53A-412 PROGRAMMABLE DIGITAL I/0 CARD

OPERATING MANUAL

Printed in U.S.A.

WARRANTY

Colorado Data Systems, Inc. (CDS) products (hardware and firmware) are warranted against defects in materials and workmanship, and are warranted to meet the performance specifications as listed in the current catalog and/or data sheet for the specific product being warranted. This warranty applies for three (3) years following the date of shipment. CDS will, at its option, repair or replace, at no cost to the customer, products which prove to be defective during the warranty period, provided the defect or failure is not due to misuse or abuse of the product. The customer is responsible for shipment of the defective product to the CDS factory. Software products are supplied on a site license basis subject to the same performance warranty provisions; the materials and distribution provision applies to the distribution media only. NO OTHER WARRANTY IS EXPRESSED OR IMPLIED, INCLUDING WARRANTY FOR FITNESS OF PURPOSE. CDS SHALL, IN NO CASE, BE LIABLE FOR CONSEQUENTIAL DAMAGES.

TRADEMARK NOTIFICATION

GW BASIC is a registered trademark of the Microsoft Corp.

IBM and PC are registered trademarks of the IBM Corporation.

All other brand and product names are trademarks
or registered trademarks of their respective companies.

53A-412 PROGRAMMABLE DIGITAL I/0 CARD

OPERATING MANUAL

DESCRIPTION 1

CONTROLS AND INDICATORS

Address Select Switch 4
Power LED 4
Fuses 4
Function LEDs and Switches 4
Built-In Test Equipment 6
External Handshake Controls 6

SPECIFICATIONS 8

OPERATION

Overview 11
Command Protocol And Syntax 11
Command Summary 12
Command Descriptions 14

INSTALLATION 48

APPENDIX A
53/63 SERIES SYSTEM COMMANDS 49

APPENDIX B
INPUT/OUTPUT CONNECTIONS 50

APPENDIX C
SAMPLE BASIC PROGRAM FOR THE 53A-412 52

APPENDIX D
OPTIONS 59

53A-412 PROGRAMMABLE DIGITAL I/0 CARD

DESCRIPTION

The 53A-412 Programmable Digital I/O Card is a printed circuit board assembly for use in a CDS 53/63 Series System. The 53A-412 provides 48 TTL and CMOS compatible, bidirectional digital I/O lines. For programming convenience, and to help in defining proper test flow, these 48 programmable I/O signal lines are organized as six 8-bit bytes. Two available options further extend the usefullness of the Card by adding optical isolation or open collector capability to the I/O.

Each of the six bytes can be completely and independently configured under full program control. All commands and responses are normally in ASCII hex notation for ease of programming, and to insure compatibility with the widest range of systems controllers. However, a binary capability is also supplied standard for those applications requiring increased data transfer speed. Programmable parameters include:

- selection of any byte as either input or output
- definition of masks for input and output data
- I/O control on command basis, on external handshake, or on a programmable time count
- logic sense of input, output and handshake lines
- full reporting of operating parameters at any time

Data output can be controlled as bits, as individual bytes, and/or as groups of bytes. Control of the output be directly applied on a command basis, or further qualified and controlled by using external handshakes, or by means of a programmable time count.

Data input is also fully under program control. The card can report the state of all input bytes, groups of input and/or output bytes, and single bits of a byte. Input data can be updated on a command request basis, or on a qualified basis using external handshakes.

User-defined masks can be overlaid on the data prior to output. Masks may also be applied to individual input bytes before they are returned to the system controller to improve data post-processing speed and to promote ease of data interpretation.

The sense of inputs and outputs can be set to either active high or active low under program control. The active edge can also be programmed for handshake lines. All I/O lines are both TTL and CMOS compatible, with up to 24 mA of sink current provided for each output. With Option 01 installed, the I/O section of the card is fully isolated from system ground using opto-isolators and an isolated power supply contained on the card.

External (handshake) control signals are provided for output and input data control. Output data control signals are Ready For Data (RFD), Data Available (DAV), and External Tri-State control (ETS0 - ETS5). Input data control signals are Data Ready (DRD) and Data Acknowledge (DAK).

A binary I/O mode is provided which allows a total of 6000 bytes to be allocated for buffered binary I/O through the card. Two modes are provided for binary input: Mode 1 inputs data as long as there is free memory in the buffer to accept it, while Mode 2 overwrites the oldest data in memory when the input buffer is filled. Binary input is controlled by the Data Ready handshake. Data can be simultaneously read from the input buffer while new data is being strobed in, dynamically freeing memory for additional input. The card can also be programmed to generate an interrupt when the input buffer is full.

Binary output can be controlled by either the Ready For Data handshake or on a programmable time basis from 100 µs to 858 seconds. Two modes are provided: mode 1 outputs data as long as there is data in the output buffer. Mode 2 allows pre-loading the output buffer, then defining a 'thread' sequence to control how preloaded data is output. For mode 1, data can simultaneously be loaded into the card while data is being output. The number of bytes left to be output can be read at any time, and the card programmed to generate an interrupt when the output buffer is empty. For mode 2, an interrupt can be programmed to occur when a breakpoint occurs in the sequence.

The 53A-412 provides full access to system status information, which is especially helpful during system trouble-shooting, software de-bugging, and operational system checks. At any time, the system controller can read the state of the external handshake lines, the programmed I/O configuration, the programmed active edges of all handshake signals, which handshake signals are active, the programmed logic sense of each I/O byte, the tri-state condition of each output byte, and up-to-date error data.

BITE

Built-in-Test-Equipment (BITE) is provided on the card by an internal loop-back path that allows the card to be tested with its outputs tri-stated, verifying I/O paths for each byte. A self test is automatically performed on power up, or it can also be commanded. Front panel LEDs indicate the error status, data handshake signals, current tri-state conditions, input/output configuration, and individual I/O bits. In addition, the Query and Operation commands can be used to determine the current state of the card during operation.

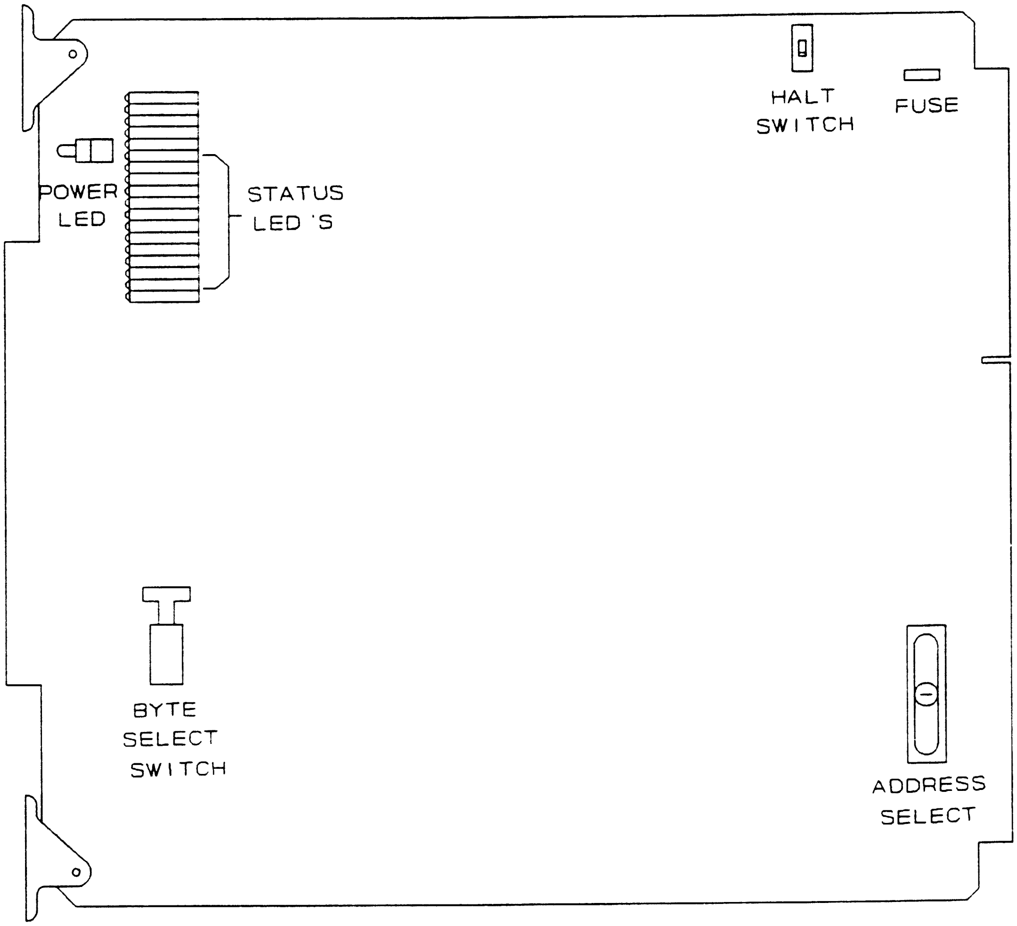


Figure 412-1: 53A-412 Controls and Indicators

CONTROLS AND INDICATORS

The following controls and indicators are provided to select and display the functions of the 53A-412 Card's operating environment. See Figure 412-1 for their physical locations.

Address Select Switch

The 53A-412 Card has a miniature 10-position switch which selects the card's address (0-9) in the 53/63 Series System. Open the switch's cover and use a screwdriver with a narrow, flat blade to turn the cam-action wiper to the desired position.

Power LED

The Power LED provides a valuable diagnostic tool by giving the system programmer a visual indication of the action which the system is currently taking. Whenever the 53A-412 Card is addressed by the system controller, the Power LED goes out. The LED remains out until another function card is addressed. Since only one function card can be addressed at a time, an unlit Power LED indicates the function card with which the system controller is currently communicating. The Power LED being lit not only indicates that the 53A-412 Card is unaddressed, but that all required dc power (5V dc) is being supplied.

Fuses

The 53A-412 Card has a single +5V fuse. The fuse protects the card in case of an accidental shorting of the power bus or any other situation where excessive current might be drawn.

If the +5 V fuse blows, remove the fault before replacing the fuse. Replacement fuse information is given in the Specifications section of this manual. Figure 412-1 shows the location of this fuse on the 53A-412 Card.

Function LEDs and Switches

LEDs

The following LEDs are provided at the top front edge of the 53A-412 Card to indicate the status of the card's operation:

ERR
Indicates a programming error has occurred. This LED will remain lit until the error condition is cleared.

RFD
Indicates the state of the Ready for Data handshake signal. This LED lights when the external device strobes ready for data indicating it is ready for more data. It is cleared when new data is output by the card.

DAV
Indicates the level of the Data Available handshake signal. This LED lights when the DAV line is low. It clears when the DAV line goes high.

DRD

Indicates the state of the Data Ready handshake signal. This LED lights when the external device strobes data ready indicating new input data is valid. It is cleared when the controller reads the input data.

DAK

Indicates the level of the Data Acknowledge handshake signal. This LED lights when the DAK line is low. It is cleared when the DAK line is high.

BYTE (B50, B51, B52)

Three LEDs that indicate in binary which of the six bytes (0 through 5) the bit LEDs (D7-D0 LEDs) are currently displaying, as follows:

Byte	LED status:		
Selected	B52	B51	B50
0	unlit	unlit	unlit
1	unlit	unlit	lit
2	unlit	lit	unlit
3	unlit	lit	lit
4	lit	unlit	unlit
5	lit	unlit	lit
*6	lit	lit	unlit
*7	lit	lit	lit

* Bytes 6 and 7 are unused.

I/O

Indicates the programmed input/output state of the current byte being displayed. The LED is lit if the byte is programmed as an output, and unlit if programmed as an input byte.

TRI

Indicates the tri-state condition of the byte currently being displayed. A lit LED indicates the byte is tri-stated.

D7 - D0

Indicates the state of each bit of the currently displayed byte. The LED being lit indicates the bit is high (TTL logic '1'). An unlit LED indicates the bit is low (TTL logic '0'). D7 is the most significant bit, and D0 the least significant bit. Bytes 6 and 7 are unused, and will light all the data bit LEDs.

