SOFTWARE REFERENCE MANUAL

MACRO STATION

UMAC MACRO & MACRO STACK

3Ax-602804-xSxx

January 29, 2003

DELTA TAU
Data Systems, Inc.
NEW IDEAS IN MOTION ...

Single Source Machine Control *Power // Flexibility // Ease of Use*
21314 Lassen Street Chatsworth, CA 91311 // Tel. (818) 998-2095 Fax. (818) 998-7807 // www.deltatau.com

Copyright Information

To report errors or inconsistencies, call or email:

Delta Tau Data Systems, Inc. Technical Support

Phone: (818) 717-5656

Fax: (818) 998-7807

Email: support@deltatau.com

Website: http://www.deltatau.com

Operating Conditions

All Delta Tau Data Systems, Inc. motion controller products, accessories, and amplifiers contain static sensitive components that can be damaged by incorrect handling. When installing or handling Delta Tau Data Systems, Inc. products, avoid contact with highly insulated materials. Only qualified personnel should be allowed to handle this equipment.

In the case of industrial applications, we expect our products to be protected from hazardous or conductive materials and/or environments that could cause harm to the controller by damaging components or causing electrical shorts. When our products are used in an industrial environment, install them into an industrial electrical cabinet or industrial PC to protect them from excessive or corrosive moisture, abnormal ambient temperatures, and conductive materials. If Delta Tau Data Systems, Inc. products are exposed to hazardous or conductive materials and/or environments, we cannot guarantee their operation.

Table of Contents

MACRO STATION MI-VARIABLE REFERENCE

The MACRO Station, whether in "MACRO Stack" or "UMAC MACRO" configuration, is set up through its own set of initialization I-variables, which are distinct from the I-variables on PMAC. They are usually referenced as "MI"-variables (e.g. MI900) to distinguish them from the PMAC's own I-variables, although they can be referenced just as "I"-variables.
These MI-variables can be accessed from the PMAC(1) or PMAC2 through the on-line **MS{node#},MI{variable#}** read and **MS{node#},MI{variable#}={constant}** write commands, or the **MSR{node#},MI{variable#},{PMAC variable}** read-copy and **MSW{node#},MI{variable#},{PMAC variable}** write-copy commands (either on-line or background PLC), where **{node#}** specifies the MACRO node number (0 to 15), **{variable#}** specifies the number of the Station MI-variable (0 - 1023), {constant} represents the numerical value to be written to the Station MI-variable, or **{PMAC variable}** specifies the value to be copied to or from the Station MI-variable.
For most Station MI-variables, the **{node#}** specifier can take the number of any active node on the station (usually the lowest-numbered active node). These variables have **MS{anynode}** in the header of their descriptions below.
However, there are several node-specific MI-variables. These variables are in the range MI910 to MI939. For these variables, the node specifier must contain the specific node number for the MACRO node they affect. These variables have **MS{node}** in the header of their descriptions below.

Global MI-Variables

MS{anynode},MI0 Station Firmware Version (Read Only)

Range: 0.000 - 9.999
Units: Revision numbers

This variable, when queried, reports the version of the firmware on the MACRO Station.

Note:

It is possible to write a value to this variable, but this should not be done.

Example:
```
MS0,MI0
1.106
```

MS{anynode},MI1 Station Firmware Date (Read Only)

Range: 01/01/00 – 12/31/99
Units: MM/DD/YY

This variable, when queried, reports the date of implementation of the firmware on the MACRO Station. The date is reported in the North American style of month/day/year with two decimal digits for each.

Note:

It is possible to write a value to this variable, but this should not be done.

Since the year is reported with only two digits, it "rolls over" at the turn of a century. If user software makes any date comparisons based on this year value, care must be taken to avoid a "Y2K" error. The earliest firmware date for the MACRO Station is in year 1997. The PMAC command MSDATE, which polls this value, turns the year into a 4-digit value before reporting the value to the host computer.

MS{anynode},MI2 Station ID and User Configuration Word

Range: $000000 - $FFFFFF
Units: none
Default: 0

This variable permits the user to write a station identification number to the MACRO Station. Typically, when the software setup of a Station is complete, a unique value is written to this MI-variable in the station, and saved with the other MI-variables. On power-up/reset, the controller can query MI2 as a quick test to see if the Station has been set up properly for the application. If it does not report back the expected value, the controller can download and save the setup values.

MS{anynode},MI3 Station Rotary Switch Setting (Read Only)

Range: $00 - $FF
Units: none

This variable, when queried, reports the setting of the two rotary hex switches on the MACRO Station. The first hex digit reports the setting of SW1; the second reports the setting of SW2.

Note:

It is possible to write a value to this variable, but this should not be done.

MS{anynode},MI4 Station Status Word (Read Only)

Range: $000000 - $FFFFFF

Units: Bits

This variable, when queried, reports the value of the current status word bits for the MACRO Station. The value reported should be broken into bits. Each bit reports the presence or absence of a particular fault on the Station. If the bit is 0, the fault has not occurred since Station faults were last cleared. If the bit is 1, the fault has occurred since Station faults were last cleared.

Bit #	Fault	Notes

0	CPU Fault	(not used)
1	Ring Communications Error	(temporary)
2	Ring Break Detected	
3	Station Fault	(shutdown)
4	Ring Fault	(“permanent”)
5	Encoder-Loss Fault	(OR of bits 8-11)
6	Amplifier Fault	
7	Ring Break Notification Received	
8	Encoder 1, 5, or 9 Loss Fault	ACC-1E and 2E stack boards only
9	Encoder 2, 6, or 10 Loss Fault	ACC-1E and 2E stack boards only
10	Encoder 3 or 7 Loss Fault	ACC-1E and 2E stack boards only
11	Encoder 4 or 8 Loss Fault	ACC-1E and 2E stack boards only
12	Ring Active	
13	Multiplexer Port Parity Error	From ACC-34 board with parity checking
14	IC Configuration Change	Difference in Servo ICs present since last SAVE command – see bits 19-23.
15-18	(Reserved for future use)	
19	Servo IC 3 Present	UBUS card at $C060
20	Servo IC 2 Present	UBUS card at $C040
21	Servo IC 1 Present	Stack card at $C020
22	Servo IC 0 Present	Stack card at $C000
23	MACRO IC Present	CPU card at $C080, always 1

Any of the fault bits that are set can be cleared with the **MSCLRF{anynode}** (clear fault) command, or the **MS$$$${anynode}** (Station reset) command.

Note:

It is possible to write a value to this variable, but this should not be done.

MS{anynode},MI5 Ring Error Counter (Read Only)

Range: $000000 - $FFFFFF
Units: Error Count

This variable, when queried, reports the number of ring communications errors detected by the MACRO Station since the most recent power-up or reset.

Note:

It is possible to write a value to this variable, but this should not be done.

The ring error counter value should only be cleared to zero using the **MSCLRF{anynode}** or **MS$$${anynode}** commands.

MS{anynode},MI6 Maximum Permitted Ring Error Frequency

Range: $0000000 - $FFFFFFF
Units: Errors per second
Default:

This variable sets the maximum number of ring errors that can be detected by the MACRO Station in a one second period without causing it to shut down for ring failure.

MS{anynode},MI7 Stack-Encoder Loss-Detect Enable

Range: 0 - 1
Units: none
Default: 0

This variable controls whether the MACRO Station will automatically report a node fault upon detection of loss of encoder signal from one of the channels on an ACC-1E or ACC-2E stack board. If MI7 is set to 1, the station will treat the loss of encoder signal as it would treat an amplifier fault, and report a fault for the node back to the controller. The controller at this point should send zero command value(s) to the node, and a disable signal.
In order to enable this function, differential encoders must be used, and the socketed resistor packs for the encoder must be reversed from their factory default setting so that the complementary encoder lines A-, B-, and C- are pulled up to 5V instead of pulled to 2.5V. These SIP packs are installed at the factory so that pin 1 of the pack – marked with a dot – is installed in pin 1 of the socket – marked with a bold white outline and a square solder pin on the board. For this encoder-loss detection to work, the SIP-pack for each encoder must be reversed so that it is at the opposite end of the socket. The following table shows these resistor packs.

Board	Encoder 1	Encoder 2	Encoder 3	Encoder 4
ACC-1E	RP30	RP31	-	-
ACC-2E	RP30	RP31	RP28	RP29

Encoder-loss faults are reported in bits 8 – 11 of MI4. Note that multiple channels may share a single bit of MI4 – a bit is set if any of the channels assigned to hit sees an encoder-loss.
This automatic shut down function is unrelated to the optional per-channel reporting function enabled by MI16 and double-wiring the encoder inputs into the T, U, V, and W flag inputs.

MS{anynode},MI8 MACRO Ring Check Period

Range: 0 - 255
Units: Station phase cycles
Default: 8

MI8 determines the period, in phase cycles, for the MACRO Station to evaluate whether there has been a MACRO ring failure or not. Every phase cycle, the Station checks the ring communications status. In MI8 phase cycles (or MACRO ring cycles), the Station must receive at least MI10 "sync packets" and detect fewer than MI9 ring communications errors, to conclude that the ring is operating correctly. Otherwise, it will conclude that the ring is not operating properly, set its servo command output values to zero, set its amplifier enable outputs to the "disable" state, and force all of its digital outputs to their "shutdown" state as defined by I72-I89, and report a ring fault.
If MI8 is set to 0 at power-on/reset, the MACRO Station will automatically set it to 8.
In Station firmware versions before 1.109, a fixed value of 8 phase cycles was automatically used.

MS{anynode},MI9 MACRO Ring Error Shutdown Count

Range: 0 - 255
Units: none
Default: 4

MI9 determines the number of MACRO communications errors detected that will cause a shutdown fault of the MACRO Station. If the Station detects MI9 or greater MACRO communications errors in MI8 phase (MACRO ring) cycles, it will shut down on a MACRO communications fault, turning off all outputs.
The Station can detect one ring communications error per phase cycle (even if more than one error has occurred). Setting MI9 greater than MI8 means that the Station will never shut down for ring communications error.
The Station can detect four types of communications errors: byte "violation" errors, packet checksum errors, packet overrun errors, and packet underrun errors. If MI9 errors have occurred in the MI8 check period, and at least half of these errors are byte "violation" errors, the Station will conclude that there is a ring break immediately upstream of it (if there are no ring input communications to the Station, there will be continual byte violation errors). In this case, not only will it set its servo command output values to zero, set its amplifier enable outputs to the "disable" state, and force all of its digital outputs to their "shutdown" state as defined by I72-I89, but it will also turn itself into a master so it can report to other devices downstream on the ring.
If MI9 is set to 0 at power-on/reset, the MACRO Station will automatically set it to 4.
In Station firmware versions before 1.109, a fixed value of 2 ring errors was used.

