence settings table of a unit in the system. This response is the same as `U6`, except the start and end of the response are defined by the `Qs,e,f` command. For example:

```
PRINT#1,"OUTPUT 15;Q10,12,50 X U8 X"
```

responds with P1F0XW00010InnnnnXW$InnnnnXW$InnnnnX, showing the durations for locations 10, 11, and 12.

The U9 response is an ASCII string identifying the product and the revision and version of the firmware installed in the unit specified by Pn. The response is "IOtech,Mux488/64,0,v.r" or "IOtech,Mux/64,0,v.r" where v is the version and r is the revision of the firmware.

U10 responds with the rear panel switch settings of output configuration and channel grouping of the unit specified by Pn in a three-character string of the form c,g, where c is the output configuration (0 = shared, 1 = separate), and g is the channel grouping (0 = 1 output, 1 = 2 outputs, 2 = 4 outputs, 3 = 8 outputs, and 4 = 16 outputs). During shared operation, the master unit's channel grouping overrides the slave units' settings and so the slave units' responses may not match the physical switch settings.

U? responds with the last Un command executed. The response Unn where nn is the number of the most recent query command from 00 to 09.

Sequence Table Location Wn

Type: System Command, Immediate

`Wn`	Sets the sequence table to location n. The range for n is 1 to 1100.
`W$`	Increments the sequence table location.
`W?`	Responds with the present sequence settings table location in the format `Wnnnnn`.

The Sequence Table Location specifies which of the 1100 elements within the sequence settings table is to be referenced by the `In`, `I?`, `Jn` and `J?` commands.

`W$` provides a simple method for incrementing the sequence table location. If the location is incremented past the maximum value of 1100, an IDDCO error occurs. If this occurs, reset the location to a lower value with the `Wn` command.

The Sequence Table Location query (`W?`) command responds with `Wnnnnn`, where `nnnnn` is the present sequence table location.

Example:

`PRINT#1,"OUTPUT 15;W10 X"`	Refer to sequence table location 10.
`PRINT#1,"OUTPUT 15;W$ X"`	Increment location to 11.
`PRINT#1,"OUTPUT 15;W? X"`	Response is `W00011`.

Execute X

Type: System Command

X Execute preceding command string.

The Execute command executes all deferred commands in a command string in the order described below, takes care of enabled actions such as command trigger, and adds output terminators to any query responses. Deferred commands are interpreted and processed when they are received, but are not executed until an X is received.

Deferred commands are executed in the following order: Mn, Nn, Cn, Kn, Qs,e,f, Yn, Zn, Ln, Tn, @, *R.

Immediate commands do not require an Execute command to be processed. For more detail, see the full description for each command.

If multiple deferred commands that refer to the same setting are used in the same string, each use of the command must be followed by an X. Any number of Execute commands can be inserted into the same command string. If a deferred command is repeated without the X between them, only the second command takes effect. For example, C1 C2 X is the equivalent of C2 X. However, P1 C1 P2 C2 X both take effect, because they are setting switch settings for different units.

If errors occur while processing the command string, X has no effect. For example:

If C1 X C0 GGGGGGG X C? X is sent, C1 is the only command that takes effect.

Examples:

PRINT#1,"OUTPUT 15;C1 C0 X"	Disconnect outputs.
PRINT#1,"OUTPUT 15;C1 X C0 X"	Connect outputs to input group 1 and then disconnect them.

Timebase Interval Yn

Type: System Command, Deferred (Query is immediate)

`Yn`	Set interval time to `n` (0.250 to 60000.000 milliseconds).
`Y?`	Responds with the present timebase interval setting in milliseconds in the form `Ynnnnn.nnn`.

The timebase interval specifies the time between the start of consecutive pulses generated by the timebase generator. The interval is specified in milliseconds ranging from 0.250 milliseconds (250 microseconds) to 60000 milliseconds (60 seconds). The timebase interval is used as a trigger source when Trigger Source is set to `T6` through `T11`.

Example:

`PRINT#1,"OUTPUT#11;Y1.25 X"`	Set timebase for 1.25 ms (800 Hz).
`PRINT#1,"OUTPUT 15;Y250 X"`	Sets internal timebase to 250 milliseconds (¼ second).

Trigger Delay Zn

Type: System Command, Deferred (Query is immediate)

`Zn`	Set the trigger delay to a specified number of milliseconds from 0.010 to 60000.000.
`Z?`	Responds with the present trigger delay setting.