Switches

The following switches are provided to select the proper functions for the 53A-412 Card's operating environment:

Halt Switch

This two-position slide switch is located near the card's backplane edge connector. It selects the state of the 53A-412 Card after an @XH (Halt) or STOP command is received by the 53/63 Series System.

a. If the Halt switch is in the ON position, then the 53A-412 Card is reset to its power-up state, all parameters are reset to their default values, and the Power LED is lit.

b. If the Halt switch is in the OFF position, then the 53A-412 Card becomes unaddressed, the Power LED is lit, and any programmed parameters of the card remain unchanged.

Byte Select Switch
The Byte Select switch located on the lower left is a momentary action switch that controls which of the six I/O bytes is currently being displayed on the LEDs. Each time the switch is depressed, the state of the next byte in sequence is displayed on the LEDs. For example, if the state of byte 0 is currently displayed, the state of byte 1 will be displayed after the switch is depressed. The BYTE LEDs will display the number of the selected byte (see BYTE in the listing of LEDs).

Built-In Test Equipment

BITE (Built-In Test Equipment) is provided on the card by an internal loop-back path, which allows the card to be tested with the outputs tri-stated. The self test automatically tests and verifies all loop-back paths for each byte.

Self test is automatically performed on power up, and can also be commanded. All the outputs are checked with their corresponding inputs, and with the output drivers in tri-state.

Front panel LEDs indicate the status of various system operating parameters. In addition, the Query and Operation commands can be used to determine the current state of the card during operation, including error codes (see the Query command in the Command Descriptions subsection).

External Handshake Controls

Output Data

Ready For Data (RFD)
Ready For Data is an input from an external device indicating it is ready for data. This signal is programmable to be either positive or negative edge triggered true.

Data Available (DAV)
Data Available is an output to an external device indicating valid data is available on the outputs. This signal is programmable to be either positive or negative edge triggered true.

External Tri-State control (ETS0 - ETS5)
Six external tri-state control lines are provided, one for each of the six bytes. These lines are inputs from an external device which cause the corresponding byte to go into tri-state (high impedance). The tri-state control lines are programmable to be either active high or active low.

Input Data

Data Ready (DRD)

Data ready is an input from an external device indicating valid data is at the inputs. This signal is programmable to be either positive or negative edge triggered true.

Data Acknowledge (DAK)

Data acknowledge is an output to an external device indicating the input data has been accepted. This signal is programmable to be either positive or negative edge triggered true.

SPECIFICATIONS

Number of I/O Channels: 48.

Configuration: I/O lines selectable as input or output on an 8-bit byte basis. Also tri-state programmable on an 8-bit byte basis.

Byte Transfer Polarity: All input and output bytes individually selectable as active high or active low.

Input Data: Returned as two hexadecimal ASCII characters per byte, or in binary.

Input Control: On program command or with external Data Ready and Data Acknowledge handshake.

Output Data: Programmed as two hexadecimal ASCII characters per byte, or on an individual bit basis, or in binary.

Output Control: On program command, with external Ready for Data and Data Available handshake, or on a programmable time count.

Tri-State Control: On program command on an individual byte, and by individual external tri-state control signals for each byte.

Mask Capability: On an individual byte basis, for input or output. AND, OR, and XOR (exclusive OR) masking provided.

Byte Ordering: A predefined sequence for input or output byte transfer may be programmed. Bytes may be transferred in any required order.

Interrupt Modes: Program selectable, on Ready For Data handshake, Data Ready handshake, Binary Input Buffer Full, Binary Output Buffer Empty.

External Control Logic Sense: Data Available, Ready For Data, Data Acknowledge, and Data Ready control line polarities are all individually program selectable as low or high true.

I/O Signal Type: TTL and CMOS compatible (74AHCT245 driver). Option 02 provides TTL open collector outputs (74LS641).

D.C. Electrical Characteristics: -10° to +55° C., typical specs at 25° C. A minus sign indicates current flowing out of the card.

Isolation Resistance: (Option 01 only) > 100e6 ohms at 500V dc.

Isolation Voltage: (Option 01 only) > 250V dc.

	min	typ	max	units
Output high voltage (Voh)				
Io = -20 μA	4.4	5.0		V
Io = -6 mA	3.84	4.2		V
Output low voltage (Vol)				
Io = 20 μA		0	0.1	V
Io = 24 mA		0	0.5	V
Output low current (Iol)			24	mA
Input high voltage (Vih)	2.0			V
Input low voltage (Vil)			0.8	V
* Input current (Iin)			230	μA
Tri-state leakage current (Ioz)		0.5	5.0	μA

* There are 22K pull-up resistors to +5V on all I/O and handshake lines to account for floating inputs. The input IC uses 1.0 μA max, while the pull-down resistors require 5V / 22K = 227.6 μA.

External Control Lines:

External Tri-state Input to Tri-state Active: Typical: 30 nS.
Maximum: 63 nS.

Valid Output Data to Data Available Strobe: 0 nS.

Data Acknowledge to Data Ready Strobe Delay: 0 nS.

Maximum Output Data Rate:
Buffered binary data using either the internal trigger or the RFD handshake:
10 KHz (1 to 3 output bytes)
8 KHz (4 to 6 output bytes)

ASCII hex data: 8 KHz.

Maximum Input Data Rate: Buffered binary data using the DRD handshake: 5 KHz.

Power Requirements: All required dc power is provided by the internal power supply in the 53/63 Series Card Cage.

Voltage (5-volt Supply): 4.75 V dc to 5.25 V dc.

Current (5-volt Supply): 1.10 A, maximum quiescent.
3.2 A, peak.

With Opt. 01: 1.7 A, maximum quiescent.
3.8 A, peak.

Power-up Defaults: All I/O pins tri-stated.
All bytes defined as inputs, active high.
All external handshake lines disabled.
Interrupts disabled.

Fuses: Replacement fuse: CDS P/N 42202-52001 (4 or 5 amp).

Cooling: Less than 10°C temperature rise with 1.2 liters/sec. of air at a pressure drop of 0.03 mm of H_2O.

Temperature: 0°C to +50°C, operating.
-40°C to +85°C, storage.

Humidity: Less than 95% R.H. non-condensing, 0°C to +30°C.
Less than 75% R.H. non-condensing, +31°C to +40°C.
Less than 45% R.H. non-condensing, +41°C to +50°C.

Dimensions: 197 mm x 221 mm x 19 mm (7.75 in x 8.69 in x 0.5 in)

Dimensions, Shipping: When ordered with a CDS card cage, this card will be installed and secured in one of the instrument card slots (slots 1 - 12).

When ordered alone, the card's shipping dimensions are:

254 mm x 254 mm x 127 mm.
(10 in x 10 in x 5 in).

Weight: 0.34 kg. (0.74 lb).
With Option 01: 0.42 kg. (0.93 lb.)

Weight, Shipping: When ordered with a 53/63 card cage, this card will be installed and secured in one of the function card slots.

When ordered alone, the card's shipping weight is:
0.8 kg. (1.64 lb).

With Option 01: 0.92 kg. (1.83 lb.)

Mounting Position: Any orientation.

Mounting Location: Installs in any function-card slot of the 53/63 Series Card Cage.

Front Edge Signal Connector: 1 - 76 pin connector.
Refer to Appendix B for connector pinouts.

Equipment Supplied:
1 - 53A-412 Card.
1 - Spare fuse (Part # 42202-52001)
1 - Operating Manual (Part # 00000-14120).
1 - Service Manual (Part # 00000-24120).

Optional Equipment: Option 01: Optical Isolation.
Option 02: Open collector I/O.
See Appendix D for a description of the options.

OPERATION

Overview

The 53A-412 Card is programmed by ASCII characters issued from the system controller to the 53/63 System's communications card. The 53A-412 Card is interfaced to the communications card through the 53 Series or 63 Series Card Cage's backplane.

To address a function card for the first time, the system command @XY must be issued. X is the card cage address (0-9) selected on the 53A-171 Control Card in the addressed card cage; Y is the 53A-412 Card's address (0-9) within the addressed card cage. The 53A-412 Card's address is selected using the card's Address Select switch. Once a function card is addressed, it remains addressed until the system receives another @ character. Appendix A fully discusses the @XY command and the other 53/63 Series system commands. After the 53A-412 Card is addressed, the commands listed below may be issued until another function card is addressed.

Command Protocol And Syntax

Command protocol and syntax for the 53A-412 Card is as follows:

1) A command string consists of a string of ASCII encoded characters (up to 1024 maximum) terminated by a line feed <LF>. Multiple commands may be entered within a string, with a semi-colon used as a delimiter between individual commands.

2) Valid command delimiters for the card are the line feed <LF> and the semi-colon (;). The command string is buffered up until a <LF> is encountered, at which time the entire string is evaluated.

   If no delimiter is entered, then the card interprets the next command as being part of the previous command. This causes either an error or an improper setup if the subsequent command is interpreted as part of the previous command.

3) All commands are operated on in the order they are received, and executed when the <LF> delimiter is received.

4) If a given parameter is omitted within a command, either its default state or its last programmed state will be in effect (depending on the particular command issued).

5) Any character may be sent in either upper or lower case form.

6) Any command syntax or programming errors will cause the command to be ignored, and an error will be flagged. All commands up to the occurrence of the error will remain valid. The invalid command and all subsequent commands will be lost, and no commands will be accepted until the error condition is cleared, either through a hardware or software reset, or by reading the error out with the Q command. If an error is queued, the card will respond with 'ERROR' until the condition is cleared.

7) All responses from the card are terminated by a carriage return and line feed <CR><LF>.

8) The table below shows the valid ASCII character set for the 53A-412. All other characters are treated as white space characters, and are ignored if received.

Character	ASCII Code (hex)
<LF>	0A (line feed)
#	23
&	26
*	2A
+	2B
-	2D
.	2E
/	2F
’0’ – ’9’	30 – 39
;	3B
’A’ – ’Z’	41 – 5A
’a’ – ’z’	61 – 7A

Command Summary

Detailed descriptions of each command (in alphabetical order) are given following the summary. An overview of the commands is as follows:

I The Input Data command specifies which bytes are to be read, the order in which they are to be read (and reported), and any masks to be overlaid onto the data prior to reporting it. ASCII hex data representing all input bytes, selected input and/or output bytes, or selected bits of a byte (by using a mask) can be returned to the system controller using this command.

L The Load Output command specifies the data to be output, the order of output, and any masks to be overlaid onto the data prior to output. The ASCII hex data representing all output bytes, selected output bytes, single bits of a byte, or mask overlays onto the byte(s) can be used to update the cards' output data latches.

M The Mode command defines which bytes are inputs and which are outputs, and their active logic sense (active high true or active low true).

N These commands set up and control buffered binary input and output.

O The Operational Setup command returns information on the current setup of the card.

P The Strobe Pulse Sense command specifies the active edge of the handshake signals (positive or negative edge triggered).

Q The Query Status command is used to read the current state of the card. The information which can be obtained includes:

- error data;
- the state of the external handshake lines (DRD,RFD);
- the current tri-state condition of the I/O latches;
- the programmed active edges of the handshake signals, and whether the handshake(s) are active;

- the amount of data queued in the output binary buffer;
- the amount of data queued in the input binary buffer;
- the current thread sequence and its state.

R The Reset command resets the board to its power-up state.

S The Self Test command causes the card to execute a self test, and then return to its power-up state.

T The Tri-state Control command specifies whether the output bytes are tri-stated (high-impedance), or active. This command is logically OR'ed with the external tri-state lines.

U The Update command specifies the conditions for which the inputs and outputs are updated (update on command, update on external handshake control, update on time), and for update on time, specifies the timing interval.

V The Version command returns the current software revision level of the board.

X The X command is used to enable or disable the interrupt to the system controller. This interrupt can be programmed to be active when either external handshake is valid (DRD, RFD), when the binary input buffer is full, or when the binary output buffer is empty or a breakpoint occurs.

Z The Tri-state Level command specifies the active level of the external tri-state control lines ETS0-ETS5 (active high true or active low true).