MS{anynode},MI10 MACRO Sync Packet Shutdown Count

Range: 0 – 65,535
Units: none
Default: 4

MI10 determines the number of MACRO ring "sync packets" that must be received during a check period for the Station to consider the ring to be working properly. If the Station detects fewer than MI10 sync packets in MI8 phase (MACRO ring) cycles, it will shut down on a MACRO communications fault, setting its servo command output values to zero, setting its amplifier enable outputs to the "disable" state, and forcing all of its digital outputs to their "shutdown" state as defined by I72-I89.

The node number (0-15) of the sync packet is determined by bits 16-19 of Station variable MI996. On the MACRO Station, this is always node 15 ($F), because this node is always active for MACRO Type 1 auxiliary communications.

The Station checks each phase cycle to see if a sync packet has been received or not. Setting MI10 to 0 means the Station will never shut down for lack of sync packets. Setting MI10 greater than MI8 means that the Station will always shut down for lack of sync packets.

If MI10 is set to 0 at power-on/reset, the MACRO Station will automatically set it to 4.

In Station firmware versions before 1.109, a fixed value of 2 sync packets was used.

MS{anynode},MI11 Station Order Number

Range: 0 – 254
Units: none
Default: 0

MI11 contains the "station-order" number of the MACRO Station on the ring. This permits it to respond to auxiliary MACROSTASCIIn commands from a Turbo PMAC ring controller, regardless of the MACRO Station's rotary-switch settings.

The station ordering scheme permits the ring controller to isolate each master or slave station on the ring in sequence and communicate with it, without knowing in advance how the ring is configured or whether there are any conflicts in the regular addressing scheme. This is very useful for the initial setup and debugging of the ring configuration.

Normally, station order numbers of devices on the ring are assigned in numerical order, with the station downstream of the ring controller getting station-order number 1. This does not have to be the case, however.

"Unordered" stations have the station-order number 0. When the ring controller executes a MACROSTASCII255 command, the first "unordered" station in the ring will respond.

MI11 can also be set with the ASCII command **STN={constant}**. The value of MI11 can also be queried with the ASCII command **STN**.

MS{anynode},MI12 - MI14 (Reserved for future use)

MS{anynode},MI15 Node-14 Auxiliary-Communications Disabled

Range: 0 - 1
Units: none
Default: 0

MI15 controls whether Node 14 can be used in broadcast mode for auxiliary communications with the ring controller, or not. If MI15 is 0, it can be used in this mode, and so is not available for general-purpose I/O use. If MI15 is 1, it cannot be used in this mode, so it is available for general-purpose I/O use.
MI15 is used only at Station power-up/reset, so to change the use of Node 14, you must change the value of MI15, store this change to flash memory with the **MSSAVE** command, and reset the station with the **MS$$$** command.

MS{anynode},MI16 Encoder-Fault Reporting Control

Range: 0 - 1
Units: none
Default: 0

MI16 permits the user to control which type of encoder error is reported back to PMAC in the channel status flag word for each servo interface channel.
If MI16 is set to 0 (default), then the encoder count-error status bit (bit 8 in the channel hardware status word) for each encoder channel is copied into bit 8 of the matching node's status flag word for transmission back to the PMAC. An encoder count error is reported when both A and B encoder signals have a transition in the same SCLK hardware sampling cycle.
If MI16 is set to 1, then the ASIC's own encoder-loss status bit (bit 7 in the channel hardware status word) for each encoder channel is copied into bit 8 of the matching node's status flag word for transmission back to the PMAC. Note that this reporting function is unrelated to the automatic encoder-loss shutdown function using external circuitry that can be enabled with MI7 and reported in MI4.
In order for this encoder-loss detection to work properly, several conditions must apply:

- A "B" version or newer of the DSPGATE1/2 Servo/MACRO IC must be used (true on boards built since Spring 1998).
- Differential encoders must be used.
- The A+, A-, B+, and B- encoder signals must be wired into the T, U, V, and W supplemental flag inputs, respectively, as well as into the regular encoder lines. On the ACC-1E and 2E stack axis boards, this can only be done through the PMAC2-style 100-pin connectors.
- The socketed resistor SIP packs for the encoder channels must be reversed from their factory default configuration. These SIP packs are installed at the factory so that pin 1 of the pack – marked with a dot – is installed in pin 1 of the socket – marked with a bold white outline and a square solder pin on the board. For this encoder-loss to work, the SIP-pack for each encoder must be reversed so that it is at the opposite end of the socket. The SIP packs are:

Board	Encoder 1	Encoder 2	Encoder 3	Encoder 4
ACC-1E	RP30	RP31	-	-
ACC-2E	RP30	RP31	RP28	RP29
ACC-24E2	RP22	RP24	RP22*	RP24*
ACC-24E2A	RP22	RP24	RP22*	RP24*
ACC-24E2S	RP19	RP21	RP27	RP29

*Resistor packs on Option 1 top board of 2-board assembly

- MI16 must be set to 1.

If the T, U, V, and W input flags are used for different purposes, such as Hall commutation sensors, or sub-count information from an analog encoder interpolator, the state of the encoder-loss status bit would appear random and arbitrary.

The state of the encoder-loss hardware status bit for a channel can be polled with MI927 for the node mapped to the channel. If it has been set, it can be cleared by writing a 0 value to MI927.

Note:

As long as the socketed resistor pack for an encoder is reversed from the factory default configuration, the MACRO Station will be able to detect differential encoder loss and shut down on it, even without wiring the encoder signals into T, U, V, and W. However, unless the signals are wired into these flag lines and MI16 is set to 1, the MACRO Station will not be able to notify PMAC exactly which encoder sustained the loss.

MS{anynode},MI17 Amplifier Fault Disable Control

Range: $00 - $FF
Units: none
Default: $00 (amplifier function enabled for all axes)

This variable controls whether the amplifier input to the machine interface channel mapped to each servo node by SW1 is used as one of the conditions that creates a "node fault" to be sent back to the PMAC over the MACRO ring.

The variable consists of 8 bits; each bit controls the disabling of the amplifier fault input for one of the nodes on the Station. A 0 in the bit specifies that the amplifier fault input is to be used ("enabled"); a 1 in the bit specifies that the amplifier fault input is not to be used ("disabled"). The corresponding bit of MI18 determines the polarity of the input if it is enabled.

The following table shows the relationship between the bits of MI17 and the servo nodes on the Station:

MI17 Bit #	7	6	5	4	3	2	1	0
Node #	13	12	9	8	5	4	1	0

I/O Transfer MI-Variables

MS{anynode},MI18 Amplifier Fault Polarity

Range: $00 - $FF
Units: none
Default: $00 (low-true fault for all nodes)

This variable controls how the MACRO Station interprets the polarity of the amplifier fault inputs for each servo node. The variable consists of 8 bits; each bit controls the polarity for one of the servo nodes on the Station. A 0 in a bit specifies a low-true fault (low voltage input means fault); a 1 in a bit specifies a high-true fault (high voltage input means fault). A bit of MI18 is only used if the corresponding bit of MI17 is set to 0, enabling the amplifier fault function for that node.

The following table shows the relationship between the bits of MI18 and the servo nodes on the Station:

MI18 Bit #	7	6	5	4	3	2	1	0
Node #	13	12	9	8	5	4	1	0

MS{anynode},MI19 I/O Data Transfer Period

Range: 0 - 255
Units: Phase Clock Cycles
Default: 0

MI19 controls the data transfer period on a MACRO Station between the MACRO node interface registers and the I/O registers, as specified by station MI-variables MI20 through MI71, and MI169 through MI172. If MI19 is set to 0, this data transfer is disabled. If MI19 is greater than 0, its value sets the period in Phase clock cycles (the same as MACRO communications cycles) at which the transfer is done.

MS{anynode},MI20 Data Transfer Enable Mask

Range: $000000000000 - $FFFFFFFFFFFF
Units: Bits
Default: 0

MI20 controls which of 48 possible data transfer operations are performed at the data transfer period set by MI19. MI20 is a 48-bit value; each bit controls whether the data transfer specified by one of the variables MI21 through MI68 is performed. The relationship of MI20 bits to MI21-MI68 transfers is explained in the following table.

MI20 Bit #	Bit Value	Transfer-Control MI-Variable	MI20 Bit #	Bit Value	Transfer-Control MI-Variable
0	$1	MI21	24	$1000000	MI45
1	$2	MI22	25	$2000000	MI46
2	$4	MI23	26	$4000000	MI47
3	$8	MI24	27	$8000000	MI48
4	$10	MI25	28	$10000000	MI49
5	$20	MI26	29	$20000000	MI50
6	$40	MI27	30	$40000000	MI51
7	$80	MI28	31	$80000000	MI52
8	$100	MI29	32	$100000000	MI53
9	$200	MI30	33	$200000000	MI54
10	$400	MI31	34	$400000000	MI55
11	$800	MI32	35	$800000000	MI56
12	$1000	MI33	36	$1000000000	MI57
13	$2000	MI34	37	$2000000000	MI58
14	$4000	MI35	38	$4000000000	MI59
15	$8000	MI36	39	$8000000000	MI60
16	$10000	MI37	40	$10000000000	MI61
17	$20000	MI38	41	$20000000000	MI62
18	$40000	MI39	42	$40000000000	MI63
19	$80000	MI40	43	$80000000000	MI64
20	$100000	MI41	44	$100000000000	MI65
21	$200000	MI42	45	$200000000000	MI66
22	$400000	MI43	46	$400000000000	MI67
23	$800000	MI44	47	$800000000000	MI68

MS{anynode},MI21-MI68 Data Transfer Source and Destination Address

Range: $000000000000 - $FFFFFFFFFFFF
Units: Double MACRO Station Addresses
Default: 0

These MI-variables each specify a data transfer (copying) operation that will occur on the MACRO Station at a rate specified by Station Variable MI19, and enabled by Station variable MI20.
Each variable specifies the address from which the data will be copied (read), and the address to which the data will be copied (written). These variables are 48-bit values, usually specified as 12 hexadecimal digits.
The first 24 bits (6 hex digits) specify the address of the register on the MACRO Station from which the data is to be copied; the second 24 bits (6 hex digits) specify the address on the MACRO Station to which the data is to be copied. In each set of six hex digits, the last four hex digits specify the actual address. The first 2 digits (8 bits) specify what portion of the address is to be used.
The following diagram shows what each digit represents:

Hex Digit #	1	2	3	4	5	6	7	8	9	10	11	12
Contents	"From" Register Format Code		"From" Register Address				"To" Register Format Code		"To" Register Address			

The following table shows the 2-digit hex format codes and the portions of the address that each one selects.