The delayed trigger output is used with applications that require some delay or setup time after the Mux's outputs have been commanded to switch before meaningful action can be taken. The time delay is specified in milliseconds from 0.010 to 60000.000 (10 microseconds to 60 seconds).

When a trigger is detected, the normally-high delayed trigger output immediately goes low. It remains low for the length of time specified by the trigger delay and then returns to a high level. Thus the trigger delay output provides a rising edge signal that is delayed from the trigger event by the trigger delay.

For meaningful operation, the trigger delay must be less than the trigger interval. If the internal timebase interval is used for triggering, the Mux488/64 adjusts the trigger delay, if necessary, until it is at least 10 microseconds less than the trigger interval. When other trigger sources are used, the programmer must set the trigger delay to an appropriately small value.

The response of `Z?` is `Znnnnn.nnn`, where `n` is the trigger delay in milliseconds. This might be different from the commanded value if the timebase interval (set by `Yn`) is the trigger source and has caused the trigger delay to be adjusted.

Modifying the trigger out register is a deferred operation.

Example:

`PRINT#1,"OUTPUT 15;Y10 Z5 X"`	Set trigger interval to 10 msec and the trigger delay to 5 msec.

IEEE 488 Primer

7.1 History

The IEEE 488 bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978.

Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the IEEE 488 interface (sometimes called the General Purpose Interface Bus or GPIB) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the IEEE 488 standard has moved the responsibility of the user from design of the interface to design of the high level software that is specific to the measurement application.

7.2 General Structure

The main purpose of the GPIB is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will talk and who will listen usually falls on the System Controller which is, at power on, the Active Controller.

The System Controller is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the Active Talker. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called Addressing to Talk. If the committee chairman can not attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the GPIB, this device becomes the Active Controller.

At a committee meeting, everyone present usually listens. This is not the case with the GPIB. The Active Controller selects which devices will listen and commands all other devices to ignore what is being transmitted. A device is instructed to listen by being Addressed to Listen. This device is then referred to as an Active Listener. Devices which are to ignore the data message are instructed to Unlisten.

The reason some devices are instructed to Unlisten is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (Unlisten) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The GPIB transfers information in a similar way. This method of data transfer is called handshaking. More on this later.

For data transfer on the IEEE 488, the Active Controller must:

a. Unlisten all devices to protect against eavesdroppers.
b. Designate who will talk by addressing a device to talk.
c. Designate all the devices who are to listen by addressing those devices to listen.
d. Indicate to all devices that the data transfer can take place.

7.3 Send It To My Address

In the previous discussion, the terms Addressed to Talk and Addressed to Listen were used. These terms require some clarification.

The IEEE 488 standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Addresses are sent with universal (multiline) commands from the Active Controller. These commands include My Listen Address (MLA), My Talk Address (MTA), Talk Address Group (TAG), and Listen Address Group (LAG).

7.4 Bus Management Lines

Five hardware lines on the GPIB are used for bus management. Signals on these lines are often referred to as uniline (single line) commands. The signals are active low, i.e. a low voltage represents a logic "1" (asserted), and a high voltage represents a logic "0" (unasserted).

7.4.1 Attention (ATN)

ATN is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the Active Listeners. The Active Controller is the only bus device that has control of this line.