A detailed description of each command, in the same order as listed above, is given on the following pages. The syntax used in the command descriptions is:

( ) optional parameter
{ } group of parameters
' ' ASCII character
... optional repetition

NOTE: The (), {}, ' ', and ... characters are not part of the command.

Command Descriptions

Command | **Description**

I or IO

The I (Input) command specifies the data to be input, the order in which it is be input, and any masks which are to be overlaid onto the data prior to reporting it.

The IO (Input Override) command provides the capability to read a different input sequence one time, without destroying the last defined I command input sequence.

Syntax:
```
I{b(o)(d)(/) }...
IO{b(o)(d)(/) }...
```

I input command
IO input override
b one to six digits which specify the byte number, 0 through 5, or * for all bytes.
o one of the following:
  & AND the data specified by (d) to the specified input byte(s).
  # OR the data specified by (d) to the specified input byte(s).
  X XOR the data specified by (d) to the specified input byte(s).
d ASCII mask value 00 through FF (required with o)
/ an optional character which is allowed to make the command more readable.

Default: I* (input all bytes)

The bytes can be programmed in any order, and once programmed, the setup remains valid unless specifically redefined by another I command, or by a Reset or Self Test command. Input can be requested for both input and output bytes.

Typical use of the I command simply defines a sequence of bytes to be read, with the sequence defined by the order of the digits following the I command. For example, 'I123' specifies that the data from bytes 1, 2, and 3 are to be reported in the order of byte 1 first, byte 2 second, and byte 3 third (followed by <CR><LF>).

Additional input of the same sequence does not require redefining the I command. Successive reads from the card will return new data in the defined sequence, each terminated by <CR><LF>.

For example, 'I321' specifies an input sequence of 3-2-1. If bytes 1, 2, and 3 contained 10, 22, and 43 hex, the card would report '432210<CR><LF>' when read. Subsequent reads of the card will report the update state of bytes 3, 2, and 1.

A * in the I command automatically defines the byte sequence to be 0-1-2-3-4-5.

Each time an I command is issued, it defines a new input sequence. The input override command (IO) is used to look at a specific byte(s) without affecting the I command's sequence, as shown in the example below. Once the I command has been issued, its setup and sequence (including masks) remain valid until overridden by another I command, reset, or self test.

If external Data Ready Strobe has been defined as the condition to latch input data into the card, and no strobe has been received since the last input request, an 'N<CR><LF>' will be returned for both the I and IO commands, indicating no new data is available.

The state of the data returned represents the logic sense programmed with the mode (M) command.

If an I command is issued with no arguments ((b) is omitted), the sequence will be cleared and the card will respond with a <CR><LF> only. If (o) and (d) are omitted, the command specifies data in its raw input form. If (o) is specified without (d), an error will be generated. If any error is queued, the card will respond with ERROR<CR><LF> on the subsequent input requests.

NOTE: All responses from the I and IO commands are terminated in <CR><LF>.

Example:
The example cases on the following page show how a sequence of I commands and implicit inputs will be reported (each case assumes the I/O lines are at 00, 11, 22, 33, 44, and 55 for bytes 0 to 5 respectively).

		Byte	
Case	Command	Sequence	Card Response
Setup	M*0;T*I;L*;001122334455;I*<LF>		
1	Initial State	0-1-2-3-4-5	001122334455<CR><LF>
2	I123<LF>	1-2-3	112233<CR><LF>
3	read (no I command)	1-2-3	112233<CR><LF>
4	I*&55<CR><LF>	0-1-2-3-4-5	001100114455<CR><LF>
5	read (no I command)	0-1-2-3-4-5	001100114455<CR><LF>
6	I*<LF>	0-1-2-3-4-5	001122334455<CR><LF>
7	IO3X11<LF>	3	22<CR><LF>
8	read (no I command)	0-1-2-3-4-5	001122334455<CR><LF>
9	I543012<LF>	5-4-3-0-1-2	554433001122<CR><LF>
10	I0#55/1XAA/2345<LF>	0-1-2-3-4-5	55BB22334455<CR><LF>

Setup sets all bytes to outputs (M command); un-tri-states them (T command); defines the load sequence as 0-1-2-3-4-5 (L command); loads the specified data; and inputs all the data in the 0-1-2-3-4-5 sequence (I command).

Case 1 is the initial setup condition.

Case 2 requests the input from bytes 1, 2, and 3 in that order.

Case 3 reports the data from the input sequence set up in case 2.

Case 4 masks (ANDs) each input byte with a 55 hex prior to reporting it. Note that the <CR> character is treated as a white space and is ignored.

Case 5 reports the data using the sequence and mask set up in case 4.

Case 6 overrides the mask from case 4 and reports the data in its raw form.

Case 7 uses the override command to look at byte 3 XOR'd with an 11 hex.

Case 8 reports the data from the sequence defined in Case 6.

Case 9 reports the data in the newly defined sequence 543012.

Case 10 reports the data in the newly defined sequence 012345, ORs byte 0 with a 55 hex, XORs byte 1 with an AA hex, and reports bytes 2, 3, 4, and 5 in their raw form. The optional / is used to make the command more readable.

Command Description

L or LO

The L (Load Output) command specifies the data bytes to be output, the sequence in which it is be output, and any masks which are to be overlaid onto the data prior to output by the card.

The LO (Load Override) command provides the capability to output a different sequence of bytes one time, without destroying the last defined L command output sequence.

Syntax: L{b(o)(d)(/) }...
LO{b(o)(d)(/) }...

L load output command

LO load override

b one to six digits which specify the byte number, 0 through 5, or * for all bytes. A * automatically defines the sequence 0-1-2-3-4-5.

o one or more of the following letters which specify various parameters as follows:

- D Load the data specified by (d) to the specified output byte(s).
- S Set the bit indicated by (d) to a logic high (the eight bits of a byte are defined as 00 through 07, with bit 00 being the least significant bit).
- R Reset the bit indicated by (d) to a logic low (the eight bits of a byte are defined as 00 through 07, with bit 00 being the least significant bit).
- & AND the data specified by (d) to the specified output byte(s).
- # OR the data specified by (d) to the specified output byte(s).
- X XOR the data specified by (d) to the specified output byte(s).

d ASCII value 00 through FF (required with o). Note that d is an 8-bit wide byte value if the o parameter is a D, &, #, or X, and a bit number if the o parameter is an S or R.

/ optional character allowed to make the command more readable.

Default: NONE [all bytes are initially defined as inputs (M command), set to TTL logic 0, and tri-stated (T command)].

The bytes can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another L command, by redefining the I/O configuration (M command), or by a reset or self test command. If any bytes are not programmed, they will remain in their default (or previously programmed) state. The Set, Reset, and Mask parameters (S, R, X, #, and &) all operate on the last data output to a byte(s), and are valid only for the current command.

Typical use of the L command specifies an output byte and the data to be output. For example, 'L2D55' specifies that a 55 hex is to be output to byte 2.

Once an output sequence has been defined with the L command, ASCII hex data (00 - FF) may be written to the card without additional L commands.

The order in which the outputs were specified in the last Load command defines an output sequence. For example, 'L321D55' specifies that a 55 is to be output to bytes 3, 2, and 1. This command also defines the output sequence to be bytes 3-2-1. ASCII hex data sent to the card will then be buffered until the amount of data received matches the amount required by the output sequence. All bytes are thus physically updated at the same time when the total amount of data is received. In this example, six ASCII hex bytes are required since two bytes are required for each output byte. If '123456' were then sent to the card, byte 3 would be loaded with 12, byte 2 with 34, and byte 1 with 56, to match the 3-2-1 sequence. If an L command had not been previously issued, this data would be ignored.

Each time an L command is issued, it defines a new output sequence. The Load Override (LO) command is used to change specific data without affecting the L command's sequence, as shown in the examples below. Note that whenever a new L command is issued, any buffered data in an incompleted buffer is lost. The output sequence is also cleared whenever a new Mode (M) command is issued.

Note that a particular byte should only be defined once within the L command, because it can appear only once in the sequence. If a byte is defined more than once within the command, only the last specified action is taken. For example, 'L0D55/0D44' would load a 44 hex into byte 0 (the load 55 hex action is ignored). Similarly, 'L0S01/0S03' would only set bit 3 of byte 0. Setting both bits can be accomplished by using the mask command 'L0#0A'.

The byte(s) will be physically output based on the conditions defined by the U command.

If output is commanded to a byte which is defined as an input (M command), an error will be flagged, and the command ignored. If the (b) parameter is omitted, the load sequence will be cleared and the card will respond with a <CR><LF> only. The (o) and (d) parameters are optional. However, if (o) is specified without (d), an error will be generated. If an invalid parameter is specified, an error will be generated.

Note that when using the RFD external handshake, the most recent data received by the card is always the next to be output. If two L commands, or two full buffers of data are received before a strobe occurs, the first data will be lost, and the most recent data will be output. To prevent this overwriting of data, read the state of Ready For Data (RFD) with the QR command (see Query Status command) before sending additional data to the card. If the data reported back by the QR command is a 0, then the last data output has not yet been accepted by the external device. If a 1 is reported back, then the outputs can be updated with no loss of data.

Example:
The following examples show how a sequence of L commands and data will be output.

Case	Command	Byte Sequence	Output Data Bytes (hex) 0	1	2	3	4	5
1	Power-up (default)	none	--	--	--	--	--	
2	M*O;T*I;I*<LF>	N/A	00	00	00	00	00	00
3	L*D55<CR><LF>	0-1-2-3-4-5	55	55	55	55	55	55
4	001122334455	0-1-2-3-4-5	00	11	22	33	44	55
5	L1D0150DFA2D204D883DCC<LF> or L1D01/50DFA/2D20/4D88/3DCC<LF>	1-5-0-2-4-3	FA	01	20	CC	88	FA
6	001122334455	1-5-0-2-4-3	22	00	33	55	44	11
7	LO1S04<LF>	no change	22	10	33	55	44	11
8	L123#80<CR><LF>	1-2-3	22	90	B3	D5	44	11
9	001122	1-2-3	22	00	11	22	44	11
10	L150243<LF>	1-5-0-2-4-3	22	00	11	22	44	11
11	001122334455	1-5-0-2-4-3	22	00	33	55	44	11
12	L*D33<LF>	0-1-2-3-4-5	33	33	33	33	33	33
13	L0S02/1R04/2&22/3X22/4#44<CR><LF>	0-1-2-3-4	37	23	22	11	77	33
14	0011		37	23	22	11	77	33
15	223344		00	11	22	33	44	33
16	LO41D55<LF>		00	55	22	33	55	33
17	AABBCCDDEE		AA	BB	CC	DD	EE	33

Case 1 is the initial state of the outputs. All outputs are in a tri-state condition.

Case 2 defines all bytes as outputs and un-tri-states them. The I* command at the end of the line can be used to read back the output data and verify that it is all 0s, if an input request is issued following this command.

Case 3 loads all outputs with 55 hex, with the * defining the sequence as 012345.

Case 4 is data received from the system controller. The data is output in the order it is received according to the current sequence.

Case 5 loads each byte individually, and redefines the sequence to be 150243. The line below case 5 shows the same command using the optional / character.

Case 6 is more data, again output in the order it is received, according to the current sequence.

Case 7 uses the load override command to force bit 4 of byte 1 high without changing the sequence.

Case 8 OR's the current data of bytes 1, 2, and 3 with an 80 hex, and redefines the output sequence to 123.

Case 9 loads new data into bytes 1, 2, and 3.

Case 10 redefines the output sequence without affecting the data.

Case 11 loads data for the newly defined sequence.

Case 12 loads all bytes with 33 hex.

Case 13 sets bit 2 of byte 0, resets bit 4 of byte 1, AND's byte 2 with hex 22, XOR's byte 3 with hex 22, and OR's byte 4 with a hex 44.

Case 14 has no effect on the outputs because not enough data has been received based on the last sequence defined (01234).

Case 15 supplies the rest of the data needed for the sequence, and the new data is output.

Case 16 uses the override command to force bytes 4 and 1 to a hex 55.

Case 17 outputs new data based on the sequence from case 13, which is still in effect.