Code	X or Y	Bit Width	Bit Range	Notes
$40	Y	8	0-7	
$48	Y	8	8-15	
$50	Y	8	16-23	
$54	Y	12	0-11	Lower 12-bit ADC registers
$60	Y	12	12-23	Upper 12-bit ADC registers
$64	Y	16	0-15	
$6C	Y	16	8-23	16-bit MACRO Servo Node Registers
$78	Y	24	0-23	24-bit MACRO Servo Node Registers
$B0	X	8	0-7	
$B8	X	8	8-15	
$C0	X	8	16-23	
$C4	X	12	0-11	
$D0	X	12	12-23	
$D4	X	16	0-15	
$DC	X	16	8-23	16-bit MACRO I/O Node Registers
$E8	X	24	0-23	24-bit MACRO I/O Node Registers

The memory and I/O map at the back of this Software Reference manual provides a detailed list of registers that can be copied using these MI-variables.

Note:

For copying data between digital I/O cards with byte-wide data paths (ACC-3E, 4E, 9E, 10E, 11E, 12E, and 14E) and MACRO nodes, it is generally better to use MI69 – MI71, and MI169 – MI172.

Example:
```
MI21=$780200E8C0A0
```
copies 24-bit data from Station address Y:$0200 to X:$C0A0

MS{anynode},MI69, MI70 I/O-Board 16-Bit Transfer Control

Range: $000000000000 - $FFFFFFFFFFFF
Units: Extended addresses
Default: 0

MI69 and MI70 specify the registers used in 16-bit I/O transfers between MACRO node interface registers and I/O registers on the ACC-3E, 4E, 9E, 10E, 11E, 12E and 14E I/O boards on a MACRO station. They are only used if MI19 is greater than 0.
MI69 and MI70 are 48-bit variables represented as 12 hexadecimal digits. The first 6 digits specify the number and address of 48-bit (3 x 16) real-time MACRO-node register sets to be used. The second 6 digits specify the number and address of 16-bit I/O sets on an ACC-3E or ACC-4E board to be used. The individual digits are specified as follows:

Digit #	Possible Values	Description
1:	0, 1, 2, 3	Number of MACRO I/O nodes to use (0 disables); this should also match the number of 48-bit I/O sets you intend to use (see Digit 7)
2:	0	(Reserved for future use)
3-6	$C0A1 (Node 2), $C0A5 (Node 3), $C0A9 (Node 6), $C0AD (Node 7), $C0B1 (Node 10), $C0B5 (Node 11)	MACRO Station X Address of MACRO I/O node first of three 16-bit registers
7	0, 1, 2, 3	Number of 16-bit I/O sets to use (1x16, 2x16, 3x16; 0 disables)
8	0	(Reserved for future use)
9-12	$FFC0, $FFC8, $FFD0, $FFD8 $FFE0, $FFE8, $FFF0, $FFF8	MACRO Station Y Base Address of I/O Board as set by Board Jumper E1-E4 (ACC-3E board) or E15-E18 (ACC-4E board) MACRO Station Y Base Address of ACC-9E, 10E, 11E, 12E, or 14E UMAC I/O board as set by jumpers/switches on board

When this function is active, the MACRO Station will copy values from the MACRO command (input) node registers to the I/O board addresses; it will copy values from the I/O board addresses to the MACRO feedback (output) node registers. Writing a ‘0’ to a bit of the I/O board enables it as an input, letting the output pull high. Writing a ‘1’ to a bit of the I/O board enables it as an output and pulls the output low.

The following table shows the mapping of I/O points on the I/O piggyback boards to the MACRO node registers.

I/O Point #s	ACC-3E Part	Present on ACC-4E?	Matching MACRO X Register
I/O00 - I/O15	Option A	Yes	Specified MACRO X Address + 0
I/O16 - I/O31	Option A	Yes	Specified MACRO X Address + 1
I/O32 - I/O47	Option A	Yes	Specified MACRO X Address + 2
I/O48 - I/O63	Option B	No	Specified MACRO X Address + 4
I/O64 - I/O79	Option B	No	Specified MACRO X Address + 5
I/O80 - I/O95	Option B	No	Specified MACRO X Address + 6
I/O96 - I/O111	Option C	No	Specified MACRO X Address + 8
I/O112 - I/O127	Option C	No	Specified MACRO X Address + 9
I/O128 - I/O143	Option C	No	Specified MACRO X Address + 10

The following table shows the mapping of I/O points on the I/O backplane boards to the MACRO node registers:

Board # at Set Address	E6x Rows Connected	Byte on Data Bus	I/O Point #s on Board	Matching MACRO X Register
1st	1 & 2	Low	0 –15	Specified MACRO X Address + 0
1st	1 & 2	Low	16 – 31	Specified MACRO X Address + 1
1st	1 & 2	Low	32 - 47	Specified MACRO X Address + 2
2nd	2 & 3*	Middle	0 –15	Specified MACRO X Address + 4
2nd	2 & 3*	Middle	16 – 31	Specified MACRO X Address + 5
2nd	2 & 3*	Middle	32 - 47	Specified MACRO X Address + 6
3rd	4 & 5	High	0 –15	Specified MACRO X Address + 8
3rd	4 & 5	High	16 – 31	Specified MACRO X Address + 9
3rd	4 & 5	High	32 - 47	Specified MACRO X Address + 10

* Rows 3 & 4 connected creates same setting

Note

The ACC-14E backplane I/O board can only be set up for the low byte on the data bus.

Examples:

MI69=$30C0A130FFC0 transfers 3 sets of 48-bit I/O between an I/O board set at $FFC0 and MACRO Nodes 2 ($C0A1-$C0A3), 3 ($C0A5-$C0A7), and 6 ($C0A9-$C0AB).

MI70=$10C0B130FFC8 transfers 1 set of 48-bit I/O between an I/O board set at $FFC8 and MACRO Node 10 ($C0B1-$C0B3).

MS{anynode},MI71 I/O-Board 24-Bit Transfer Control

Range: $000000000000 - $FFFFFFFFFFFF
Units: Extended addresses
Default: 0

MI71 specifies the registers used in 24-bit I/O transfers between MACRO I/O node interface registers and I/O registers on the ACC-3E, 4E, 9E, 10E, 11E, 12E, and 14E I/O boards on a MACRO station. It is only used if MI19 is greater than 0.

MI71 is a 48-bit variable represented as 12 hexadecimal digits. The first 6 digits specify the number and address of 48-bit real-time MACRO-node register sets to be used. The second 6 digits specify the number and address of 48-bit I/O sets on an ACC-3E or ACC-4E board to be used. The individual digits are specified as follows:

Digit #	Possible Values	Description
1:	0, 1, 2, 3	Number of MACRO I/O nodes to use times 2 (0 disables); this should also match the number of 48-bit I/O sets you intend to use (see Digit 7)
2:	0	(Reserved for future use)
3-6	$C0A0 (Node 2), $C0A4 (Node 3), $C0A8 (Node 6), $C0AC (Node 7), $C0B0 (Node 10), $C0B4 (Node 11)	MACRO Station X Address of MACRO I/O node first of three 16-bit registers
7	0, 1, 2	Number of 24-bit I/O sets to use (1x24, 2x24; 0 disables)
8	0	(Reserved for future use)
9-12	$FFC0, $FFC8, $FFD0, $FFD8 $FFE0, $FFE8, $FFF0, $FFF8	MACRO Station Y Base Address of I/O Board as set by Board Jumper E1-E4 (ACC-3E board) or E15-E18 (ACC-4E board) MACRO Station Y Base Address of ACC-9E, 10E, 11E, 12E, or 14E UMAC I/O board as set by jumpers/switches on board

When this function is active, the MACRO Station will copy values from the MACRO command (input) node registers to the I/O board addresses; it will copy values from the I/O board addresses to the MACRO feedback (output) node registers. Writing a ‘0’ to a bit of the I/O board enables it as an input, letting the output pull high. Writing a ‘1’ to a bit of the I/O board enables it as an output and pulls the output low.

The following table shows the mapping of I/O points on the I/O piggyback boards to the MACRO node registers.

I/O Point #s	ACC-3E Part	Present on ACC-4E?	Matching MACRO Node X Register
I/O00 - I/O23	Option A	Yes	Specified MACRO Node X Address + 0
I/O24 - I/O47	Option A	Yes	Specified MACRO Node X Address + 4
I/O48 - I/O71	Option B	No	Specified MACRO Node X Address + 8
I/O72 - I/O95	Option B	No	Specified MACRO Node X Address + 12
I/O96 - I/O119	Option C	No	Specified MACRO Node X Address + 16
I/O120 - I/O143	Option C	No	Specified MACRO Node X Address + 20

The following table shows the mapping of I/O points on the I/O backplane boards to the MACRO node registers:

Board # at Set Address	E6x Rows Connected	Byte on Data Bus	I/O Point #s on Board	Matching MACRO X Register
1st	1 & 2	Low	0 –23	Specified MACRO X Address + 0
1st	1 & 2	Low	24 – 47	Specified MACRO X Address + 4
2nd	2 & 3*	Middle	0 –23	Specified MACRO X Address + 8
2nd	2 & 3*	Middle	24 – 47	Specified MACRO X Address + 12
3rd	4 & 5	High	0 –23	Specified MACRO X Address + 16
3rd	4 & 5	High	24 – 47	Specified MACRO X Address + 20

* Rows 3 & 4 connected creates same setting

Note

The ACC-14E backplane I/O board can only be set up for the low byte on the data bus.

MS{anynode},MI72-MI89 Output Power-On/Shutdown State

Range: $000000 - $FFFFFF

Units: Individual bit values

Default: $000000

MI72 through MI89 are used to determine the states of the digital outputs for MACRO Station I/O boards at power-on and on controlled station shutdown due to a ring error condition.

Each of these MI-variables is a 24-bit value controlling 24 consecutively numbered I/O points on a MACRO I/O board. Each bit controls one I/O point. The least significant bit of the MI-variable controls the lowest-numbered I/O point; the most significant bit controls the highest-numbered I/O point.

A value of 0 in a bit specifies that the corresponding output is to be turned off at power-on or shutdown; a value of 1 in a bit specifies that the corresponding output is to be turned on at power-on or shutdown. If an I/O point has been set up as an input, the value of the bit is not important.

The following table shows which I/O points are controlled by each of these MI-variables

Variable	Board Addressed by Variable:	I/O Points Controlled	ACC-3E Option required	Present on ACC-4E?