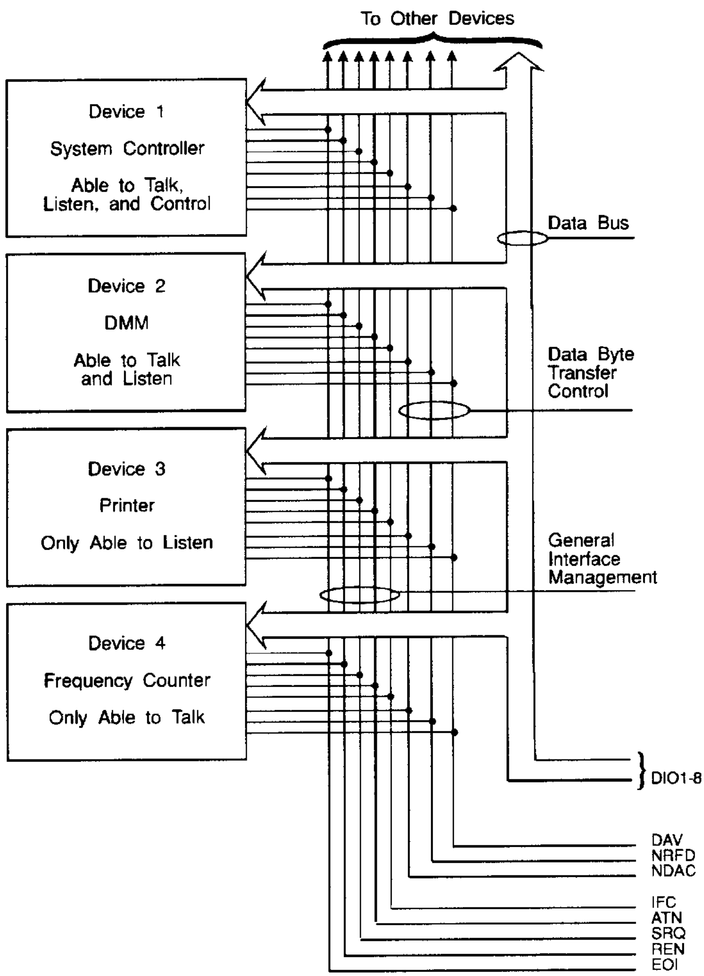

Figure 7.1: IEEE 488 Bus Structure

7.4.2 Interface Clear (IFC)

The IFC line is used only by the System Controller. It is used to place all bus devices in a known state. Although device configurations vary, the IFC command usually places the devices in the Talk and Listen Idle states (neither Active Talker nor Active Listener).

7.4.3 Remote Enable (REN)

When the System Controller sends the REN command, bus devices will respond to remote operation. Generally, the REN command should be issued before any bus programming is attempted. Only the System Controller has control of the Remote Enable line.

7.4.4 End or Identify (EOI)

The EOI line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts EOI during the transfer of the last data byte. The EOI signal is not always necessary as the end of the data may be indicated by some special character such as carriage return.

The Active Controller also uses EOI to perform a Parallel Poll by simultaneously asserting EOI and ATN.

7.4.5 Service Request (SRQ)

When a device desires the immediate attention of the Active Controller it asserts SRQ. It is then the Controller's responsibility to determine which device requested service. This is accomplished with a Serial Poll or a Parallel Poll.

7.5 Handshake Lines

The GPIB uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other IEEE 488 lines, are active low.

7.5.1 Data Valid (DAV)

The DAV line is controlled by the Talker. The Talker verifies that NDAC is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The Talker then outputs data on the bus and waits until NRFD is unasserted (high) which indicates that all Addressed Listeners are ready to accept the information. When NRFD and NDAC are in the proper state, the Talker asserts DAV (active low) to indicate that the data on the bus is valid.

7.5.2 Not Ready for Data (NRFD)

This line is used by the Listeners to inform the Talker when they are ready to accept new data. The Talker must wait for each Listener to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.

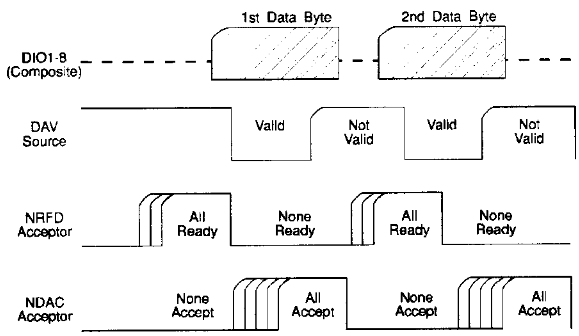

Figure 7.2: IEEE 488 Bus Handshaking

7.5.3 Not Data Accepted (NDAC)

The NDAC line is also controlled by the Listeners. This line indicates to the Talker that each device addressed to listen has accepted the information. Each device releases NDAC (high) at its own rate, but the NDAC will not go high until the slowest Listener has accepted the data byte.

7.6 Data Lines

The GPIB provides eight data lines for a bit parallel/byte serial data transfer. These eight data lines use the convention of DIO1 through DIO8 instead of the binary designation of D0 to D7. The data lines are bidirectional and are active low.

7.7 Multiline Commands

Multiline (bus) commands are sent by the Active Controller over the data bus with ATN asserted. These commands include addressing commands for talk, listen, Untalk and Unlisten.

7.7.1 Go To Local (GTL)

This command allows the selected devices to be manually controlled. ($01)

7.7.2 Listen Address Group (LAG)

There are 31 (0 to 30) listen addresses associated with this group. The 3 most significant bits of the data bus are set to 001 while the 5 least significant bits are the address of the device being told to listen.