Note that each time a Load command is received, a new sequence is defined for any subsequent data, and that the Load Override command does not affect the output sequence.

Command   Description

M

The M (Mode) command defines which bytes are inputs and which are outputs, and their active sense.

Syntax: M {(b)(m)(l)}...

b   byte number, 0 through 5, or * for all bytes.
m   I or O, Input or Output respectively
l   H or L, Logic state, High or Low true respectively.

Default: M*IH (all inputs, active high true)

The bytes can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another M command, or by a Reset or Self Test command. If any bytes are not programmed, they will remain in their default (or previously programmed) state.

If (m) or (l) is omitted, the default (or previously programmed state) will be used for the omitted parameter of the byte(s) being programmed. If (b) is omitted, the command will have no effect. If an invalid parameter is sent, an error will be generated.

If the logic state is programmed as active high true, then a 1 on an input or output command reflects a TTL logic 1 on the I/O pin. If the logic state is programmed as active low true, then a 1 on an input or output command reflects a TTL logic 0 on the I/O pin.

NOTE: The Mode command automatically resets the sequence set up by the L (Load) command to 'null', and clears any pending RFD handshakes.

Example:
The following examples show how a sequence of mode commands will affect the configuration setup of the card:

Case	Command	Byte I/O and Sense (H or L)					
		0	1	2	3	4	5
1	Power-up (default)	I/H	I/H	I/H	I/H	I/H	I/H
2	M3O<LF>	I/H	I/H	I/H	O/H	I/H	I/H
3	M105L<CR><LF>	I/L	I/L	I/H	O/H	I/H	I/L
4	M012OH345IL<LF>	O/H	O/H	O/H	I/L	I/L	I/L
5	M*OL<CR><LF>	O/L	O/L	O/L	O/L	O/L	O/L
6	M23I*H<LF>	O/H	O/H	I/H	I/H	O/H	O/H

Case 1 is the power-up default state.

Case 2 sets up byte 3 as an output. Since the logic sense was not specified, it remains in its previously programmed (default) state of active high.

Case 3 sets up bytes 1, 0, and 5 as active low. All other bytes remain in their previously programmed state. A <LF> is used to delimit this command with the <CR> being ignored.

Case 4 sets up bytes 0, 1, and 2 as outputs, active high, and bytes 3, 4, and 5 as inputs, active low.

Case 5 sets all bytes to outputs, active low.

Case 6 sets up bytes 2 and 3 as inputs, and all bytes as active high.

Command Description

N The N commands are used to set up and control binary input and output.

For binary input, the commands are:

NIA - allocates the size of the binary input buffer and the binary input mode.
NIT - enables the binary input
NIX - transfers binary data from the card
NID - disables the binary input mode

related commands:

M - (mode) define which bytes are inputs and outputs
I - (input) define the input data sequence
UD - (update) update the input on the data ready handshake
QB - (query) returns the number of bytes queued in the input buffer
XAB - (interrupt) generates an interrupt when the input buffer is full

For binary output, the commands are:

NOA - allocates the size of the binary output buffer and the binary output mode.
NOX - preloads/loads binary data into the output buffer
NOS - defines a threading sequence for output mode 2
NOC - continues the thread sequence after a breakpoint for output mode 2
NOR - redefines part or all of a thread for output mode 2
NOT - enables the binary output
NOD - disables the binary output mode

related commands:

M - (mode) define which bytes are inputs and outputs
L - (load) define the output data sequence
U - (update) define the update condition as the ready for data interrupt or the programmable timers.
QO - (query) returns the number of bytes still in the output buffer that have not yet been transmitted (output mode 1)
QK - returns the current thread sequence in effect and its break condition (output mode 2)
XAO - generates an interrupt when the output buffer is empty (output mode 1), or when a breakpoint occurs (output mode 2)

A detailed description of each of the N commands follows.

NIA / NOA — Allocate Binary Buffer Memory.

The NIA and NOA commands are used to set up and allocate binary input and output buffers respectively, and to define the binary mode.

Syntax:	NIA[x]M[y]	Allocate the binary input buffer memory
	NOA[x]M[y]	Allocate the binary output buffer memory

[x] amount of memory allocated; 0 to 6000 max.
[y] 1 or 2, the binary mode

A total of 6000 bytes is available for the input and output binary buffers. This memory can be split between the two buffers in any proportion. For example, if 4000 bytes were allocated to the output buffer, up to 2000 bytes could be allocated to the input buffer.

For an NOA command, the amount of memory allocated should be an integer multiple of the number of bytes defined as outputs (Load (L) command). For an NIA command, the amount of memory allocated should be an integer multiple of the amount of bytes defined as inputs (Input (I) command). For example, if 3 bytes were defined as outputs, and it is desired to output data 100 times, then (3 * 100) = 300 bytes should be allocated to the output buffer. Similarly, if 2 bytes were defined as inputs, and it is desired to input data 2500 times, then (2 * 2500) = 5000 bytes should be allocated to the input buffer.

Issuing an NOA or NIA command will reset the associated buffer addresses and counters to 0. If the total memory allocated to the two buffers exceeds 6000, then an error will be generated. The NOA or NIA command must be issued once before any other N output or input command, respectively.

Binary Input Modes

Binary input can only be used in conjunction with the data ready (DRD) handshake (see the Update (U) command). Two modes are provided for buffered binary input. Mode 1 (e.g NIA100M1) inputs data as long as there is free memory in the buffer to accept it. For mode 1, if a DRD handshake had occurred when the memory was full, the handshake will be queued until enough memory to input another sequence is freed. When the memory is freed, the data will be input and the handshake acknowledged.

Mode 2 (e.g NIA100M2) will overwrite the oldest data in memory if the input buffer is full. If the mode is improperly specified, an error will be generated. By using the NIX command, data can be simultaneously read from the input buffer while new data is being strobed in. This in effect dynamically frees up memory for additional input.

The number of bytes queued up in the input buffer can be read at any time by using the QB query command. Also, the X command can be used to generate an interrupt when the input buffer is full.

Binary Output Modes

Binary output can be used in conjunction with the ready for data (RFD) handshake, or the programmable timers (see the Update U command). Two modes are provided for buffered binary output. Mode 1 (e.g. NOA100M1) outputs data as long as there is enough data loaded in memory for another output sequence. For mode 1, if a RFD handshake had occurred when the memory was empty, the handshake will be queued until enough data has been loaded to output another sequence. The new data will then be output and the handshake acknowledged.

Mode 2 (e.g. NOA100M2) outputs the data based on a programmed thread sequence (NOS command). If the mode is improperly specified, an error will be generated. By using the NOX command, data can be simultaneously loaded into the output buffer while data is being output to the front end.

For mode 1, the number of bytes left to be output can be read at any time by using the QO query command. For mode 2, the QK command can be used to determine the current sequence and its state. Also, the X command can be used to generate an interrupt when the output buffer is empty for mode 1, or when a breakpoint occurs for mode 2.

****

NID / NOD

Disabling Binary I/O

The NOD and NID commands are used to disable binary outputs or inputs respectively.

Syntax:	NID	Disable the input binary transfers
	NOD	Disable the output binary transfers

****

NIX / NOX

Binary Data Transfer

The NIX and NOX commands are used respectively to read binary data from the card or to load binary data into the card.

Syntax:	NIX[x]/[y]	Transfer binary data from the card.
	NOX[x]/[y]	Transfer binary data to the card.

[x]	amount of binary data to transfer, 0 to the allocated buffer size
[y]	optional parameter specifying a specific buffer location

If [x] or [y] is less than zero or greater than the respective allocated buffer size, an error will be generated.

Loading Binary Output Data

For the NOX command, the card will go into the binary input mode to accept binary data from the backplane. The card will stay in binary until either the specified amount of data has been loaded, or the card is read from. The data will be loaded and output based on the current output sequence as defined by the Load command. For example, if three bytes were defined as outputs and output sequence was 5-4-0 (L540), then the first byte loaded would correspond to latch 5, the second to latch 4, the third to latch 0, the fourth to latch 5, the fifth to latch 4, etc. [y] is an optional parameter which specifies where the data is to be loaded in the output buffer.

NOTE: The [y] option is primarily intended for preloading the output data. To prevent erroneous operation, do not use it if the card is actively outputting data.

If [y] is specified, the data will be loaded beginning at the address specified by [y]. For example, an NOX1000/0 command would load the subsequent downloaded binary data into the output buffer beginning at address 0 (the first location) of the binary output buffer. To load an additional 1000 bytes, issuing either an NOX1000 or NOX1000/1000 would then load the data into the next 1000 locations. The card automatically wraps around to the beginning of the buffer when the end of the buffer is reached. Note that the NOA command automatically sets the starting address to 0.

For output mode 1, once the data has been initially loaded into the card and the binary output enabled (NOT command), the NOX command can be used to dynamically load additional data into the output buffer as memory is freed. A QO command should first be executed to verify there is enough free memory in the buffer to download the additional data (to prevent erroneous operation by possibly overwriting data which has not yet been output). After receipt of the NOX command the data will be loaded into the buffer following the data still queued to be output. For example, if 30 bytes were still left to be output and an NOX1000 command were issued, the new downloaded binary data would be loaded immediately following the location of the 30th data byte. As each byte is transferred, the counter (QO) response is automatically incremented to reflect the additional amount of data loaded into the card.

For output mode 2, the NOX command can be used to download the individual threads into specific locations in memory. After the data has been loaded and the thread sequence defined (NOS command), the NOT command can be used to begin the output.

If extreme caution is used, new data can be loaded into the card to dynamically change a data thread. To do this, either the thread should be at a breakpoint, or the new data loaded should be loaded into allocated but unused output buffer memory. For either of these cases, the new data can be loaded without the chance of overwriting the data of an active thread. Also, the NOR command can be used to redefine the threading sequence if desired at this point.The NOC command is used to restart the sequence after a breakpoint. The QK command can be used to determine the current sequence in effect and its state (executing or at a breakpoint).

Reading Binary Input Data

For the NIX command, the card will go into the binary input mode to transfer binary data from the input buffer to the backplane. The card will stay in binary mode until either the specified amount of data has been transferred, or the card is written to. The QB command can be used to determine how much data is queued up in the input buffer. The NIX command automatically outputs data beginning from the oldest data in memory. It can be used to dynamically transfer data from the card (thus freeing up memory for more input) while data is actively being input into the card.

For example, if 100 input bytes were queued, issuing an NIX50 would return the oldest 50 bytes in memory (virtual locations -100 to - 51). Issuing another NIX50 command would then return the next 50 bytes (virtual locations -50 to -1). As each byte is transferred, the QB response is automatically decremented by one to indicate the new amount of data queued in the input buffer. The data is reported based on the input sequence as defined by the input (I) command. For example, if three bytes were defined as inputs, and the input sequence was 3-2-1 (I321), then the first byte returned would correspond to input latch 3, the second to input latch 2, the third to input latch 1, the fourth to input latch 3, etc. The card automatically wraps around to the beginning of the buffer when the end of the buffer is reached.

[y] is an optional parameter which specifies where in the input buffer data is to be transferred from. If [y] is specified, it should be an integer multiple of the number of bytes defined by the input (I) command to prevent skewing of the reported data relative to the input sequence.

NOTE: The [y] option is primarily intended for post-processing of the input data (NID issued to stop the input). To prevent erroneous operation, do not use it if the card is actively inputting data.

[y] is referenced to the address of the next byte to be input, which is virtual address 0. If [y] is specified, the data is transferred beginning at the address specified by [y]. For example, issuing an NIX1/1 will transfer the most recent byte input by the card. Similarly, an NIX100/100 will transfer the most recent 100 bytes input by the card. As each byte is transferred, the QB response data is decremented by one. Note that the NIA command automatically sets the starting virtual address to physical address 0 of the input buffer.

****

NOS Defining A Thread Sequence For Output Mode 2

A thread is sequential data beginning at a specified address, and of a specified size. For example, one thread could begin at the output buffer address 100 and stretch 10 bytes. Another thread could begin at address 9 and stretch 18 bytes. The data in the threads is loaded by the NOX command. By defining a thread sequence with the NOS command, very specific data patterns and control can be effected on the output data. Up to 50 threads can be defined with the NOS command.