MI72	MI69	I/O00 – I/O23	Option A	Yes
MI73	MI69	I/O24 – I/O47	Option A	Yes
MI74	MI69	I/O48 – I/O71	Option B	No
MI75	MI69	I/O72 – I/O95	Option B	No
MI76	MI69	I/O96 – I/O119	Option C	No
MI77	MI69	I/O120 – I/O143	Option C	No
MI78	MI70	I/O00 – I/O23	Option A	Yes
MI79	MI70	I/O24 – I/O47	Option A	Yes
MI80	MI70	I/O48 – I/O71	Option B	No
MI81	MI70	I/O72 – I/O95	Option B	No
MI82	MI70	I/O96 – I/O119	Option C	No
MI83	MI70	I/O120 – I/O143	Option C	No
MI84	MI71	I/O00 – I/O23	Option A	Yes
MI85	MI71	I/O24 – I/O47	Option A	Yes
MI86	MI71	I/O48 – I/O71	Option B	No
MI87	MI71	I/O72 – I/O95	Option B	No
MI88	MI71	I/O96 – I/O119	Option C	No
MI89	MI71	I/O120 – I/O143	Option C	No

MS{anynode},MI90 Multiplexer Port #1 Read Address

Range: $00 - $FF
Units: MACRO Station Multiplexer Port Addresses
Default: $00

MI90 specifies the address on the MACRO Station's JTHW multiplexer port of a 32-bit input port on an ACC-34 family I/O board. If MI90 is set greater than 0, the 32 input values will be copied periodically into Station variable MI91.

MI90 must match the multiplexer port address of the ACC-34 board from which the inputs are to be read, as set by the SW1 DIP-switch bank on the board. For the regular input port, the value of MI90 should be set 1 greater than the base address of the board set by the DIP switch bank. For the ACC-34C's optional second 32-bit input port, the value of MI90 should be set 3 greater than the base address.

The addressing number for the ACC-34 port in MI90 is the same as the number in the TWS M-variable definition on PMAC to access the port directly from the PMAC.

Example:

The DIP-switch bank on an ACC-34 board sets a multiplexer port base address of 8. MI90 should be set to 9 to read the 32-bit input port of the ACC-34.

MS{anynode},MI91 Multiplexer Port #1 Read Value

Range: $00000000 - $FFFFFFFF
Units: Individual bits

MI91 contains the 32-bit value read from the input port of the ACC-34 whose multiplexer port address is specified by MI90. Each bit represents one input from the port. Bit n of MI91 represents Input n on the port.

MS{anynode},MI92 Multiplexer Port #1 Write Address

Range: $00 - $FF
Units: MACRO Station Multiplexer Port Addresses
Default: $00

MI92 specifies the address on the MACRO Station's JTHW multiplexer port of a 32-bit output port on an ACC-34 family I/O board. If MI92 is set greater than 0, the 32 output values will be copied periodically from Station variable MI93.

MI92 must match the multiplexer port address of the ACC-34 board from which the inputs are to be read, as set by the SW1 DIP-switch bank on the board. The value of MI92 should be set 6 greater than the base address of the board set by the DIP switch bank.

The addressing number for the ACC-34 port in MI92 is the same as the number in the TWS M-variable definition on PMAC to access the port directly from the PMAC.

Example:

The DIP-switch bank on an ACC-34 board sets a multiplexer port base address of $10 (16). MI92 should be set to $16 (22) to write to the 32-bit output port of the ACC-34.

MS{anynode},MI93 Multiplexer Port #1 Write Value

Range: $00000000 - $FFFFFFFF
Units: Individual bits

MI93 contains the 32-bit value written to the output port of the ACC-34 whose multiplexer port address is specified by MI92. Each bit represents one output on the port. Bit n of MI93 represents Output n on the port.

MS{anynode},MI94 Multiplexer Port #2 Read Address

Range: $00 - $FF
Units: MACRO Station Multiplexer Port Addresses
Default: $00

MI94 specifies the address on the MACRO Station's JTHW multiplexer port of a 32-bit input port on an ACC-34 family I/O board. If MI94 is set greater than 0, the 32 input values will be copied periodically into Station variable MI95.

MI94 must match the multiplexer port address of the ACC-34 board from which the inputs are to be read, as set by the SW1 DIP switch bank on the board. For the regular input port, the value of MI94 should be set 1 greater than the base address of the board set by the DIP switch bank. For the ACC-34C's optional second 32-bit input port, the value of MI94 should be set 3 greater than the base address.

The addressing number for the ACC-34 port in MI94 is the same as the number in the TWS M-variable definition on PMAC to access the port directly from the PMAC.

Example:

The DIP switch bank on an ACC-34 board sets a multiplexer port base address of $18 (24). MI90 should be set to $19 (25) to read the 32-bit input port of the ACC-34.

MS{anynode},MI95 Multiplexer Port #2 Read Value

Range: $00000000 - $FFFFFFFF
Units: Individual bits

MI95 contains the 32-bit value read from the input port of the ACC-34 whose multiplexer port address is specified by MI94. Each bit represents one input from the port. Bit n of MI95 represents Input n on the port.

MS{anynode},MI96 Multiplexer Port #2 Write Address

Range: $00 - $FF
Units: MACRO Station Multiplexer Port Addresses
Default: $00

MI96 specifies the address on the MACRO Station's JTHW multiplexer port of a 32-bit output port on an ACC-34 family I/O board. If MI96 is set greater than 0, the 32 output values will be copied periodically from Station variable MI97.

MI96 must match the multiplexer port address of the ACC-34 board from which the inputs are to be read, as set by the SW1 DIP-switch bank on the board. The value of MI96 should be set 6 greater than the base address of the board set by the DIP switch bank.

The addressing number for the ACC-34 port in MI96 is the same as the number in the TWS M-variable definition on PMAC to access the port directly from the PMAC.

Example:

The DIP-switch bank on an ACC-34 board sets a multiplexer port base address of $20 (32). MI92 should be set to $26 (38) to write to the 32-bit output port of the ACC-34.

MS{anynode},MI97 Multiplexer Port #1 Write Value

Range: $00000000 - $FFFFFFFF
Units: Individual bits

MI97 contains the 32-bit value written to the output port of the ACC-34 whose multiplexer port address is specified by MI96. Each bit represents one output on the port. Bit n of MI97 represents Output n on the port.

MS{anynode},MI98 Resolver Read Address

Range: $000000 - $0700FF
Units: Extended Station Multiplexer Port Addresses
Default: $000000

MI98 specifies the addresses on the MACRO Stations JTHW Multiplexer port of a set of resolver-to-digital converter absolute position registers on an ACC-8D Option 7 R/D converter board. If MI98 is set greater than 0, then the positions of these registers is copied periodically into Station variable MI99. If MI98 is set to 0, this function is not performed.

The low 8 bits – bits 0 to 7 (last two hex digits) – of MI98 must match the multiplexer port address of the ACC-8D Option 7 board from which the resolver position registers are to be read, as set by the SW1 DIP switch bank on the board.

The middle 8 bits – bits 8 to 15 (middle two hex digits) – of MI98 are normally all set to 0, making the middle two hex digits equal to $00. However, if the multiplexer port address in the low 8 bits is $00, then bit 8 of MI98 is set to 1, making the middle two hex digits equal to $01.

The high 8 bits – bits 16 to 23 (first two hex digits) – of MI98 are set to represent the number of the first (lowest numbered) of three consecutive converters on the board to be read. The value in these bits is a function of the setting of DIP-switch SW1-1 and the location of the converter on the board. The following table shows the possible settings and what each one represents:

MI98 Value	ACC-8D Opt. 7 SW1-1 Setting	# of R/D Converter on ACC-8D Opt. 7
$000xxx	CLOSED (0)	1
$010xxx	CLOSED (0)	2
$020xxx	CLOSED (0)	3
$030xxx	CLOSED (0)	4
$040xxx	OPEN (1)	1
$050xxx	OPEN (1)	2
$060xxx	OPEN (1)	3
$070xxx	OPEN (1)	4

The format of MI98 is the same as the format for MI11x on the MACRO Station, and Ix10 or Ix81 on the PMAC.

Examples:

To read R/D converters 2, 3, and 4 of an ACC-8D Opt. 7 board set for multiplexer port address 16 ($10) with SW1-1 closed, MI98 would be set to $020010.

To read R/D converters 1, 2, and 3 of an ACC-8D Opt. 7 board set for multiplexer port address 0 with SW1-1 closed, MI98 would be set to $000100.

MS{anynode},MI99 Resolver Read Value

Range: $00000000 - $FFFFFFFFF
Units: Extended Station Multiplexer Port Addresses
Default: $000000

If MI98 is set greater than 0, MI99 contains the absolute position values from 3 consecutive R/D converters at the multiplexer port address specified by MI98.
MI99 is a 36-bit value representing 3 12-bit positions.

- Bits 0 – 11 contain the position of the R/D converter whose address is directly specified by MI98.
- Bits 12 – 23 contain the position of the R/D converter numbered one higher than that specified by MI98.
- Bits 24 – 35 contain the position of the R/D converter numbered two higher than that specified by MI98

Position Processing MI-Variables

MS{anynode},MI101-MI108 Ongoing Position Source Address

Range: $0000 - $FFFF
Units: MACRO Station "X" Addresses
Default:

MI101 (1st motor node: Node 0):	$0010	{1st line of encoder conversion table}
MI102 (2nd motor node: Node 1):	$0011	{2nd line of encoder conversion table}
MI103 (3rd motor node: Node 4):	$0012	{3rd line of encoder conversion table}
MI104 (4th motor node: Node 5):	$0013	{4th line of encoder conversion table}
MI105 (5th motor node: Node 8):	$0014	{5th line of encoder conversion table}
MI106 (6th motor node: Node 9):	$0015	{6th line of encoder conversion table}
MI107 (7th motor node: Node 12):	$0016	{7th line of encoder conversion table}
MI108 (8th motor node: Node 13):	$0017	{8th line of encoder conversion table}

MI101 through MI108 (MI10*x*) determine what registers are used for feedback for the eight possible motor nodes (MI10*x* controls the *x*th motor node, which usually corresponds to Motor *x* on PMAC) on a MACRO Station.
For each active motor node, the value in the specified register is copied into the 24-bit position feedback MACRO register. Typically, the addresses specified are those from the MACRO Station's encoder conversion table, at Station registers X:$0010 to X:$002F, corresponding to Station MI-variables MI120 to MI151, respectively.

MS{anynode},MI109 - MI110 (Reserved for future use)

MS{anynode},MI111-MI118 Power-Up Position Source Address

Range: $000000 - $FFFFFF
Units: Extended MACRO Station Addresses
Default: 0

MI111 (1st motor node: Node 0)
MI112 (2nd motor node: Node 1)

MI113 (3rd motor node: Node 4)
MI114 (4th motor node: Node 5)
MI115 (5th motor node: Node 8)
MI116 (6th motor node: Node 9)
MI117 (7th motor node: Node 12)
MI118 (8th motor node: Node 13)

MI111 through MI118 (MI11*x*) specify whether, where, and how absolute position is to be read on the MACRO Station for a motor node (MI11*x* controls the *x*th motor node, which usually corresponds to Motor *x* on PMAC) and sent back to the PMAC or PMAC2.