7.7.3 Unlisten (UNL)

This command tells all bus devices to Unlisten. The same as Unaddressed to Listen. ($3F)

7.7.4 Talk Address Group (TAG)

There are 31 (0 to 30) talk addresses associated with this group. The 3 most significant bits of the data bus are set to 010 while the 5 least significant bits are the address of the device being told to talk.

7.7.5 Untalk (UNT)

This command tells bus devices to Untalk. The same as Unaddressed to Talk. ($5F)

7.7.6 Local Lockout (LLO)

Issuing the LLO command prevents manual control of the instrument's functions. ($11)

7.7.7 Device Clear (DCL)

This command causes all bus devices to be initialized to a pre-defined or power up state. ($14)

7.7.8 Selected Device Clear (SDC)

This causes a single device to be initialized to a pre-defined or power up state. ($04)

7.7.9 Serial Poll Disable (SPD)

The SPD command disables all devices from sending their Serial Poll status byte. ($19)

7.7.10 Serial Poll Enable (SPE)

A device which is Addressed to Talk will output its Serial Poll status byte after SPE is sent and ATN is unasserted. ($18)

7.7.11 Group Execute Trigger (GET)

This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously. ($08)

7.7.12 Take Control (TCT)

This command passes bus control responsibilities from the current Controller to another device which has the ability to control. ($09)

7.7.13 Secondary Command Group (SCG)

These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a talk or listen address. ($60 to $7F)

7.7.14 Parallel Poll Configure (PPC)

This configures devices capable of performing a Parallel Poll as to which data bit they are to assert in response to a Parallel Poll. ($05)

7.7.15 Parallel Poll Unconfigure (PPU)

This disables all devices from responding to a Parallel Poll. ($15)

7.8 More on Service Requests

Most of the commands covered, both uniline and multiline, are the responsibility of the Active Controller to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, SRQ is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the Controller that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert SRQ. A DMM might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might SRQ if its output has current limited. This is a powerful bus feature that removes the burden from the System Controller to periodically inquire, "Are you done yet?". Instead, the Controller says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since SRQ is a single line command, there is no way for the Controller to determine which device requested the service without additional information. This information is provided by the multiline commands for Serial Poll and Parallel Poll.

7.8.1 Serial Poll

Suppose the Controller receives a service request. For this example, let's assume there are several devices which could assert SRQ. The Controller issues an SPE (Serial Poll enable) command to each device sequentially. If any device responds with DIO7 asserted it indicates to the Controller that it was the device that asserted SRQ. Often times the other bits will indicate why the device wanted service. This Serial Polling sequence, and any resulting action, is under control of the software designer.

7.8.2 Parallel Poll

The Parallel Poll is another way the Controller can determine which device requested service. It provides the who but not necessarily the why. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical OR/AND conditions on one data line by more than one device. When SRQ is asserted, the Controller (under user's software) conducts a Parallel Poll. The Controller must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a Serial Poll might be used to determine the why.

Of the two polling types, the Serial Poll is the most popular due to its ability to determine the who and why. In addition, most devices support Serial Poll only.

Appendix A: Mux Command Summary

Command	Code	Type	Description	Page #
Command Trigger	@	System	Enables generation of a command trigger.	6.14
Reset	*R	System	Restores the master unit and any attached slave units to their initial power-up state.	6.15
Set Switch Setting	Cn	Unit	Connect the outputs of the unit selected by Pn to the input group specified by n.	6.16
	C?	Unit	Responds with Cnnnnn where nnnnn is the present switch setting of the unit specified by the Pn command.	
Output Terminator	Dn	System	Set output terminator used by RS-232C and IEEE 488.	6.18
	D0		LF	
	D1		CR	
	D2		CR-LF	
	D3		LF-CR	
	D4		IEEE 488: LF plus EOI; RS-232C: LF	
	D5		IEEE 488: CR plus EOI; RS-232C: CR	
	D6		IEEE 488: CR-LF plus EOI; RS-232C: CR-LF	
	D7		IEEE 488: LF-CR plus EOI; RS-232C: LF-CR	
	D?		Response is Dn, where n is the present termination setting	