Syntax: NOS{[w]/[x]/[y][z]}...

[w] the starting address of the thread
[x] the number of data outputs
[z] the number of times to repeat this thread
[z] either B or C:
B = break after completing this thread
C = continue to the next thread after completing this one

[w] the starting address of the thread, and the physical location of the output data buffer where the thread is to begin.

[x] the number of data outputs for the current thread. Note that the actual number of bytes to be output will be the number of bytes defined as outputs by the load (L) command times [x]. For example, if 3 bytes were defined as outputs, and [x] were 10, then 30 contiguous bytes in memory would be required for the thread.

[y] the number of times to repeat the current thread. Using the above example, if [y] were defined as 2, then a total of 60 bytes would be output (3 * [x] * [y]). Since three bytes are output per handshake (or timer tick), a total of 20 handshakes (or ticks) would be required to output the thread.

[z] the action to be taken when the current thread is completed. If [z] = B, then output will be suspended (or broken) when the thread is complete. At this point, new thread data could be loaded with the NOX command, and/or the thread sequence could be modified using the NOR command. if [z] = C, then the data will continue to the next thread in the sequence. The first thread defined is thread 0, the next thread 1, etc. When the end of the threading sequence is complete, the sequence will automatically restart back to thread 0 if the last thread is programmed to continue to the next thread.

Issuing an NOS command automatically puts the card into output mode 2. The NOS command is used to define the threads for output mode 2 and will have no effect if specified in output mode 1.

If an RFD handshake occurs when a breakpoint is active, the handshake will be queued, and the next data output when the breakpoint is deactivated with an NOC command. If [w] or [x] is less than zero or greater than the allocated buffer size, an error will be generated. If [y] is less than zero or greater than 32767, an error will be generated. If the number of sequences defined is greater than 50, an error will be generated. Also, if any parameter is missing or improperly specified, an error will be generated.

To illustrate the threading operation, assume that byte 0 only has been defined as an output (L0), and that 10 sequential binary bytes beginning with 0 have been loaded into the output buffer at address 0 (NOX10/0 command). If the threading sequence were defined as NOS 0/5/1C 7/2/2C 3/3/1B then the data that would be output would be 0-1-2-3-4-7-8-7-8-3-4-5. Note that the spaces in the NOS are white space characters and ignored by the card. If the sequence were restarted with an NOCB command (continue from break), then the entire sequence would be repeated. If the sequence were again restarted with an NOC1 command (continue from break at sequence 1), then the data that would be output would be 7-8-7-8-3-4-5.

****

NOC Continuing A Thread From A Breakpoint (Output Mode 2)

The NOC command is used to continue the threading sequence after a breakpoint has occurred in output mode 2, and will be ignored if specified in output mode 1.

Syntax: NOC[x][I]

[x] B, or an integer value:
B continue from the current breakpoint to the next thread
integer value from 0 to the number of threads defined
continue from the breakpoint at the defined thread.

[I] optional parameter which specifies that the next thread is to be initiated regardless of the state of the RFD handshake.

If an NOCB command were issued, the thread following the one causing the break will be executed. If the thread is the last one in the sequence, it will be set back to the first one (thread 0). The threading can be continued from any point after a breakpoint by specifying which thread to continue from. For example, an NOC5 will continue with the sixth defined thread of the sequence (thread 0 is the first defined thread). The I option of the command is useful when it is necessary to 'bootstrap' the output to get the handshaking process started (e.g. NOCBI or NOC0I commands). If a thread number is defined outside of the total number of threads defined by the NOS command, an error will be generated.

****

NOR

Redefine Thread Parameters

The NOR command redefines some or all the parameters of a thread.

Syntax:	NOR[n][y][z]	redefine a particular parameter of a thread
	NOR[n]A[w]/[x]/[y][z]	redefine all the parameters of a thread

For specific parameters:

[n] the thread number
[y] one of the following letters:
S redefine the starting address of the thread
D redefine the number of data outputs
N redefine the number of times to repeat the current this thread
B break after completing this thread
C continue to the next thread after completing this one
[z] the new parameter value. [z] should not be issued for [y] = B or C.

The NOR command can be issued at any time once a sequence has been defined by the NOS command. However, caution should be exercised if data is being actively output by the card to prevent changing a sequence which is active. The QK command can be used to read the current active sequence, and its associated state (executing or at a breakpoint). To change the entire threading sequence, issue another NOC command.

If an NOS command of NOS1/2/3C 4/5/6B were issued, the following examples show how the applicable thread would be modified.

command	thread 0	thread 1
initial	1/2/3c	4/5/6b
nor0s9	9/2/3c	4/5/6b
nor1d8	9/2/3c	4/8/6b
nor0n10	9/2/10c	4/8/6b
nor1c	9/2/10c	4/8/6c
nor0b	9/2/10b	4/8/6c
nor0a5/4/3c	5/4/3c	4/8/6c

****

NIT / NOT

Enable Binary Input/Output Mode

Syntax:	NIT[I]	Enable (trigger) the binary input mode
	NOT[x][I]	Enable (trigger) the binary output mode

[x] the number of times to output the binary output data or to execute the threading sequence. * or 0 will output the data continuously until an NOD command is issued to disable the output.

[I] optional parameter to initiate the binary input or output regardless of the state of the applicable handshake line.

Enabling Binary Input

The NIT command is used to enable binary input. If the card has not been programmed to update on the data ready (DRD) handshake, the command will have no effect. The NIT command sets all buffer pointers to 0, clears any pending DRD handshakes, and clears the data queued counters (QB command).

The I option of the command is useful when it is necessary to 'bootstrap' the input to get the handshaking process started (e.g. NITI). If issued, the card will input the first data sequence regardless of the state of the data ready handshake. This will cause the data acknowledge (DAK) handshake to go active. Once the card has been set up for binary inputs (NIA command), the NIT command can be issued any time to restart binary input. For example, if an NID command were issued to disable binary inputs in order to look at the data, then issuing an NIT command will flush the buffer by clearing the pointers and counters, and re-enable the binary input.

Enabling Binary Output

The NOT command is used to enable binary output. If the card has not been programmed to update on the ready for data (RFD) handshake or the timers, then the command will have no effect. For output mode 1, [x] is the number of bytes that have been preloaded into the buffer (from 0 to the allocated buffer size). Any pending RFD handshakes will be cleared, the buffer starting address will be set to 0, the data queued counter (QO) will be set to the value of [x], and the appropriate update condition will be enabled. If [x] = 0, then the card will wait for data to be loaded (NOX command) before any output will occur.

For output mode 2, the NOT command sets all buffer pointers to 0, resets the QK response and the thread sequence to 0, clears any pending RFD handshakes, and enables the appropriate update condition. [x] is the number of times to repeat the entire thread sequence. The maximum value of [x] is 32767. If [x] is 0, * , or not specified, the sequence will be repeated indefinitely until either the card is reset or the binary output is disabled with an NOD command.

The I option of the command is useful when it is necessary to bootstrap the output to get the handshaking process started (e.g. NOT1I). If issued, the card will output the first sequence regardless of the state of the ready for data handshake. This will cause the data available (DAV) handshake to go active. Once the card has been set up, the NOT command can be issued at any time to restart the binary output.

Command   Description

O

The O (Operational setup) command returns the operational setup parameters for the card.

Syntax: O

The information returned by the O command reflects the programmed setup of the card. The information is returned in the following format. _ indicates an ASCII space character.

```
MODE_[A]_SENSE_[B]_TRI-STATE_[C]_TRI-LEVEL_[D]_
INPUT_[E][F]_OUTPUT_[G][F]_DAV[F]_DAK[F]_OUT-SEQ[]_
[H]_IN-SEQ[]_[H]_IN-FUNC_[I]_IN-MASK_[J]_ INTR_[KLMN]
```

The meaning of each variable is:

[A] = XXXXXX where X = I (Input byte) or O (Output byte). (The rightmost X corresponds to byte 0.)

[B] = XXXXXX where X = H (Logic high true) or L (Logic low true). (The rightmost X corresponds to byte 0.)

[C] = XXXXXX where X = A (Tri-state active) or I (Tri-state inactive). (The rightmost X corresponds to byte 0.)

[D] = XXXXX where X = H (External tri-state active high) or L (External tri-state active low).

[E] = CMD Update the input on command.
DRD Update the input on data ready handshake.

[F] = + or - Active edge of the handshake (meaningless for CMD updates).

[G] = CMD Update the output on command.
RFD Update the output on ready for data handshake.
TIME Update the output on time value

[H] = XXXXXX
where X is an ASCII integer; the programmed input/output load sequence is read from left to right. X = - if the sequence is less than 6 bytes.

[I] = XXXXXX
where X indicates the input data mask function. Valid values for X are: & (AND); # (OR); X (exclusive OR); or - (no mask). The leftmost X corresponds to the first byte in the load sequence.

[J] = XX_XX_XX_XX_XX_XX
where XX is the ASCII Hex mask value. The leftmost XX corresponds to the mask of the first byte in the load sequence.

[KLMN] interrupt status, where
K = D Data ready interrupt enabled
L = R Ready for data interrupt enabled
M = B Binary input buffer full interrupt enabled
N = O Binary output buffer empty interrupt enabled; or - Applicable interrupt not enabled

The power-up default operational response of the card is:

MODE IIIIII SENSE HHHHHH TRI-STATE AAAAAA TRI-LEVEL LLLLLL INPUT CMD+ OUTPUT CMD + DAV+ DAK+ OUT-SEQ[] ------ IN-SEQ[] 012345 IN-FUNC[] ------ IN-MASK[] 00 00 00 00 00 00 INTR ----

Command Description

P

The P (Strobe Pulse Sense) command specifies the active edge of the handshake signals.

Syntax: P {(p)...(e)}

p a single character which specifies the strobe pulse to be defined:

A	Data Available Strobe
R	Ready for Data Strobe
D	Data Ready Strobe
K	Data Acknowledge Strobe
*	All strobes

e specifies the active edge of the specified strobe as follows:

+	positive edge triggered strobe pulse
-	negative edge triggered strobe pulse

Default: P*+ (all pulse senses positive edge triggered)

The bytes can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another P command, or by a Reset or Self Test command. If an edge is not programmed, it will remain in its default (or previously programmed) state. If (e) or (p) is omitted, the command will have no effect. If an invalid parameter is specified, an error will be generated.

This command assumes the U (update) command has specified the use of the handshake signals. If not, this command will have no effect until a U command is issued.

Example:
The following examples show how a sequence of pulse commands will control the pulse trigger active edges:

		Pulse Edges			
Case	Command	A	R	D	K
1	Power up (default)	+	+	+	+
2	PAK-<CR><LF>	-	+	+	-
3	P*-<LF>	-	-	-	-
4	PKD+<LF>	-	-	+	+
5	PAR+DK-<CR><LF>	+	+	-	-

Case 1 is the power up (default) condition, which sets all handshake lines as positive edge triggered.

Case 2 sets the DAV and DAK pulses as negative edge true, leaving RFD and DRD in their previously programmed (default) state.

Case 3 sets all handshake lines negative edge triggered.

Case 4 sets the DAK and DRD strobes as positive edge triggered.

Case 5 sets the DAV and RFD strobes as positive edge triggered, and the DRD and DAK strobes as negative edge triggered.

Command Description

Q The Q (Query Status) command returns the status of various hardware and software states.

Syntax: Q(s)

s one of the following letters, which specifies what is to be returned:

A returns an ASCII error message (see also the N parameter).

B returns the number of bytes queued in the binary input buffer. The response is formatted as a four-character ASCII integer.

D returns the state of the external Data Ready Strobe. 0 indicates the handshake has not occurred, and 1 indicates it has.

I returns the programmed state of the interrupts, and which conditions were active at the time the 53A-412's controller last acknowledged an interrupt from the card. The response is formatted as a two character hexadecimal string. Bit 0 represents interrupt on binary output buffer empty (or breakpoint active), bit 1 is binary input buffer full, bit 2 is RFD, and Bit 3 is DRD. A 1 in any of these bit positions indicates the interrupt is enabled, while a 1 in bit positions 4, 5, 6, and 7 indicate respectively which conditions were active when the interrupt was generated. Bit 7 is the most significant bit of the first hexadecimal character. (See the X command.)