If MI11*x* is set to 0, no power-on/reset absolute position value will be returned to PMAC. If MI11*x* is set to a value greater than 0, then when the PMAC requests the absolute position because its Ix10 and/or Ix81 values are set to obtain absolute position through MACRO (sending an auxiliary **`MS{node},MI920`** or **`MS{node},MI930`** command), the MACRO Station will use MI11*x* to determine how to read the absolute position, and report that position back to PMAC as an auxiliary response.

MI11*x* consists of two parts. The low 16 bits (last 4 hexadecimal digits) specify the address on the MACRO Station from which the absolute position information is read. The high 8 bits (first 2 hexadecimal digits) tell the MACRO Station how to interpret the data at that address (the "method".

The following table shows the possible values for MI11*x*, organized by the first 2 digits:

MI11n Bits 16-23 for Unsigned (Signed)	Type of Feedback	Notes
$00-$07 ($80-$87)	Resolver-to-Digital Converter	Used for ACC-8D Opt 7 connected to CPU board JTHW connector; address is multiplexer port address ($00 - $FF)
$08-$18 ($88-$98)	Single-Y-Word Parallel (8 to 24 bits)	Used for MDLT feedback; Value in B16-21 is number of bits to read
$17-$2A ($97-$AA)	Double-Y-Word Parallel (25 to 42 bits)	Value in B16-21 is number of bits; most significant bits are at {address + 1}
$2B ($AB)	Double-Byte Parallel (16 bits) in low bytes of 24-bit words	Used for ACC-3E parallel feedback; Most significant byte is at {address + 1}
$2C ($AC)	Double-Byte Parallel (16 bits) in middle bytes of 24-bit words	Used for ACC-3E parallel feedback; Most significant byte is at {address + 1}
$2D ($AD)	Double-Byte Parallel (16 bits) in middle bytes of 24-bit words	Used for ACC-3E parallel feedback; Most significant byte is at {address + 1}
$2E ($AE)	Triple-Byte Parallel (24 bits) in low bytes of 24-bit words	Used for ACC-3E parallel feedback; Middle byte is at {address + 1}; Most significant byte is at {address + 2}
$2F ($AF)	Triple-Byte Parallel (24 bits) in middle bytes of 24-bit words	Used for ACC-3E parallel feedback; Middle byte is at {address + 1}; Most significant byte is at {address + 2}
$30 ($B0)	Triple-Byte Parallel (24 bits) in middle bytes of 24-bit words	Used for ACC-3E parallel feedback; Middle byte is at {address + 1}; Most significant byte is at {address + 2}
$31 ($B1)	16-Bit Parallel in high 16 bits of 24 bit word	Used for ACC-28B A/D converter feedback
$32 ($B2)	Double 13-Bit Parallel	Used for Sanyo Absolute Encoder Interface
$33 ($B3)	12-Bit Parallel in high 12 bits of 24-bit word	Used for ACC-1E-B2 or ACC-6E A/D converter feedback
$48-$56 ($C8-$D6)	Single-X-Word Parallel (8 to 24 bits)	Value in B16-23 is number of bits to read
$57-$6A ($D7-$EA)	Double-X-Word Parallel (25 to 42 bits)	Value in B16-23 is number of bits; most significant bits are at {address + 1}
$71 ($F1)	Yaskawa Absolute Encoder Converter thru Multiplexer Port	Used for ACC-8D Opt 9 connected to CPU board JTHW port; address is multiplexer port address ($00 - $FF)
$72 ($F2)	Yaskawa Absolute Encoder Converter thru RS-232 interface	Used for ACC-8D Opt 9 connected to CPU board serial port.

If Bit 23 of MI11*x* is set to 1 (providing the value for Bits 16-23 shown in parentheses), then the position value read is sign extended to produce a signed position value. If Bit 23 is set to 0, no sign extension is performed, producing an unsigned positive position value. Bit 23 of PMAC's Ix10 for the motor using this MACRO node must be the same as Bit 23 of the Station's MI11*x*.

MS{anynode},MI119 (Reserved for future use)

MS{anynode},MI120-MI151 Encoder Conversion Table Entries

Range: $000000 - $FFFFFF
Units: Extended MACRO Station Addresses
Default: (dependent on SW1 setting)

MI120 through MI151 form the 32 setup lines of the MACRO Station's Encoder Conversion Table (ECT). The Encoder Conversion Table on the Station is similar in concept to that of the PMAC or PMAC2 itself; it is identical in structure to the Encoder Conversion Table of the Turbo PMAC. The MACRO Station's table is executed every ring cycle to prepare the feedback data to be sent back to the PMAC over the MACRO ring, where it will likely be passed through the PMAC's own table.

The ECT consists of a series of "entries", with each entry processing one feedback value. An entry in the ECT can have 1, 2, or 3 lines, therefore 1, 2, or 3 of these 24-bit MI-variables. Each MI-variable occupies a fixed register in the MACRO Station's memory map. The register addresses are important, because the results of the ECT are accessed by address.

Table Addresses: The following table shows the Station Y-address for each of the MI-variables in the table. The processed feedback value for an entry resides in the X-register of the same address as the last line of the entry. Variable MI10*x* for the *x*th motor node on the Station should contain the address of this X-register for the feedback it wants to send back to PMAC over the MACRO ring.

MI-Var.	Address	MI-Var.	Address	MI-Var.	Address	MI-Var.	Address
MI120	$0010	MI128	$0018	MI136	$0020	MI144	$0028
MI121	$0011	MI129	$0019	MI137	$0021	MI145	$0029
MI122	$0012	MI130	$001A	MI138	$0022	MI146	$002A
MI123	$0013	MI131	$001B	MI139	$0023	MI147	$002B
MI124	$0014	MI135	$001C	MI140	$0024	MI148	$002C
MI125	$0015	MI136	$001D	MI141	$0025	MI149	$002D
MI126	$0016	MI137	$001E	MI142	$0026	MI150	$002E
MI127	$0017	MI138	$001F	MI143	$0027	MI151	$002F

Entry First Line: The first line (MI-variable) in each entry consists of a source address in the low 16 bits, which contains the Station address of the raw data to be processed, and a "method" value in the high 8 bits, which specifies how this data is to be processed.

Entry Additional Lines: Depending on the method, 1 or 2 additional lines (MI-variables) may be required in the entry to provide further instructions on processing. If the first line (MI-variable) in the entry is $000000, this signifies the end of the active table, regardless of what subsequent entries in the table (higher numbered MI-variables) contain.

Method	# of lines	Process Defined	1st Additional Line	2nd Additional Line
$0x	1	1/T Extension of Incremental Encoder	-	-
$1x	1	ACC-28 style A/D converter (high 16 bits, no rollover)	-	-
$2x	2	Parallel Y-word data, no filtering	Bits-Used Mask	-
$3x	3	Parallel Y-word data, with filtering	Bits-Used Mask	Max Change per Cycle
$4x	2	"Time Base" scaled digital differentiation	Time Base Scale Factor	-
$5x	2	Integrated ACC-28 style A/D converter	Input Bias	-
$6x	2	Parallel X-word data, no filtering	Bits-Used Mask	-
$7x	3	Parallel X-word data, with filtering	Bits-Used Mask	Max Change per Cycle
$8x	1	Parallel Extension of Incremental Encoder	-	-
$9x	2	Triggered Time Base, frozen	Time Base Scale Factor	-
$Ax	2	Triggered Time Base, running	Time Base Scale Factor	-
$Bx	2	Triggered Time Base, armed	Time Base Scale Factor	-
$Cx	1	Incremental Encoder, no extension	-	-
$Dx	3	Exponential filter of parallel data	Max Change per Cycle	Filter Gain (Inverse Time Constant)
$Ex	1	Sum or difference of entries	-	-
$Fx	3	High-resolution Interpolator	Address of 1st A/D converter	A/D Bias Term

Digital Incremental Encoder Entries ($0x, $Cx): These two conversion table methods utilize the incremental encoder registers in the "DSPGATE" ASICs on the Station. Each method provides a processed result with the units of (1/32) count – the low 5 bits are fractional data.
With the $0x method, the fractional data is computed by dividing the "Time Since Last Count" register by the "Time Between Last 2 Counts" register. This technique is known as "1/T extension", and is the most commonly used method. It can be used with a digital incremental encoder connected directly to the Station
With the $Cx method, the fractional data is always set to zero, which means there is no extension of the incremental encoder count. This setting is used mainly to verify the effect of one of the 1/T extension, or the parallel extension of an analog encoder, explained below.
The 'x' in the second digit is always 0 in both of these methods.
With either of these conversion methods, the source address in the low 16 bits is that of the starting register of the machine interface channel. The addresses of the machine interface channels that can be used, and the ECT entry MI-variables that correspond to them, are shown in the following tables. The 'm' is the conversion method, representing '0' or 'C'.

Entries for Stack Axis boards (ACC-1E, ACC-2E)

Machine	MACRO	Conversion	Machine	MACRO	Conversion

Interface Channel #	Station Base Address	Table Entry	Interface Channel #	Station Base Address	Table Entry
1	$C000	$m0C000	6	$C028	$m0C028
2	$C008	$m0C008	7	$C030	$m0C030
3	$C010	$m0C010	8	$C038	$m0C038
4	$C018	$m0C018	9	$C090	$m0C090
5	$C020	$m0C020	10	$C098	$m0C098

Entries for Backplane Axis Boards (ACC-24E2x)

Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry	Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry
1	$C040	$m0C040	5	$C060	$m0C060
2	$C048	$m0C048	6	$C068	$m0C068
3	$C050	$m0C050	7	$C070	$m0C070
4	$C058	$m0C058	8	$C078	$m0C078

These are single-line entries in the table, so the next line (MI-Variable) is the start of the next entry.

Analog Incremental Encoder Entries ($8x, $Fx): These two entries process data from analog "sinewave" encoders through a Delta Tau interpolator, providing a high number of position states per line using "fractional count" data.

Low Resolution: With the $8x method, the fractional data is computed by reading the 5 inputs at bits 19-23 of the specified address (USER, W, V, U, and T flag inputs, respectively). This technique is known as "parallel extension", and can be used with an analog incremental encoder processed through the Option 1 interpolator of an ACC-8DE or 8FE 3U-format breakout board, or an external ACC-8D Opt 8 Analog Encoder Interpolator board, with the data brought in through an ACC-1E or 2E stack axis interface board.