Command	Code	Type	Description	Page #
Error Query	`E?`	System	Responds with and clears present error condition. Responds with one of the following error codes:	6.19
	`E000`		No error has occurred.	
	`E001`		Invalid device dependent command (IDDC).	
	`E002`		Invalid device dependent command option (IDDCO).	
	`E004`		No such unit.	
	`E008`		Self-test failure.	
	`E016`		Trigger overrun.	
	`E064`		Master/slave communication error.	
	`E128`		Conflict error.	
	`Ennn`		If two or more errors occurred, nnn is the sum of the corresponding error codes.	
Set Sequence Switch Setting	`In`	Unit	Modifies the output settings in the sequence settings table at the location specified by the `Wn` command for the selected unit.	6.21
	`I?`		Responds with the present setting of `In`.	
Sequence Duration	`Jn`	System	Sets up how many triggers occur for a location in the sequence settings table to `n` (1 to 65,535).	6.23
	`J?`		Responds with the present sequence duration.	
Sequence Repetition	`Kn`	System	Determines how many times the sequence defined by `Qs,e,f` is repeated when `L2` is selected.	6.24
	`K?`		Responds with the sequence repetitions specified.	
Sequence Mode	`Ln`	System	Specifies how the sequence table is repeated.	6.25
	`L0`		Disables sequenced operation.	
	`L1`		Enables once-through operation.	
	`L2`		Enables counted sequenced operation.	
	`L3`		Enables continuous sequenced operation.	
	`L?`		Responds with the present sequence mode setting.	

Command	Code	Type	Description	Page #
SRQ Mask	Mn	System	Specifies which conditions generate a service request.	6.26
	M0		Clear SRE (default)	
	M16		SRQ on Message Available	
	M32		SRQ on event detected	
	Mnnn		Enable SRQ on specified conditions where n is the sum of the corresponding condition codes.	
	M?		Responds with M0Mnnn where nnn is the service request enable mask.	
Event Mask	Nn	System	Determines the conditions in the Event Status Register (ESR) that are reported in the Event Status register Bit (ESB) in the Serial Poll Status Register.	6.27
	N0		Clear event mask (default).	
	N1		Sequence operation complete.	
	N2		Trigger detected.	
	N4		Query error.	
	N16		Execution error.	
	N32		Command error.	
	N128		Power-on.	
	Nnnn		Detect the specified conditions where nnn is the sum of the corresponding condition codes.	
	N?		Responds with the event mask in the form N0Nn.	
Select Unit	Pn	System	Select the unit (master or specified slave) used in subsequent operations.	
	P?		Responds with the presently selected unit.	
Sequence Range	Qs,e,f	System	Sets the sequence range. s is the starting location of the sequence. e is the end location. f is the location the unit is left at after sequenced operation is complete.	6.29
	Q?		Responds with the sequence range.	

Command	Code	Type	Description	Page #
Trigger Source	Tn	System	Chooses trigger source for sequenced operation.	6.30
	T0		Disable triggering (default).	
	T1		Trigger on IEEE 488 bus (GET) trigger	
	T2		Trigger on BNC input TTL rising edge	
	T3		Trigger on BNC input TTL falling edge	
	T4		Trigger on BNC input TTL either edge	
	T5		Trigger on enabled command (@) trigger	
	T6		Trigger every timebase interval after the X.	
	T7		Trigger every timebase interval after a GET.	
	T8		Trigger every timebase interval after a BNC input TTL rising edge.	
	T9		Trigger every timebase interval after a BNC input TTL falling edge.	
	T10		Trigger every timebase interval after a BNC input TTL rising or falling edge.	
	T11		Trigger every timebase interval after an @.	
	T?		Responds with the present trigger setting.	
Status	U0	System	Query and clear Event Status Register (ESR)	6.32
	U1	System	Query the status byte register (STB).	
	U2	System	Query system settings.	
	U3	System	Query sequence duration table.	
	U4	System	Query connected units.	
	U5	Unit	Query unit settings.	
	U6	Unit	Query sequence setttings table.	
	U7	System	Query a range of the sequence duration table.	
	U8	Unit	Query a range of the sequence settings table.	
	U9	Unit	Query product name and revision.	
	U10	Unit	Query output configuration.	
	U?	System	Responds with the last Un command executed.	