K returns the current thread sequence number and its associated state. The response is formatted as a two-character ASCII integer followed by either 'E' or 'B'. An E indicates the thread is executing, and a B indicates the thread is at a breakpoint.

N returns an ASCII 00 - 99 numeric error code. The codes and their messages are listed below.

O returns the number of bytes queued in the output binary buffer (the number of bytes left to be output). The response is formatted as a four-character ASCII integer.

R returns the state of the external Ready for Data Strobe. 0 indicates the handshake has not occurred, and 1 indicates it has.

T returns the actual current tri-state condition of each output byte (the OR of each byte's external tri-state control line and its tri-state condition as programmed by the T command). The response is formatted as a two character hexadecimal string (00 - 3F). Bits 0 through 5 represent bytes 0 through 5. For example, bit 0 (01) represents byte 0, bit 5 (20) represents byte 5. A 1 in a bit position indicates the corresponding byte is tri-stated.

For the QR, QD, QB, QK, and QO commands, once the Q command has been issued, subsequent input requests will continuously return the respective information until overridden by another Q command, by an I command, or by reset or self test.

If an error is queued while the I command or any Q command other than QA or QN is the active input request mode, the card will respond with ERROR<CR><LF> until either a QA or QN command is issued to acknowledge the error condition. If (s) is not one of the specified characters, the card will respond with 'READY'.

Examples:
The following examples show how each of the above commands will respond on power-up:

Command	Response
read (no command)	READY<CR><LF>
QA<LF>	NO ERRORS<CR><LF>
QB<LF>	0000<CR><LF>
QD<LF>	1<CR><LF>
QI<LF>	00<CR><LF>
QK<LF>	00E<CR><LF>
QN<LF>	00<CR><LF>
QO<LF>	0000<CR><LF>
QR<LF>	1<CR><LF>
QT<LF>	3F<CR><LF>

Error Responses:

Number	Error Message
00	NO ERRORS
01	SELF TEST FAILURE BYTE X COUNT Y where X is an ASCII 0 through 5, indicating the byte failing self test, and Y is an ASCII 00 to FF, indicating the hex bit pattern causing the failure.
02	UNRECOGNIZED COMMAND (X) where X is the first letter of the invalid command.
03	INPUT BUFFER OVERFLOW
04	INVALID COMMAND (X) where X is the first letter of the invalid command.
05	MAXIMUMSEQUENCE LENGTH EXCEEDED - XX where XX is the length of the sequence (up to six sequence numbers are valid).
06	OUTPUT SPECIFIED ON AN INPUT BYTE - X where X is the invalid byte specified.
07	INVALID BIT SPECIFIED
08	INVALID (OR MISSING) HEX VALUE
09	NUMBER OUT OF RANGE
10	INVALID COMMAND DELIMITER
99	UNKNOWN ERROR

Command Description

R

The R (Reset) command resets the board to its power-up state.

Syntax: R

The power-up state of the 53A-412 is:

All I/O pins tri-stated.
All bytes defined as inputs, active high.
All external handshake lines disabled.
Interrupts disabled.

Command Description

S

The S (Self Test) command causes the card to execute a self test, and then return to its power-up state.

Syntax: S

The self test consists of internal circuitry tests, and I/O wraparound tests. The results of a self test can be read using the Query Status commands QA or QN. If the self test fails, error 01 will be generated, and the card's Error LED will be lit.

Command Description

T

The T (Tri-state Control) command specifies under software control whether individual output bytes are tri-stated (high-impedance), or not tri-stated.

Syntax: T {(b)...(a)}...

This command is logically OR'd with the external tri-state control lines ETS0 - ETS5, so if either is active, the byte(s) will be tri-stated. The tri-state command does not imply that bytes are output bytes, or cause bytes to become output bytes.

b byte number, 0 through 5, or * for all bytes

a A or I, Tri-state control active (high impedance) or inactive (not tri-stated).

Default: T*A (all bytes tri-stated, high impedance)

The bytes can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another T command, or by a Reset or Self Test command. If any bytes are not programmed, they will remain in their default state (high impedance) or previously programmed state. If (b) is omitted, the command will have no effect. If (a) is omitted, an error will be generated.

Example:

The following examples show how a sequence of tri-state commands will control the output state of each byte. For this example, it is assumed that all external tri-state inputs (ETS0 - ETS5) are inactive.

Case	Command	Byte Tri-state Control					
		0	1	2	3	4	5
1	Power up (default)	A	A	A	A	A	A
2	T123I<LF>	A	I	I	I	A	A
3	T01A23I45A<LF>	A	A	I	I	A	A
4	T*I<LF>	I	I	I	I	I	I

A = tri-state control active (high impedance)

Case 1 is the power-up (default) condition, which tri-states all bytes (high impedance).

Case 2 sets the tri-state control inactive (non-tri-stated)for bytes 1, 2, and 3, leaving 0, 4, and 5 in their previously programmed state.

Case 3 tri-states bytes 0 and 1, enables bytes 2 and 3 (non-tri-stated), and tri-states bytes 4 and 5.

Case 4 sets the tri-state controls inactive for all bytes (non-tri-stated).

Command Description

U

The U (Update) command specifies whether inputs and outputs are updated immediately on receiving a programming command (I or L command), when external handshake signals (Data Ready or Ready For Data strobes) occur, or on a programmable time count (when an internal timer counts out).

Syntax: U (c)...

c a single letter which specifies the update conditions. Valid entries are:

L Update the output data immediately on command (see the L (load output) command).

R Update the output data with the latest received data when the Ready For Data strobe (RFD) occurs.

I Update the input data immediately on command (see the I (input) command).

D Update the input data when the Data Ready strobe (DRD) occurs.

O[x] Update the output (binary data only; see the N command) every [x] * 200e-9 seconds. [x] is an ASCII integer from 500 to 4,294,967,290.

Default: ULI (update the output on command, update the input on command).

Any or all of the update parameters can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another U command, or by a Reset or Self Test command. If a condition is not programmed, it will remain in its default (or previously programmed) state. If (c) is omitted, the command will have no effect. If an invalid parameter is specified, an error will be generated.

For the L condition, the output data is updated based on the L (Load) command.

For the R condition, the latest data received by the card will be output when an RFD strobe occurs or immediately if an RFD strobe has occurred since the last output command.

Note that when using the RFD/DAV handshakes, the output data may easily be overwritten and lost, since the most recent data received is always output. For example, if two L commands are received before a strobe occurs, the first data will be lost, and the most recent data will be output. To prevent this overwriting of data from occurring, use the Data Available (DAV) handshake and the QR command to read the state of Ready For Data (RFD). Each time the output data is updated, DAV is strobed to tell the external device that new data is available. The external device should then set RFD when it's ready for another output.

If the data reported back by the QR command is a 0, then the last data output has not yet been accepted by the external device. If a 1 is reported back, then the outputs can be updated with no loss of data (the external device has indicated a Ready For Data state).

If programmed for buffered binary output data, the next byte(s) in the buffer will be output when the RFD strobe occurs.

For the I condition, input data is updated immediately when the I (Input) command is received.

For the D condition, data is strobed in when the DRD strobe occurs (the external device indicates it has data ready). The card will respond with a data acknowledge (DAK) strobe when the input data is read from the card. The external device may use the data acknowledge strobe to update its output data, and indicate that it has new data ready for the card by setting the DRD line.

Note that once a DRD handshake occurs, the card will ignore subsequent DRD handshakes until the data is read by the controller. Use of the DAK handshake by the external device will prevent any DRD handshakes and data from being lost.

For the O condition, internal interval timers are set to control when the data is output based on the programmed count. the counters are enabled based on the N command. Once armed, data is output each time the programmed interval occurs ([x] * 200e-9 s). This command can only be used when transferring binary buffered data. For one to three bytes, the maximum update frequency is 10 KHz, or a count of 500. For four to six bytes, the maximum update frequency is 8 KHz, or a count of 625.

The DRD and RFD LEDs light when the handshake occurs (edge triggered), and do not reflect the active logic state of the handshake. A lit LED indicates that a valid handshake has occurred on the DRD or RFD handshake lines.

The DAK and DAV LEDs reflect the logic state of the signal. A lit LED indicates the handshake signal is at a TTL logic high for DAK and DAV.

When the DRD handshake is programmed, the card will immediately drive DAK active to signal the external device that it is ready for input data.

When the RFD handshake is programmed, the DAV signal will go active when a RFD strobe has occurred, and data is output by the card.

Example:

The following example shows how a sequence of update commands will control the update condition(s):

		Update Conditions:	
Case	Command	Output	Input
1	Power-up (default)	L	I
2	UD<LF>	L	D
3	UR<LF>	R	D
4	UIL<LF>	L	I
5	ULD<CR><LF>	L	D
6	UO500I	time	I
7	UO10000D	time	D

Case 1 is the power-up (default) condition, which updates both outputs and inputs on command.

Case 2 updates the input on the DRD strobe, the output remains in its previously programmed condition to update on command.

Case 3 will update the output on an RFD strobe.

Case 4 will update the inputs and outputs on command.

Case 5 will update the output on command, and the input on a DRD strobe.

Case 6 updates the output every 500 * 200e-9 seconds, and updates the input on command.

Case 7 updates the output every 10000 * 200e-9 seconds and updates the input on the Data Ready handshake.

Command Description

V

The V (Version) command returns the current software revision level of the card.

Syntax: V

The format of the returned data is:

```
53A-412 VX.X
```

where X.X is the current revision level (1.0, for example).

Command	Description
X	The X (interrupt enable or disable control) command is used to enable and disable interrupts to the system controller.

Syntax: XA(c)...
XI(c)...

XA enable interrupt

XI disable interrupt

c specifies one of the following:

R	enable interrupt on RFD handshake
D	enable interrupt on DRD handshake
B	enable interrupt on binary input buffer full (see the N command)
O	enable interrupt on binary output buffer empty (see the N command)
*	enable/disable interrupt on any of the above conditions

Default: XI (interrupt disabled)

The data can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another X command or by a Reset or Self Test command. If an interrupt is not specified, it will be disabled. If (c) is invalid, an error will be generated.

The card interrupt is cleared when the @XS command is issued (see Appendix A).

Determining Interrupt Status With the Q Command

The QI command can be used to determine the cause of an interrupt. Since multiple interrupts can be enabled, the QI command response serves the dual purpose of indicating which interrupts are enabled (least significant four bits of the encoded response) and which are active (most significant four bits of the encoded response). A 1 in a bit position indicates it is active. If an interrupt is not enabled, its corresponding active bit is also disabled. The response is formatted as a two-character hexadecimal string. (See the Q command.)

Bits	Represent
0, 4	Interrupt on binary output buffer empty or on a breakpoint
1, 5	Interrupt on binary input buffer full
2, 6	Interrupt on Ready For Data (RFD) handshake
3, 7	Interrupt on Data Ready (DRD) handshake

Bits 0 (enabled) and 4 (active) represent the interrupt on binary output buffer empty (for mode 1), or when a breakpoint occurs (for mode 2). This causes an interrupt to be generated (and the active bit set) each time the applicable condition occurs. The active bit is cleared if the interrupt is reprogrammed with another X command, if a NOC or NOT command is issued, on reset, or on receipt of the QI command. For example, a QI command response of '11' indicates this interrupt is enabled and active. A '01' response indicates it is enabled, but not active.

Bits 1 (enabled) and 5 (active) represent the binary input buffer full. It is set each time the binary input buffer is filled. The active bit is cleared if the interrupt is reprogrammed with another X command, if an NIT command is issued, on reset, or on receipt of the QI command. A QI command response of '22' indicates this interrupt is enabled and active.

Bits 2 (enabled) and 6 (active) represent the interrupt on Ready For Data (RFD) handshake. It is set each time an RFD handshake occurs. The active bit is cleared if the interrupt is reprogrammed with another X command, when new data is strobed out, on reset, or on receipt of the QI command. A QI command response of '44' indicates this interrupt is enabled and active. When outputting binary buffered data, the RFD interrupt is disabled after the specified number of blocks have been transferred.