This is a single-line entry, and the possible settings are shown in the table below.

Entries for Stack Axis boards (ACC-1E, ACC-2E)

Machine Interface Channel #	MACRO Station Address	Conversion Table Entry	Machine Interface Channel #	MACRO Station Address	Conversion Table Entry
1	$C000	$80C000	6	$C028	$80C028
2	$C008	$80C008	7	$C030	$80C030
3	$C010	$80C010	8	$C038	$80C038
4	$C018	$80C018	9	$C090	$80C090
5	$C020	$80C020	10	$C098	$80C098

Entries for Backplane Axis Boards (ACC-24E2x)

Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry	Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry
1	$C040	$80C040	5	$C060	$80C060
2	$C048	$80C048	6	$C068	$80C068
3	$C050	$80C050	7	$C070	$80C070
4	$C058	$80C058	8	$C078	$80C078

Depending on the jumper setting of the interpolator, the least significant bit of the result, which Station and PMAC software consider 1/32 of a count, is 1/128 or 1/256 of a line, or cycle, of the encoder.

High Resolution: With the $Fx method, the table computes the fractional information using the A/D-converter data from an ACC-51E high-resolution encoder interpolator, producing a value with 4096 states per line. The entry must read both an encoder channel for the whole number of lines of the encoder, and a pair of A/D converters to determine the location within the line, mathematically combining the values to produce a single position value.

***Encoder Channel Address*:** The first line of the three-line entry contains $F in the first hex digit and the base address of the encoder channel to be read in the last four digits (bits 0 to 15). The following table shows the possible entries for an ACC-51E in the station.

Entry First Lines for ACC-51E Backplane Interpolator Boards

ACC-51E #	Channel 1	Channel 2	Channel 3	Channel 4
1st	$F0C040	$F0C048	$F0C050	$F0C058
2nd	$F0C060	$F0C068	$F0C070	$F0C078

***A/D Converter Address*:** The second line of the entry contains the base address of the first A/D converter to be read in the last four digits (bits 0 to 15). The second A/D converter will be read at the next higher address. The following table shows the possible settings when the ACC-51E is used.

Entry Second Lines for ACC-51E Backplane Interpolator Boards

ACC-51E #	Channel 1	Channel 2	Channel 3	Channel 4
1st	$00C045	$00C04D	$00C055	$00C05D
2nd	$00C065	$00C06D	$00C075	$00C07D

***A/D Bias Term*:** The third line of the entry contains the bias in the A/D converter values. This line should contain the value that the A/D converters report when they should ideally report zero. The MACRO Station subtracts this value from both A/D readings before calculating the arctangent. Many users will leave this value at 0, but it is particularly useful to remove the offsets of single-ended analog encoder signals.

This line is scaled so that the maximum A/D converter reading provides the full value of the 24-bit register (+/-2^{23}). It is generally set by reading the A/D converter values directly as 24-bit values, computing the average value over a cycle or cycles, and entering this value here.

***Conversion Result*:** The result of the conversion is placed in the X-register of the third line of the entry. Careful attention must be paid to the scaling of this 24-bit result. The least significant bit (Bit 0) of the result represents 1/4096 of a line of the sine/cosine encoder.

When this data is passed to a PMAC, and it reads this data for servo use with Ix03, Ix04, Ix05, or Ix93, it expects to find data in units of 1/32 of a "count". Therefore, PMAC software regards this format as producing 128 "counts" per line. (The fact that the hardware counter used produces 4

counts per line is not relevant to the actual use of this format; this fact would only be used when reading the actual hardware counter for debugging purposes.)

***Example*:**

This format is used to interpolate a linear scale with a 40-micron pitch (40μm/line), producing a resolution of about 10 nanometers (40,000/4096), used as position feedback for a motor. PMAC considers a "count" to be 1/128 of a line, yielding a count length of 40/128 = 0.3125 μm. To set user units of millimeters for the axis, the axis scale factor would be:

$$AxisScaleFactor = \frac{1mm}{UserUnit} * \frac{1000\mu m}{mm} * \frac{count}{0.3125\mu m} = 3200\frac{counts}{UserUnit}$$

ACC-28 Style A/D Entries ($1x, $5x): The "A/D" feedback entries read from the high 16 bits of the specified address and shift the data right three bits so that the least significant bit of the processed result in bit 5. Unlike the "parallel feedback" methods, this method will not "roll over" and extend the result.

This data can come from an ACC-28B connected through an ACC-1E or 2E stack axis board through a 100-pin connector (usually with an ACC-8T breakout board), or from an ACC-28E backplane A/D board.

The $1x method processes the information directly, essentially a copying with shift. The $5x integrates the input value as it copies and shifts it. That is, it reads the input value, shifts it right three bits, adds the bias term in the second line, and adds this value to the previous processed result.

If the second digit 'x' of the entry is '0', the 16-bit source value is treated as a signed quantity; if it is '8', the 16-bit value is treated as an unsigned quantity. Presently, the only A/D accessory of this format that can interface to the MACRO Station is the ACC-28B, which provides an unsigned value, so $18 and $58 should be used.

The following table shows the addresses of the ADC registers that can be used with an ACC-28B connected through an ACC-1E or 2E, and the corresponding ECT entry for each register. The 'm' represents the conversion method, either '1' or '5'.

Entries for ACC-28B ADCs

Machine Interface Channel #	Channel ADC A Register Address	Encoder Conversion Table Entry	Channel ADC B Register Address	Entry
1	Y:$C005	$m8C005	Y:$C006	$m8C006
2	Y:$C00D	$m8C00D	Y:$C00E	$m8C00E
3	Y:$C015	$m8C015	Y:$C016	$m8C016
4	Y:$C01D	$m8C01D	Y:$C01E	$m8C01E
5	Y:$C025	$m8C025	Y:$C026	$m8C026
6	Y:$C02D	$m8C02D	Y:$C02E	$m8C02E
7	Y:$C035	$m8C035	Y:$C036	$m8C036
8	Y:$C03D	$m8C03D	Y:$C03E	$m8C03E
9	Y:$C095	$m8C095	Y:$C096	$m8C096
10	Y:$C09D	$m8C09D	Y:$C09E	$m8C09E

The following table shows the entries for ACC-28E backplane converter board ADCs. The 'm' represents the conversion method, either '1' or '5'.

Entries for ACC-28E ADCs

ACC-28E Base Address	Entry for ADC1	Entry for ADC2	Entry for ADC3	Entry for ADC4
$FFE0	$m8FFE0	$m8FFE1	$m8FFE2	$m8FFE3
$FFE8	$m8FFE8	$m8FFE9	$m8FFEA	$m8FFEB
$FFF0	$m8FFf0	$m8FFF1	$m8FFF2	$m8FFF3
$FFF8	$m8FFF8	$m8FFF9	$m8FFFA	$m8FFFB

Parallel Feedback Entries ($2x, $3x, $6x, $7x): The "parallel feedback" entries read a word from the address specified in the low 16 bits of the first entry. The four methods in this class are:

- **$2x:** Y-word parallel, no filtering (2-line entry)
- **$3x:** Y-word parallel, with filtering (3-line entry)
- **$6x:** X-word parallel, no filtering (2-line entry)
- **$7x**: X-word parallel, with filtering (3-line entry)

The second digit in the first line of the entry, represented above by 'x', specifies how the parallel data at the specified address is to be processed. Currently there are 5 valid values of 'x':

- **x=0:** Shift data so that the least significant bit of the source register as specified in the "bits used" mask word is placed in bit 5 of the processed result.
- **x=4:** Read the least significant byte from the low byte of the specified address; read the middle byte from the low byte of the (specified address + 1); read the most significant byte from the low byte of the (specified address + 2). This is used for feedback brought in through the ACC-3E 144-I/O board J4, J5, or P1 connectors.
- **x=5:** Read the least significant byte from the middle byte of the specified address; read the middle byte from the middle byte of the (specified address + 1); read the most significant byte from the middle byte of the (specified address + 2). This is used for feedback brought in through the ACC-3E 144-I/O board J6 or J7 connectors.
- **x=6:** Read the least significant byte from the middle byte of the specified address; read the middle byte from the middle byte of the (specified address + 1); read the most significant byte from the middle byte of the (specified address + 2). This is used for feedback brought in through the ACC-3E 144-I/O board J8 or J9 connectors.

- **x=8:** Process the data from the source register without any shifting, so the least significant bit of the source register as specified in the "bits used" mask word is place in bit 0 of the processed result.

Time Base Entries ($4x): A time-base entry performs a scaled digital differentiation of the value in the source register. It is a two-line entry. The first line contains a '4' in the first hex digit and the address of the source register in the last four hex digits. The source register is almost always the result register of an incremental encoder entry higher in the table (addresses $0020 to $003F). The second line in the entry is the "time-base scale factor". The result value *equals 2 * Time-Base-Scale-Factor * (New Source Value - Old Source Value).* When this entry is used to synchronize a motion program to a master encoder, creating an electronic cam function, this scale factor should be set equal to 2^{17} */ Real-Time-Input-Frequency*, where the RTIF is expressed in counts per millisecond. The program is then written assuming that the master encoder is always putting out this RTIF.

Triggered Time Base Entries ($9x, $Ax, $Bx): A triggered time-base entry is like a regular time-base entry, except that it is easy to "freeze" the time base, then start it exactly on receipt of a trigger that captures the "starting" master position or time.

The source register for triggered time base must be the starting (X) address for one of the machine interface channels on the Station. The following table shows the addresses for each channel on the ACC-1E and 2E stack axis boards, and the corresponding ECT entry. The 'm' represents the method, either '9', 'A', or 'B'.

Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry	Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry
1	$C000	$m0C000	6	$C028	$m0C028
2	$C008	$m0C008	7	$C030	$m0C030
3	$C010	$m0C010	8	$C038	$m0C038
4	$C018	$m0C018	9	$C090	$m0C090
5	$C020	$m0C020	10	$C098	$m0C098

The following table shows the addresses for each channel on the ACC-24E2x backplane axis boards, and the corresponding ECT entry. The 'm' represents the method, either '9', 'A', or 'B'.

Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry	Machine Interface Channel #	MACRO Station Base Address	Conversion Table Entry
1	$C040	$m0C040	5	$C060	$m0C060
2	$C048	$m0C048	6	$C068	$m0C068
3	$C050	$m0C050	7	$C070	$m0C070
4	$C058	$m0C058	8	$C078	$m0C078

In use, the method byte is changed as needed by setting of the MI-variable. It is set to $90 (e.g. MI129=$90C018) before the calculations of the triggered move are started, to freeze the time base. It is set to $B0 (e.g. MI129=$B0C018) after the calculations of the triggered move are finished, to "arm" the time base for the trigger. When the Table sees the trigger (the capture trigger for the machine interface channel as defined by MI912 and MI913 for the channel), it automatically sets the method byte to $A0 for "running" time base.
The second line in the entry is the "time-base scale factor". The result value *equals 2 * Time-Base-Scale-Factor * (New Source Value - Old Source Value).* When this entry is used to synchronize a motion program to a master encoder, creating an electronic cam function, this scale factor should be set equal to 2^{17} */ Real-Time-Input-Frequency*, where the RTIF is expressed in counts per millisecond. The program is then written assuming that the master encoder is always putting out this RTIF.
Addition/Subtraction of Entries ($E0, $E8): The $Ex entry is used to add or subtract two other entries in the Table. If the method byte is $E0, the two specified entries are added. If the method byte is $E8, the second entry is subtracted from the first.
Bits 0-7 of the entry specify the address offset from this entry to the first entry to be used, as a signed 8-bit quantity. Bits 8-15 of the entry specify the offset from this entry to the second entry to be used. For example, MI131 is to be used to subtract the result values with MI121 from that of MI120, the offset to the first entry is -11 ($F5), and the offset to the second entry is -10 ($F6). Therefore MI131=$E8F6F5.

MS{anynode},MI152 - MI153 Phase-Clock Latched I/O

Range: $000000000000 - $FFFFFFFFFFFF
Units: Extended MACRO Station Y Addresses
Default: $000000000000

MI152 and MI153 permit the use of inputs latched by the phase clock on Station I/O boards. This function is used to get reliable parallel-data feedback on the MACRO Station. It is useful mainly on ACC-3E stack boards and ACC-14E backplane boards.

Note:

Jumper E5 on the ACC-3E 144-I/O piggyback board must connect pins 2 and 3 to permit this function. Jumper E19 on the ACC-4E 48-Isolated-I/O piggyback board must connect pins 2 and 3 to permit this function.

MI152 and MI153 are 48-bit values represented by 12 hexadecimal digits. These digits have the following functions:

Digits	Function and Setting

1 & 2	3rd I/O ASIC Latch Control (Maps into high bytes; Option C on ACC-3E 144-I/O board, ACC-9E, 10E, 11E, 12E with E6x connecting rows 4 & 5) =$C0 for latched inputs =$00 for transparent inputs or ASIC not present
3 & 4	2nd I/O ASIC Latch Control (Maps into middle bytes; Option B on ACC-3E 144-I/O board, ACC-9E, 10E, 11E, 12E with E6x connecting rows 2 & 3) =$C0 for latched inputs =$00 for transparent inputs or ASIC not present
5 & 6	2nd I/O ASIC Latch Control (Maps into low bytes; Option A on ACC-3E 144-I/O board, only ASIC on ACC-4E &14E I/O board, ACC-9E, 10E, 11E, 12E with E6x connecting rows 1 & 2) =$C0 for latched inputs =$00 for transparent inputs or ASIC not present
7	Number of bytes (1 to 6) on each ASIC (starting with lowest byte) to latch
8	(Reserved for future use; set to 0)
9 - 12	Base address of I/O Board =$FFC0 (ACC-3E board w/ E1 ON; ACC-4E board w/ E15 ON) =$FFC8 (ACC-3E board w/ E2 ON; ACC-4E board w/ E16 ON) =$FFD0 (ACC-3E board w/ E3 ON; ACC-4E board w/ E17 ON) =$FFD8 (ACC-3E board w/ E4 ON; ACC-4E board w/ E18 ON) =$FFE0 (ACC-9E, 10E, 11E, 12E board w/ E1 ON, ACC-14E) =$FFE8 (ACC-9E, 10E, 11E, 12E board w/ E2 ON, ACC-14E) =$FFF0 (ACC-9E, 10E, 11E, 12E board w/ ON, ACC-14E) =$FFF8 (ACC-9E, 10E, 11E, 12E board w/ E4 ON, ACC-14E)

Examples:

```
MS0,MI152=$C0C0C060FFC8     ; Latches inputs on all 3 ASICs, all 6 bytes per ASIC,
                            ; of an ACC-3E board with E2 ON.
MS0,MI153=$0000C030FFD0     ; Latches inputs on 1st ASIC, 1st 3 bytes, of an
                            ; ACC-3E board with E3 ON, or an ACC-4E board
                            ; with E17 ON
```

MS{anynode},MI154 - MI160 (Reserved for future use)

MS{anynode},MI161-MI168 MLDT Frequency Control

Range: $000000 - $FFFFFF

Units: PFMCLK cycles

Default: 0

MI161 (1st motor node: Node 0)
MI162 (2nd motor node: Node 1)
MI163 (3rd motor node: Node 4)
MI164 (4th motor node: Node 5)
MI165 (5th motor node: Node 8)
MI166 (6th motor node: Node 9)
MI167 (7th motor node: Node 12)
MI168 (8th motor node: Node 13)

MI161 through MI168 (MI16*x*) on the MACRO Station permit the 'C' output channel associated with the MACRO motor node (MI16*x* controls the *x*th motor node, which usually corresponds to Motor *x* on PMAC) to put out a specified output frequency, starting immediately on power-on/reset, for the purposes of creating an excitation signal for an MLDT sensor.

If MI16*x* is set to 0, this function is not enabled, and the 'C' output channel can be used for servo control functions such as PFM stepper control or direct PWM servo control.

If MI16*x* is set to a value greater than 0, then the 24-bit value in MI16*x* is automatically written to the 'C' output register of the machine interface channel associated with the MACRO node upon power-up or reset of the MACRO Station. Also, during the normal operation of the node, the value in the third MACRO register is not copied into the 'C' output register.

For the MLDT excitation to work properly, the MACRO Station variable MI916 for the node must be set for 2 or 3 to get PFM style output from the 'C' output channel. MI910 for the node must be set to 12 to use the timer for the MLDT feedback.

To compute the output frequency as a function of MI16*x*, the following formula can be used:

Output_Freq (Hz) = PFMCLK_Freq (Hz) * MI16*x* / 16,777,216

To compute the value of MI16*x* required to produce a desired output frequency, the following formula can be used:

MI16*x* = 16,777,216 * Output_Freq (Hz) / PFMCLK_Freq (Hz)

The PFMCLK frequency is set by MI903 for machine interface channels 1 – 4; by MI907 for machine interface channels 5 – 8; and by MI993 for machine interface channels 9-10.

I/O Transfer MI-Variables

MS{anynode},MI169, MI170 I/O-Board 72-Bit Transfer Control

Range: $000000000000 - $FFFFFFFFFFFF

Units: Extended addresses

Default: 0

MI69 and MI70 specify the registers used in 72-bit I/O transfers between MACRO node interface registers and I/O registers on the ACC-3E, 4E, 9E, 10E, 11E, 12E and 14E I/O boards on a MACRO station. They are only used if MI19 is greater than 0.

MI169 and MI170 are 48-bit variables represented as 12 hexadecimal digits. The first 6 digits specify the number and address of the 72-bit (1x24 and 3x16) real-time MACRO-node register set to be used. The second 6 digits specify the number and address of 16-bit I/O sets on an I/O board to be used. The individual digits are specified as follows:

Digit #	Possible Values	Description

1	0	(Reserved for future use)
2	0	(Reserved for future use)
3-6	$C0A0 (Node 2), $C0A4 (Node 3), $C0A8 (Node 6), $C0AC (Node 7), $C0B0 (Node 10), $C0B4 (Node 11)	MACRO Station X Address of MACRO I/O node 24-bit register
7	0	(Reserved for future use)
8	0	(Reserved for future use)
9-12	$FFC0, $FFC8, $FFD0, $FFD8 $FFE0, $FFE8, $FFF0, $FFF8	MACRO Station Y Base Address of I/O Board as set by Board Jumper E1-E4 (ACC-3E board) or E15-E18 (ACC-4E board) MACRO Station Y Base Address of ACC-9E, 10E, 11E, 12E, or 14E UMAC I/O board as set by jumpers/switches on board

When this function is active, the MACRO Station will copy values from the MACRO command (input) node registers to the I/O board addresses; it will copy values from the I/O board addresses to the MACRO feedback (output) node registers. Writing a '0' to a bit of the I/O board enables it as an input, letting the output pull high. Writing a '1' to a bit of the I/O board enables it as an output and pulls the output low.

The following table shows the mapping of I/O points on the I/O piggyback boards to the MACRO node registers.

I/O Point #s	ACC-3E Part	Present on ACC-4E?	Matching MACRO X Register
I/O00 - I/O15	Option A	Yes	Specified MACRO X Address + 1
I/O16 - I/O31	Option A	Yes	Specified MACRO X Address + 2
I/O32 - I/O47	Option A	Yes	Specified MACRO X Address + 3
I/O48 - I/O71	Option B	No	Specified MACRO X Address + 0

The following table shows the mapping of I/O points on the I/O backplane boards to the MACRO node registers:

Board # at Set Address	E6x Rows Connected	Byte on Data Bus	I/O Point #s on Board	Matching MACRO X Register
1st	1 & 2	Low	0 –15	Specified MACRO X Address + 1
1st	1 & 2	Low	16 – 31	Specified MACRO X Address + 2
1st	1 & 2	Low	32 - 47	Specified MACRO X Address + 3
2nd	2 & 3*	Middle	0 –23	Specified MACRO X Address + 0

* Rows 3 & 4 connected creates same setting

Note

The ACC-14E backplane I/O board can only be set up for the low byte on the data bus.

Examples:

MI169=$00C0A000FFE0 transfers 72-bit I/O between an I/O board set at $FFE0 and MACRO Node 2 ($C0A0-$C0A3).
MI170=$10C0B000FFE8 transfers 72-bit I/O between an I/O board set at $FFE8 and MACRO Node 10 ($C0B0-$C0B3).