Command	Code	Type	Description	Page #
Sequence Table Location	Wn	System	Sets the sequence table to location n. The range for n is 1 to 1100.	6.35
	W$		Increments the sequence table location.	
	W?		Responds with the present sequence settings table location.	
Execute	X	System	Execute preceding command string.	6.36
Timebase Interval	Yn	System	Set interval time to n (0.250 to 60000.000 milliseconds).	6.37
	Y?		Responds with the present timebase interval setting in milliseconds in the form Ynnnnn.nnn.	
Trigger Delay	Zn	System	Set the trigger delay to the specified number of milliseconds from 0.010 to 60000.000.	6.38
	Z?		Responds with the present trigger delay setting.	

Appendix B: Additional Mux Programming Examples

```
10  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
20  ' Mux488/64 Programming Example 1
30  '
40  ' This example switches among the different input signals under program
50  ' control.  It assumes that the Mux488/16 is configured for 8 groups
60  ' of 8 single-ended inputs and 8 single-ended outputs.
70  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
80
90  'Initialize the communication with IOtech's Driver488
100 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
110 IOCTL #1, "BREAK"
120 PRINT #1, "RESET"
130 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
140 PRINT #1, "TERM LF EOI"
150
160
170 'Switch between all input groups
180 PRINT #1, "OUTPUT 15; C1X"      'Switch to input group 1
190 PRINT #1, "OUTPUT 15; C?X"      'Query the switch connection
200 PRINT #1, "ENTER 15"
210 INPUT #2, A$
220 PRINT A$                        'The response is C00001
230
240 PRINT #1, "OUTPUT 15; C2X"      'Switch to input group 2
250 PRINT #1, "OUTPUT 15; C3X"      'Switch to input group 3
260 PRINT #1, "OUTPUT 15; C4X"      'Switch to input group 4
270 PRINT #1, "OUTPUT 15; C5X"      'Switch to input group 5
280 PRINT #1, "OUTPUT 15; C6X"      'Switch to input group 6
290 PRINT #1, "OUTPUT 15; C7X"      'Switch to input group 7
300 PRINT #1, "OUTPUT 15; C8X"      'Switch to input group 8
310 PRINT #1, "OUTPUT 15; C0X"      'Turn off the outputs
320
330 END
```

```
10  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
20  ' Mux488/64 Programming Example 2
30  '
40  ' This example programs the Mux488/64 for sequenced operation.  Again the
50  ' Mux is configured for 8 groups of 8 single-ended inputs and 8 single-
60  ' ended outputs. The sequence is programmed for the following operations:
70  '     * Sequence from input group 1 to input group 8
80  '     * Each input group will be connected for 1 trigger duration
90  '     * Repeat the sequence twice
100 '     * When completed, turn the outputs off
110 '     * Trigger on a command trigger
120 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
130
140 'Initialize the communication with IOtech's Driver488
150 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
160 IOCTL #1, "BREAK"
170 PRINT #1, "RESET"
180 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
190 PRINT #1, "TERM LF EOI"
200
210
220 PRINT #1, "OUTPUT 15;T0X"        'Disable any previous triggering mode
230
240 'Load the sequence table
250 PRINT #1, "OUTPUT 15;W1 I1 J1 X" 'Set sequence address 1 for outputs
260                                 'connected to input group 1 for a duration
270                                 'of 1 trigger event
280
290 PRINT #1, "OUTPUT 15;W$ I2 J1 X"'Increment the sequence addresses and load
300 PRINT #1, "OUTPUT 15;W$ I3 J1 X"'with outputs connected to input groups 2
310 PRINT #1, "OUTPUT 15;W$ I4 J1 X" 'through 8 for a duration of 1 trigger
320 PRINT #1, "OUTPUT 15;W$ I5 J1 X" 'event
330 PRINT #1, "OUTPUT 15;W$ I6 J1 X"
340 PRINT #1, "OUTPUT 15;W$ I7 J1 X"
350 PRINT #1, "OUTPUT 15;W$ I8 J1 X"
360
370 PRINT #1, "OUTPUT 15;W$ I0 J1 X" 'Sequence address 9 will turn off the
380                                 'outputs
390
400
410 PRINT #1, "OUTPUT 15;Q1,8,9 X"   'Program the sequence to start at address 1,
420                                  'continue through address 8, and end at
430                                  'address 9
440
450 PRINT #1, "OUTPUT 15;L2 K2 X"    'Program the sequence to execute two times