Bits 3 (enabled) and 7 (active) represent the interrupt on Data Ready (DRD) handshake. It is set each time a DRD handshake occurs. The active bit is cleared if the interrupt is reprogrammed with another X command, when new data is input by the card, on reset, or on receipt of the QI command. A QI command response of '88' indicates this interrupt is enabled and active. When outputting binary buffered data, the DRD interrupt is disabled after the specified number of blocks have been transferred.

Examples:

XAR<CR><LF>	interrupts when the RFD handshake occurs
XARDO<LF>	interrupts when any of the three indicated conditions occur
XI*<CR><LF>	disables all interrupts

Command Description

Z

The Z [Tri-state (high impedance) level] command specifies the active level of the external tri-state control lines ETS0 - ETS5.

Syntax: Z {(b)...(l)}

b byte number, 0 through 5, or * for all bytes

l H or L, Tri-state line active high (TTL logic 1) or low (TTL logic 0) respectively, where tri-state active is the state that puts the output lines in a high impedance state.

Default: Z*L (all bytes, external tri-state active low)

The bytes can be programmed in any order, and once programmed, the setup remains valid unless specifically overridden by another Z command, or by a Reset or Self Test command. If any bytes are not programmed, they will remain in their default (or previously programmed) state. Note that all external tri-state lines have 22k pull-ups on them, so the external tri-states (by default) are not active if left unconnected. If (b) is omitted, the command will have no effect. If (l) is omitted or an invalid parameter is specified, an error will be generated.

NOTE: The external Tri-state lines are logically OR'd with the Tri-state Control command (T), so if either is active, the byte(s) will be tri-stated.

Example:
The following example shows how a sequence of tri-state level commands will control the external tri-state active levels of each byte:

		Individual Byte					
Case	Command	Tri-state Active Levels					
		0	1	2	3	4	5
1	Power-up (default)	L	L	L	L	L	L
2	Z123H<CR><LF>	L	H	H	H	L	L
3	Z01H23L45H<LF>	H	H	L	L	H	H
4	Z*H<LF>	H	H	H	H	H	H

Case 1 is the power-up (default) condition, which sets all external tri-state level inputs to active low.

Case 2 sets external tri-states for bytes 1, 2, and 3 as active high, leaving 0, 4, and 5 in their previously programmed state.

Case 3 sets 0 and 1 high, 2 and 3 low, and 4 and 5 high.

Case 4 sets all external tri-states for all bytes active high.

INSTALLATION

The 53A-412 Card is a function card; therefore, it may be installed in any blue card slot. Setting the Address Select switch defines the card's programming address. To avoid confusion, it is recommended that the slot number and the programming address be the same.

CAUTION:

To avoid plugging the card in backwards, observe the following:

a. Match the keyed slot on the card to the key in the backplane connector. The component side should be to the right for a 53 Series Chassis and to the top for a 63 Series Chassis.

b. There are two ejectors on the card. Make sure the ejector marked '53A-412' is at the top for a 53 Series Chassis and to the left for a 63 Series Chassis.

CAUTION:

The 53A-412 Card is a piece of electronic equipment and therefore has some susceptibility to electrostatic damage (ESD). ESD precautions must be taken whenever the card is handled.

APPENDIX A

53/63 SERIES SYSTEM COMMANDS

Command Description

@XY The @XY (Address) command addresses a function card in the 53/63 Series System.

@ is a delimiter used by the 53/63 Series System.

X is a card cage address (0-9) defined by the Address Select switch on the 53A-171 Control Card in the addressed card cage.

Y is a function-card address (0-9) defined by the Address Select switch on the function card. Once a card cage/function-card combination is addressed, it remains addressed until the 53/63 Series System detects a new @ character.

@XS The @XS (Status) command provides the interrupt status of all function cards within the card cage defined by X. The interrupt status of all function cards in the addressed card cage is latched into the 53A-171 Control Card when the @XS command is issued. All function cards in all card cages become unaddressed after the @XS command is issued. The @XS command allows the interrupt status of the 53A-412 Card to be read as programmed by the Interrupt command (see the Card Commands subsection in the Operation section of this manual for details of the command). The 53A-171 Control Card Operating Manual describes the @XS command in detail.

@XH The @XH (Halt) command halts all function cards within the card cage defined by X. This command does not affect function cards in other card cages. How a function card reacts to the @XH command depends on the particular card. On the 53A-412 Card the position of the Halt switch causes the @XH command to have the following effects: If the Halt switch is ON, the 53A-412 Card is reset to its power-up state. In the OFF position, the @XH command will have no effect. In all cases, an addressed function card (Power LED out) becomes unaddressed (Power LED lit).

STOP The STOP command is not a string of ASCII characters. This command is hard-wired from the system controller to the 53/63 System's communications card in each card cage. When the system controller issues a STOP command, each function card (including the 53A-412 Card) reacts as if it had received the @XH command described above.

How the system controller executes a STOP command depends on the communications card used. For example, when using the 53A-128 IEEE-488 Communications Card, a STOP command is executed whenever the system controller asserts the IEEE-488 bus line IFC (Interface Clear) true.

APPENDIX B

INPUT/OUTPUT CONNECTIONS

Pinouts

1	Ready for Data input
2	Data Ready input
3	Data Available output
4	Data Acknowledge output
5	ground
6	ground
7	byte 0 bit 7 (MSB)
8	byte 0 bit 6
9	byte 0 bit 5
10	byte 0 bit 4
11	byte 0 bit 3
12	byte 0 bit 2
13	byte 0 bit 1
14	byte 0 bit 0 (LSB)
15	External Tri-State input for byte 0
16	ground
17	ground
18	byte 1 bit 7 (MSB)
19	byte 1 bit 6
20	byte 1 bit 5
21	byte 1 bit 4
22	byte 1 bit 3
23	byte 1 bit 2
24	byte 1 bit 1
25	byte 1 bit 0 (LSB)
26	External Tri-state input for byte 1
27	ground
28	ground
29	byte 2 bit 7 (MSB)
30	byte 2 bit 6
31	byte 2 bit 5
32	byte 2 bit 4
33	byte 2 bit 3
34	byte 2 bit 2
35	byte 2 bit 1
36	byte 2 bit 0 (LSB)

37	External Tri-State input for byte 2
38	ground
39	ground
40	byte 3 bit 7 (MSB)
41	byte 3 bit 6
42	byte 3 bit 5
43	byte 3 bit 4
44	byte 3 bit 3
45	byte 3 bit 2
46	byte 3 bit 1
47	byte 3 bit 0 (LSB)
48	External Tri-State input for byte 3
49	ground
50	ground
51	byte 4 bit 7 (MSB)
52	byte 4 bit 6
53	byte 4 bit 5
54	byte 4 bit 4
55	byte 4 bit 3
56	byte 4 bit 2
57	byte 4 bit 1
58	byte 4 bit 0 (LSB)
59	External Tri-State input for byte 4
60	ground
61	ground
62	byte 5 bit 7 (MSB)
63	byte 5 bit 6
64	byte 5 bit 5
65	byte 5 bit 4
66	byte 5 bit 3
67	byte 5 bit 2
68	byte 5 bit 1
69	byte 5 bit 0 (LSB)
70	External Tri-State input for byte 5
71	ground
72	ground
73-76	not used

APPENDIX C

SAMPLE BASIC PROGRAM FOR THE 53A-412

This section contains example programs which demonstrate how the various programmable features of the 53A-412 are used. The examples are written in BASIC using an IBM PC or equivalent computer as the system controller.

Definition of BASIC Commands

The programming examples in this manual are written in Microsoft GW BASIC. These examples use the GW BASIC commands described below. If the programming language you are using does not conform exactly to these definitions, use the command in that language that will give the same result.

Command Result

CALL ENTER (R$, LENGTH%, ADDRESS%, STATUS%)
The CALL ENTER statement inputs data into the string R$ from the IEEE-488 instrument whose decimal primary address is contained in the variable ADDRESS%. Following the input, the variable LENGTH% contains the number of bytes read from the instrument. The variable STATUS% contains the number 0 if the transfer was successful or an 8 if an operating system timeout occurred in the PC. Prior to using the CALL ENTER statement, the string R$ must be set to a string of spaces whose length is greater than or equal to the maximum number of bytes expected from the 53A-412.

CALL SEND (ADDRESS%, WRT$, STATUS%)
The CALL SEND statement outputs the contents of the string variable WRT$ to the IEEE-488 instrument whose decimal primary address is contained in the variable ADDRESS%. Following the output of data, the variable STATUS% contains a 0 if the transfer was successful and an 8 if an operating timeout occurred in the PC.

END Terminates the program.

FOR/NEXT
Repeats the instructions between the FOR and NEXT statements for a defined number of iterations.

GOSUB n
Runs the subroutine beginning with line n. EX: GOSUB 750 - runs the subroutine beginning on line 750. The end of the subroutine is delineated with a RETURN statement. When the subroutine reaches the RETURN statement, execution will resume on the line following the GOSUB command.

GOTO n
Program branches to line n. EX: GOTO 320 - directs execution to continue at line 320.

IF/THEN

Sets up a conditional (IF/THEN) statement. Used with other commands, such as PRINT or GOTO, so that IF the stated condition is met, THEN the command following is effective. EX: IF I = 3 THEN GOTO 450 - will continue operation at line 450 when the value of variable I is 3.

REM or '

All characters following the REM command or a ' are not executed. These are used for documentation and user instructions. EX: REM **CLOSE ISOLATION RELAYS**

RETURN

Ends a subroutine and returns operation to the line after the last executed GOSUB command.

<CR> Carriage return character, decimal 13.

<LF> Line feed character, decimal 10.

Programming Example In BASIC

The following sample BASIC program shows how commands for the 53A-412 might be used. This example assumes that the 53A-412 is installed in a CDS card cage that is controlled via an IEEE-488 interface from an external system controller, such as an IBM PC or equivalent using a Capital Equipment Corp. IEEE-488 interface. This example assumes the following system addresses:

1) PC controller IEEE-488 addresses - 21
2) card cage IEEE-488 address (set on the 53A-128 Card) - 24
3) card cage address (set on the 53A-171 Card) - 0
4) board address - 5

Lines which are indented and not numbered are comments which clarify what the program is doing at those points.

Example 1

Lines 10 through 70 initialize the PC's IEEE-488 interface card.

```
10 DEF SEG = &HC400
     Defines memory location for IBM PC IEEE-488 Interface Card.
20 INIT = 0
     Initialize PROM offsets for IBM PC IEEE-488 Interface Card.
30 SEND = 9 : ENTER = 21
40 PC.ADDRESS% = 21
     Defines PC controller's IEEE-488 address.
50 ADDR412% = 24
     Defines 53A-412's IEEE-488 address.
60 CONTROL% = 0
     Defines I/O card as a bus controller.
70 CALL INIT(PC.ADDRESS%,CONTROL%)
```

75 WRT$ = "@05": CALL SEND (PC.ADDRESS%,WRT$,STATUS%)

Address the 53A-412 Card. The card's power LED should go out, indicating it is addressed.

80 RD$ = SPACE$(100)

Allocate space for the input string variable.

90 TM$ = CHR$(10)

Define the command terminator to be a line feed.

100 CLS

Clear the screen.

110 WRT$ = "R" + TM$

Reset the card.

120 CALL SEND(ADDR412%,WRT$,STATUS%)

Output the reset command.

130 CALL ENTER(RD$,LENGTH%,ADDR412%,STATUS%)

Read the default message. The card should respond with "READY".

140 PRINT "DEFAULT MESSAGE -> "+ RD$

Print the default message.

150 WRT$ = "S;QA" + TM$

Issue a self test command ("S"), and read the result back using the query ("QA") command. A semi-colon is used as the command terminator for the S command, and a line feed as the terminator for the QA command. The response should be "No errors".

160 CALL SEND(ADDR412%,WRT$,STATUS%)

Output the command.