460 PRINT #1, "OUTPUT 15;T5 X"      'Set the trigger
470
480 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 1
```

```
490 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 2
500 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 3
510 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 4
520 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 5
530 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 6
540 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 7
550 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 8
560 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 1
570 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 2
580 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 3
590 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 4
600 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 5
610 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 6
620 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 7
630 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs set to group 8
640 PRINT #1, "OUTPUT 15;@X"        'send command trigger, outputs turned off
650
660
670 END
```

```
10  ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
20  ' Mux488/64 Programming Example 3
30  '
40  ' This example programs the Mux488/64 and one Mux/64 slave for sequenced
50  ' operation.  Both Muxes are configured for 8 groups of 8 single-ended
60  ' inputs and 8 single- ended outputs.  The slave is set at slave id 2, and
70  ' the units are in the separate output mode. The sequence is programmed
80  ' for the following operations:
90  '     * Sequence from input group 1 to input group 8 in both units
100 '     * Each input group will be connected for 2 trigger durations
110 '     * Loop through the sequence continuously
120 '     * Trigger on the internal timebase
130 '     * Use an internal timebase of 1.5 milliseconds
140 '     * Set the output trigger delay to .3 milliseconds
150 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
160
170 'Initialize the communication with IOtech's Driver488
180 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
190 IOCTL #1, "BREAK"
200 PRINT #1, "RESET"
210 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
220
230
240 PRINT #1, "OUTPUT 15;T0X"       'Disable any previous triggering mode
250
260 'Load the sequence table of the Master
270 PRINT #1, "OUTPUT 15;P1 W1 I1 J2 X"'Set sequence address 1 for outputs
280                                     'connected to input group 1 for a
290                                     'duration of 2 trigger events
300
310 PRINT #1, "OUTPUT 15;W$ I2 J2 X"'Increment the sequence addresses and load
320 PRINT #1, "OUTPUT 15;W$ I3 J2 X"'with outputs connected to input groups 2
330 PRINT #1, "OUTPUT 15;W$ I4 J2 X"'through 8 for a duration of 2 trigger
340 PRINT #1, "OUTPUT 15;W$ I5 J2 X"'events
350 PRINT #1, "OUTPUT 15;W$ I6 J2 X"
360 PRINT #1, "OUTPUT 15;W$ I7 J2 X"
370 PRINT #1, "OUTPUT 15;W$ I8 J2 X"
380
390 'Load the sequence settings of the Slave.  Note that we do not need to
400 'program the duration of the slave unit.  This is controlled by the Master.
410 PRINT #1, "OUTPUT 15;P2 W1 I1 X"'Set sequence address 1 for outputs
420                                 'connected to input group 1
430
440 PRINT #1, "OUTPUT 15;W$ I2 X"   'Increment the sequence addresses and load
450 PRINT #1, "OUTPUT 15;W$ I3 X"   'with outputs connected to input groups 2
460 PRINT #1, "OUTPUT 15;W$ I4 X"   'through 8
470 PRINT #1, "OUTPUT 15;W$ I5 X"
480 PRINT #1, "OUTPUT 15;W$ I6 X"
```

```
490 PRINT #1, "OUTPUT 15;W$ I7 X"
500 PRINT #1, "OUTPUT 15;W$ I8 X"
510
520 PRINT #1, "OUTPUT 15;Q1,8,8 X"  'Program the sequence to start at address 1,
530                                 'continue through address 8
540
550 PRINT #1, "OUTPUT 15;L3 X"      'Program the sequence to execute continuously
560 PRINT #1, "OUTPUT 15;Y1.5X"     'Set the timebase to 1.5 milliseconds
570 PRINT #1, "OUTPUT 15;Z0.3X"     'Set the trigger output delay to .3 mseconds
580 PRINT #1, "OUTPUT 15;T6 X"      'Set the trigger mode to the internal timebase
590
600                                 'The sequence will now be executing
610
620 END
```

Appendix C: Cable Pinouts

C.1 CA-81: Master to Slave Interface Cable

Master Connector 15-pin Male D-Shell		Signal Direction	Slave Connector 15-pin Male D-Shell	
Master	Pin		Pin	Slave
/STB	1		1	/STB
GND	14		14	GND
D0	2		2	D0
D1	3		3	D1
D2	4		4	D2
D3	5		5	D3
D4	6		6	D4
D5	7		7	D5
D6	8		8	D6
GND	14		14	GND
/ACK	9	NC	9	/ACK
READY	10		10	READY
GND	15		15	GND
/INIT	11		11	/INIT
GND	14		14	GND
SELECT	12	NC	12	SELECT
/TRIGIO	13		13	/TRIGIO
GND	15		15	GND

C.2 CA-82: Slave to Slave Interface Cable

Slave Connector 15-pin Male D-Shell		Signal Direction	Slave Connector 15-pin Male D-Shell		Signal Direction	Daisy-Chain Connector 15-pin Female D-Shell	
Master	Pin		Slave	Pin			
/STB	1	←	1	/STB	←	1	/STB
GND	14		14	GND		14	GND
D0	2	↔	2	D0	↔	2	D0
D1	3		3	D1		3	D1
D2	4	↔	4	D2	↔	4	D2
D3	5		5	D3		5	D3
D4	6	↔	6	D4	↔	6	D4
D5	7		7	D5		7	D5
D6	8	↔	8	D6	↔	8	D6
GND	14		14	GND		14	GND
/ACK	9	NC	9	/ACK	NC	9	/ACK
READY	10	→	10	READY	→	10	READY
GND	15		15	GND		15	GND
/INIT	11	←	11	/INIT	←	11	/INIT
GND	14		14	GND		14	GND
SELECT	12	NC	12	SELECT	NC	12	SELECT
/TRIGIO	13	←	13	/TRIGIO	←	13	/TRIGIO
GND	15		15	GND		15	GND

C.3 CA-83: PC Parallel Port to Master Interface Cable

PC LPT Connector 25-pin Male D-Shell		Signal Direction	Slave Connector 15-pin Male D-Shell	
Master	Pin		Pin	Slave
/STB	1	→	1	/STB
GND	18-21		14	GND
D0	2	↔	2	D0
D1	3	↔	3	D1
D2	4	↔	4	D2
D3	5	↔	5	D3
D4	6	↔	6	D4
D5	7	↔	7	D5
D6	8	↔	8	D6
GND	18-21		14	GND
D7	9	NC		
/ACK	10	←	9	/ACK
GND	22-25		15	GND
BUSY	11	←	10	READY
GND	22-25		15	GND
PAPER OUT	12	←	14	GND
SELECT	13	←	12	High
/AUTO LF	14	NC		
/ERROR	15	←	12	High
GND	18-21		14	GND
/INIT	16	→	11	/INIT
GND			14	GND
SELECT IN	17	NC		
		NC	13	/TRIGIO

C.4 CA-84: MP488 Digital Output Port (CA-59) to Master Interface Cable

CA-84 is also compatible with Metrabyte PIO-12)

CA-59 Connector 37-pin Male D-Shell		Signal Direction	Slave Connector 15-pin Male D-Shell	
Master	Pin		Pin	Slave
PA7	30		1	/STB
GND	11, 13		14	GND
PA0	37		2	D0
PA1	36		3	D1
PA2	35		4	D2
PA3	34		5	D3
PA4	33		6	D4
PA5	32		7	D5
PA6	31		8	D6
GND	11, 13		14	GND
PC6	23		9	/ACK
GND	11, 13		14	GND
PC7	22		10	READY
GND	11, 13		14	GND
		NC	11	/INIT
		NC	12	SELECT=V_{cc}
		NC	13	/TRIGIO
GND	11, 13	———	14	GND
GND	15, 17, 19	———	15	GND
	1-10	NC		
	12	NC		
	14	NC		
	16	NC		
	18	NC		
	20-29	NC		

C.5 CA-85: PC Serial Port to Master Interface Cable

PC/XT Serial Connector 25-pin Female D-Shell		Signal Direction	Master Connector 9-pin Female D-Shell	
Master	Pin		Pin	Slave
TxD	2	——►	2	RxD
RxD	3	◄——	3	TxD
RTS	4	NC		
DSR, CTS	5, 6	◄——	7	RTS
GND	7	——	5	GND
CD	8	◄——	7	RTS
DTR	20	——►	8	CTS
RI	22	NC		
	1	NC		
	9-19	NC		
	21	NC		
	23-25	NC		
		NC	1	
		NC	6	
		NC	9	

C.6 CA-86: PC/AT Serial Port to Master Interface Cable

PC/AT Serial Connector 9-pin Female D-Shell		Signal Direction	Master Connector 9-pin Female D-Shell	
Master	Pin		Pin	Slave
CD	1	←	7	RTS
RxD	2	←	3	TxD
TxD	3	→	2	RxD
DTR	4	→	8	CTS
GND	5	—	5	GND
DSR, CTS	6	←	4	DTR
RTS	7	→	1	
CTS	8	←	7	RTS
RI	9	NC		