170 CALL ENTER(RD$,LENGTH%,ADDR412%,STATUS%)

Read and print the results. The card should respond with "NO ERRORS".

180 PRINT "SELF TEST RESULT -> " + RD$

190 WRT$ = "M*O;T*I;L*D55" + TM$

Output a "55" hex to all bytes. The commands being issued do the following:

M*O	M*O Defines all bytes as outputs
T*I	T*I Un-tri-state the outputs
L*D55	Load a 55 hex into all outputs

200 CALL SEND(ADDR412%,WRT$,STATUS%)

Output the command. The front panel LEDs should display HEX "55" for all bytes.

210 WRT$ = "I*" + TM$

Read back the data using the Input command.

220 CALL SEND(ADDR412%,WRT$,STATUS%)

Output the command.

230 CALL ENTER(RD$,LENGTH%,ADDR412%,STATUS%)

Read and print the results.

240 PRINT "THE DATA IS -> " + RD$

The card should respond with "555555555555" as the data.

250 WRT$ = "778899AABBCC"

Output a 77, 88, 99, AA, BB, and CC HEX to bytes 0 through 5 respectively.

260 CALL SEND(ADDR412%,WRT$,STATUS%)

Output the command.

270 CALL ENTER(RD$,LENGTH%,ADDR412%,STATUS%)

Read and print the results.

280 PRINT "THE DATA IS -> " + RD$

The card should respond with "778899AABBCC" as the data.

290 END

Example 2

This is a more advanced program for the experienced user which demonstrates the use of the Load and Input commands. Lines 10 through 80 are required to initialize the IEEE-488 Bus Interface Card.

```
10 DEF SEG = &HC400
     Defines memory location for IBM PC IEEE-488 Interface Card.
20 SEND = 9:ENTER = 21:INIT=0
30 PC.ADDRESS% = 21
     Defines I/O card address.
40 ADDR412% = 24
     Defines 53A-412's IEEE-488 address.
50 CONTROL% = 0
     Defines I/O card as a bus controller.
60 TM$ = CHR$(10)
     Define line feed terminator.
70 CALL INIT(PC.ADDRESS%,CONTROL%)
80 IF STATUS% <> 0 THEN PRINT "*** 488 FATAL ERROR ***": STOP
85 WRT$ = "@05": CALL SEND (PC.ADDRESS%,WRT$,STATUS%)
     Address the 53A-412 Card. The card's power LED should go out, indicating it is
     addressed.
90 RD$ = SPACE$(100)
     Allocate space for the input string variable.
100 CLS
     Clear the screen.
110 WRT$ = "R" + TM$
     Reset the card.
120 CALL SEND(ADDR412%,WRT$,STATUS%)
130 CALL ENTER(RD$,LENGTH%,ADDR412%,STATUS%)
140 PRINT ("DEFAULT MESSAGE -> "+ LEFT$(RD$,LENGTH%))
     Reset the card, and then read its default message. The LEFT$ function is used to
     suppress the carriage return and line feed characters from the response data for
     printing to the screen. The default message response is "READY".
150 STEPNUM = 2
160 WRT$ = "M*O;T*I;I*" + TM$
     The commands being issued do the following:
          M*O  M*I Defines all bytes as outputs
          T*I  T*I Un-tri-states all bytes
          I*   Read back all bytes in the order 0-1-2-3-4-5.

     The following lines show how the Load ("L") command functions. They track steps
     2-17 of the example following the "L" command in this manual.
170 GOSUB 1500
180 WRT$ = "L*D55" + TM$
     Prints the command string, the step number, and the data. Output a 55 hex to all
     bytes. The expected readback data is "555555555555".
190 GOSUB 1500
200 WRT$ = "001122334455"
     Implicit load data (no command or terminator needed). Output a 00,11,22,33,44, and
     55 hex to bytes 0 through 5 respectively. The expected readback data is
     "001122334455".
```

```
210 GOSUB 1500
220 WRT$ = "L1d01/50dfa/2d20/4d88/3dcc" + TM$
```

Output a 01 hex to byte 1, an FA hex to bytes 5 and 0, a 20 hex to byte 2, an 88 hex to byte 4, and a CC hex to byte 3. This redefines the implicit output sequence to 1-5-0-2-4-3. The expected readback data is "FA0120CC88FA".

```
230 GOSUB 1500
240 WRT$ = "001122334455"
```

Output implicit data per the current sequence. The expected readback data is "220033554411".

```
250 GOSUB 1500
260 WRT$ = "LO1S04" + TM$
```

Use the load override command to set bit 4 of byte 1 high, without affecting the implicit output sequence. The expected readback data is "221033554411".

```
270 GOSUB 1500
280 WRT$ = "L123#80" + TM$
```

Use the OR mask to set bit 7 of bytes 1, 2, and 3. The new load sequence is now defined as 1-2-3. The expected readback data is "2290B3D54411".

```
290 GOSUB 1500
300 WRT$ = "001122"
```

Implicit load data. Output a 00, 11, and 22 hex to bytes 1, 2, and 3 respectively. The expected readback data is "220011224411".

```
310 GOSUB 1500
320 WRT$ = "L150243" + TM$
```

Define a new sequence. Redefine the output sequence to be 1-5-0-2-4-3. The expected readback data is "220011224411".

```
330 GOSUB 1500
340 WRT$ = "001122334455"
```

Implicit load data. Output data per the new sequence. The expected readback data is "220033554411".

```
350 GOSUB 1500
360 WRT$ = "L*D33" + TM$
```

Output a 33 hex to all bytes. Note that this redefines the sequence to be 0-1-2-3-4-5-6-7-8-9. The expected readback data is "333333333333".

```
370 GOSUB 1500
380 WRT$ = "L0s02/1r04/2&22/3X22/4#44" + TM$
```

Set bit 2 of byte 0, reset bit 4 of byte 1, AND a 22 hex to byte 2 XOR a 22 hex to byte 3, and OR a 44 hex to byte 4. The expected readback data is "372322117733".

```
390 GOSUB 1500
400 WRT$ = "0011"
```

Implicit load (not enough data to be output). Output data per the new sequence 0-1-2-3-4. Note that data is not output until the required number of bytes is received. The expected readback data is "001122334433".

```
410 GOSUB 1500
420 WRT$ = "223344"
```

Fill out the required data.

```
430 GOSUB 1500
```

```
440 WRT$ = "Lo41d55" + TM$
```

Use the load override command to set bytes 4 and 1 to a 55 hex. The expected readback data is "005522335533".

```
450 GOSUB 1500
```

```
460 WRT$ = "AABBCCDDEE"
     Implicit load. Output an AA,BB,CC,DD, and EE hex to bytes 0 through 4 respectively.
     The expected readback data is "AABBCCDDEE33".
470 GOSUB 1500
480 PRINT:INPUT "TYPE ENTER TO CONTINUE",DUMMY$
```

The following lines show how the Input ("I") command is used. They track lines 2 through 10 of the example following the "I" command in this manual.

```
490 CLS:STEPNUM = 2
500 WRT$ = "R;M*O;T*I;L*;001122334455" + TM$
     Define all bytes as outputs, un-tri-stated, data = 001122334455.
510 CALL SEND(ADDR412%,WRT$,STATUS%)
520 WRT$ = "I*;" + TM$
     Read back the state of all bytes. The expected data is "001122334455".
530 GOSUB 1500
540 WRT$ = "I123" + TM$
     Input bytes 1, 2, and 3. Read back the states of bytes 1, 2, and 3 in that order. The
     expected readback data is "112233".
550 GOSUB 1500
560 WRT$ = ""
     Read back the data without issuing a command (implicit read).
570 GOSUB 1500
580 WRT$ = "I*&55" + TM$
     Input all bytes AND'ed with a 55 hex. Read back the data using the AND mask. The
     expected data is "001100114455".
590 GOSUB 1500
600 WRT$ = ""
     Do another implicit read. Note that the data is still reported back with the mask
     overlaid.
610 GOSUB 1500
620 WRT$ = "I*" + TM$
     Input all bytes. Read back all bytes without any masks. The expected data is
     "001122334455".
630 GOSUB 1500
640 WRT$ = "IO3X11" + TM$
     Use the input override command to read byte 3 XOR'd with an 11 hex, without
     affecting the input sequence. The expected data is "22".
650 GOSUB 1500
660 WRT$ = ""
     Read back the data. Note the preceding override command now has no effect. The
     expected data is "001122334455".
670 GOSUB 1500
680 WRT$ = "I543012" + TM$
     Change the input sequence. Redefine the read sequence to be 5-4-3-0-1-2. The
     expected data is "554433001122".
690 GOSUB 1500
700 WRT$ = "I0#55/1XAA/2345" + TM$
     Read byte 0 OR'ed with a 55 hex, byte 1 XOR'd with an AA hex, and bytes 2,3,4, and
     5. The expected readback data is "55BB22334455".
710 GOSUB 1500
720 PRINT:INPUT "TYPE ENTER TO RETURN TO THE SYSTEM",DUMMY$
```

730 SYSTEM

The following subroutine outputs the contents of the string WRT$ to the 53A-412 and then inputs data from the card into the string RD$. Both the input and output data are printed on the PC's display.

```
1500  LOCATE STEPNUM,1
1510  PRINT "cmd = " + WRT$
1520  CALL SEND(ADDR412%,WRT$,STATUS%)
1530  CALL ENTER(RD$,LENGTH%,ADDR412%,STATUS%)
1540  LOCATE STEPNUM,40
1550  PRINT "step " + STR$(STEPNUM) + ": DATA = " + LEFT$(RD$,LENGTH%)
1560  STEPNUM = STEPNUM + 1
1570  RETURN
```

APPENDIX D

OPTIONS

Option 01

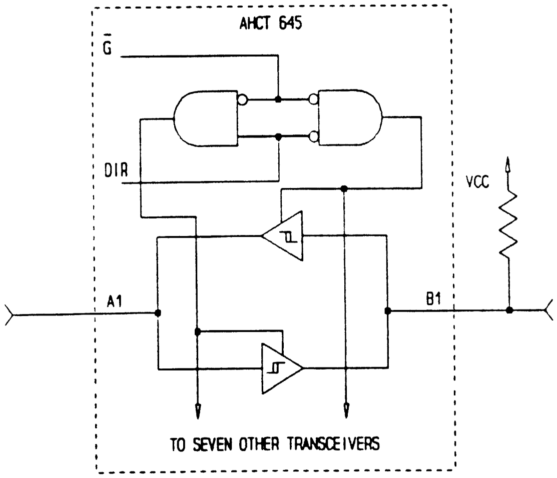
This option provides isolation of the I/O section of the card from system ground, using optical isolators and an on-card isolated power supply. See the Specifications section for the isolation voltage and resistance. There are no changes in operation.

Option 02

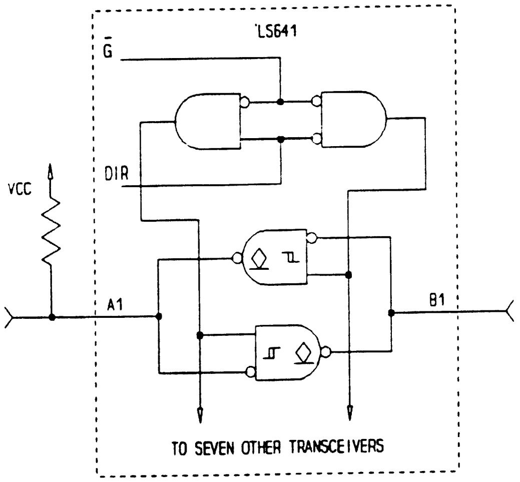
This option changes the output drivers to open collector drivers and removes the pullups from the output. There are no changes in operation.

Specification Changes:

Output high voltage (V_{OH})	(customer supplied) 5.5 maximum
Output low voltage (V_{OL})	0.5 V maximum @ I_{OL} = 24 mA
Output low current (I_{OL})	24 mA maximum
Input high voltage (V_{IH})	2 V minimum
Input low voltage (V_{IL})	0.6 V maximum



Standard Board Output Driver



Option 02 Installed Output Driver