MS{anynode},MI171, MI172, MI173 I/O-Board 144-Bit Transfer Control

Range: $000000000000 - $FFFFFFFFFFFF

Units: Extended addresses
Default: 0

MI171, MI172, and MI173 specify the registers used in 144-bit I/O transfers between MACRO I/O node interface registers and I/O registers on the ACC-3E, 9E, 10E, 11E, and 12E I/O boards on a MACRO station. It is only used if MI19 is greater than 0.
The transfer utilizes two consecutive 72-bit X-memory MACRO I/O nodes and 3 48-bit "IOGATE" I/O ICs that occupy different bytes (low, middle, and high) of the same base address. MI171, MI172, and MI173 are 48-bit variables each represented as 12 hexadecimal digits. The first 6 digits specify the address of the first 72-bit real-time MACRO-node register sets to be used. The second 6 digits specify the address of the three 48-bit I/O sets on the I/O board to be used. The individual digits are specified as follows:

Digit #	Possible Values	Description
1	0	(Reserved for future use)
2	0	(Reserved for future use)
3-6	$C0A0 (Node 2), $C0A4 (Node 3), $C0A8 (Node 6), $C0AC (Node 7), $C0B0 (Node 10), $C0B4 (Node 11)	MACRO Station X Address of MACRO I/O node first of four registers
7	0	(Reserved for future use)
8	0	(Reserved for future use)
9-12	$FFC0, $FFC8, $FFD0, $FFD8 $FFE0, $FFE8, $FFF0, $FFF8	MACRO Station Y Base Address of I/O Board as set by Board Jumper E1-E4 (ACC-3E board MACRO Station Y Base Address of ACC-9E, 10E, 11E, 12E UMAC I/O board as set by jumpers on board

When this function is active, the MACRO Station will copy values from the MACRO command (input) node registers to the I/O board addresses; it will copy values from the I/O board addresses to the MACRO feedback (output) node registers. Writing a '0' to a bit of the I/O board enables it as an input, letting the output pull high. Writing a '1' to a bit of the I/O board enables it as an output and pulls the output low.

The following table shows the mapping of I/O points on the I/O piggyback boards to the MACRO node registers.

I/O Point #s	ACC-3E Part	Byte on ACC-9E, 10E, 11E, 12E	Matching MACRO Node X Register
I/O00 - I/O15	Option A	Low	Specified MACRO Node X Address + 1
I/O16 - I/O31	Option A	Low	Specified MACRO Node X Address + 2
I/O32 - I/O47	Option A	Low	Specified MACRO Node X Address + 3
I/O48 - I/O63	Option B	Middle	Specified MACRO Node X Address + 5
I/O64 - I/O79	Option B	Middle	Specified MACRO Node X Address + 6
I/O80 - I/O95	Option B	Middle	Specified MACRO Node X Address + 7
I/O96 - I/O119	Option C	High	Specified MACRO Node X Address + 0
I/O120 - I/O143	Option C	High	Specified MACRO Node X Address + 4

The following table shows the mapping of I/O points on the I/O backplane boards to the MACRO node registers:

Board # at Set Address	E6x Rows Connected	Byte on Data Bus	I/O Point #s on Board	Matching MACRO X Register
1st	1 & 2	Low	0 –15	Specified MACRO X Address + 1
1st	1 & 2	Low	16 – 31	Specified MACRO X Address + 2
1st	1 & 2	Low	32 - 47	Specified MACRO X Address + 3
2nd	2 & 3*	Middle	0 –15	Specified MACRO X Address + 5
2nd	2 & 3*	Middle	16 – 31	Specified MACRO X Address + 6
2nd	2 & 3*	Middle	32 - 47	Specified MACRO X Address + 7
3rd	4 & 5	High	0 –23	Specified MACRO X Address + 0
3rd	4 & 5	High	24 – 47	Specified MACRO X Address + 4

* Rows 3 & 4 connected creates same setting

Note

The ACC-14E backplane I/O board can only be set up for the low byte on the data bus.

MS{anynode},MI174 – MI197 (Reserved for future use)

MS{anynode},MI198 Direct Read/Write Format and Address

Range: $000000 - $FFFFFF
Units: Modified MACRO Station Addresses
Default: $000000

MI198 controls the address and format of the register to be accessed (read from or written to) with MI199. This permits the access to any register on the MACRO Station by first assigning a value to MI198, then either reading MI199 or writing to it.

MI198 is a 24-bit variable that can be expressed as six hexadecimal digits. The low 16 bits, represented by the last four hex digits, represent the MACRO Station address of the register. The high eight bits, represented by the first two hex digits, represent the format of that address. The table below shows the legal entries for the first two digits and the format each represents.

For example, for the host computer to read the contents of the DAC1A register as a signed quantity – the high 16 bits of Y:$C002 – of the MACRO Station through a PMAC board, MI198 would be set to $6DC002, then MI199 would be read. For a MACRO Station with an active node 0, this could be done with the on-line commands:

```
MS0, MI198=$6DC002
MS0, MI199
16384
```

In another example, to read the state of Channel 2's encoder A input – bit 12 of X:$C008 – through a PMAC board, MI198 would be set to $8CC008, then MI99 would be read.

MI198 Format Digits

MI198 Digits	Address Space	Starting Bit	Bit Width	Format	MI198 Digits	Address Space	Starting Bit	Bit Width	Format
$00	Y	0	2	U	**$80**	X	0	1	U
$01	Y	2	2	U	**$81**	X	1	1	U
$02	Y	4	2	U	**$82**	X	2	1	U
$03	Y	6	2	U	**$83**	X	3	1	U
$04	Y	8	2	U	**$84**	X	4	1	U
$05	Y	10	2	U	**$85**	X	5	1	U
$06	Y	12	2	U	**$86**	X	6	1	U
$07	Y	14	2	U	**$87**	X	7	1	U
$08	Y	16	2	U	**$88**	X	8	1	U
$09	Y	18	2	U	**$89**	X	9	1	U
$0A	Y	20	2	U	**$8A**	X	10	1	U
$0B	Y	22	2	U	**$8B**	X	11	1	U
$0C	-	-	-	-	**$8C**	X	12	1	U
$0D	-	-	-	-	**$8D**	X	13	1	U
$0E	-	-	-	-	**$8E**	X	14	1	U
$0F	-	-	-	-	**$8F**	X	15	1	U
$10	Y	0	1	U	**$90**	X	16	1	U
$11	Y	1	1	U	**$91**	X	17	1	U
$12	Y	2	1	U	**$92**	X	18	1	U
$13	Y	3	1	U	**$93**	X	19	1	U
$14	Y	4	1	U	**$94**	X	20	1	U
$15	Y	5	1	U	**$95**	X	21	1	U
$16	Y	6	1	U	**$96**	X	22	1	U
$17	Y	7	1	U	**$97**	X	23	1	U
$18	Y	8	1	U	**$98**	X	0	4	U
$19	Y	9	1	U	**$99**	X	0	4	S
$1A	Y	10	1	U	**$9A**	-	-	-	-
$1B	Y	11	1	U	**$9B**	-	-	-	-
$1C	Y	12	1	U	**$9C**	X	4	4	U
$1D	Y	13	1	U	**$9D**	X	4	4	S
$1E	Y	14	1	U	**$9E**	-	-	-	-
$1F	Y	15	1	U	**$9F**	-	-	-	-
$20	Y	16	1	U	**$A0**	X	8	4	U
$21	Y	17	1	U	**$A1**	X	8	4	S
$22	Y	18	1	U	**$A2**	-	-	-	-
$23	Y	19	1	U	**$A3**	-	-	-	-
$24	Y	20	1	U	**$A4**	X	12	4	U
$25	Y	21	1	U	**$A5**	X	12	4	S
$26	Y	22	1	U	**$A6**	-	-	-	-
$27	Y	23	1	U	**$A7**	-	-	-	-
$28	Y	0	4	U	**$A8**	X	16	4	U
$29	Y	0	4	S	**$A9**	X	16	4	S
$2C	Y	4	4	U	**$AC**	X	20	4	U
$2D	Y	4	4	S	**$AD**	X	20	4	S
$30	Y	8	4	U	**$B0**	X	0	8	U
$31	Y	8	4	S	**$B1**	X	0	8	S
$34	Y	12	4	U	**$B4**	X	4	8	U

MI198 Format Digits (continued)

MI198 Digits	Address Space	Starting Bit	Bit Width	Format	MI198 Digits	Address Space	Starting Bit	Bit Width	Format
$35	Y	12	4	S	**$B5**	X	4	8	S
$38	Y	16	4	U	**$B8**	X	8	8	U
$39	Y	16	4	S	**$B9**	X	8	8	S
$3C	Y	20	4	U	**$BC**	X	12	8	U
$3D	Y	20	4	S	**$BD**	X	12	8	S
$40	Y	0	8	U	**$C0**	X	16	8	U
$41	Y	0	8	S	**$C1**	X	16	8	S
$44	Y	4	8	U	**$C4**	X	0	12	U
$45	Y	4	8	S	**$C5**	X	0	12	S
$48	Y	8	8	U	**$C8**	X	4	12	U
$49	Y	8	8	S	**$C9**	X	4	12	S
$4C	Y	12	8	U	**$CC**	X	8	12	U
$4D	Y	12	8	S	**$CD**	X	8	12	S
$50	Y	16	8	U	**$D0**	X	12	12	U
$51	Y	16	8	S	**$D1**	X	12	12	S
$54	Y	0	12	U	**$D4**	X	0	16	U
$55	Y	0	12	S	**$D5**	X	0	16	S
$58	Y	4	12	U	**$D8**	X	4	16	U
$59	Y	4	12	S	**$D9**	X	4	16	S
$5C	Y	8	12	U	**$DC**	X	8	16	U
$5D	Y	8	12	S	**$DD**	X	8	16	S
$60	Y	12	12	U	**$E0**	X	0	20	U
$61	Y	12	12	S	**$E1**	X	0	20	S
$64	Y	0	16	U	**$E4**	X	4	20	U
$65	Y	0	16	S	**$E5**	X	4	20	S
$68	Y	4	16	U	**$E8**	X	0	24	U
$69	Y	4	16	S	**$E9**	X	0	24	S
$6C	Y	8	16	U	**$EC**	-	-	-	-
$6D	Y	8	16	S	**$ED**	-	-	-	-
$70	Y	0	20	U	**$F0**	X	0	2	U
$71	Y	0	20	S	**$F1**	X	2	2	U
$72	-	-	-	-	**$F2**	X	4	2	U
$73	-	-	-	-	**$F3**	X	6	2	U
$74	Y	4	20	U	**$F4**	X	8	2	U
$75	Y	4	20	S	**$F5**	X	10	2	U
$76	-	-	-	-	**$F6**	X	12	2	U
$77	-	-	-	-	**$F7**	X	14	2	U
$78	Y	0	24	U	**$F8**	X	16	2	U
$79	Y	0	24	S	**$F9**	X	18	2	U
$7A	-	-	-	-	**$FA**	X	20	2	U
$7B	-	-	-	-	**$FB**	X	22	2	U

MS{anynode},MI199 Direct Read/Write Variable

Range: -8,388,608 – 16,777,215
Units: (dependent on register addressed)
Default: none

MI199 is a variable that can be addressed to any register in the MACRO Station's memory and I/O map, in order to read a value directly from that register, or write a value directly to that register. This permits easy access to any register on the MACRO Station.

The address of the register to be accessed, which part of this register, and how the data is to be interpreted, is set by MI198. The value of MI198 must be set properly before MI199 can be used to access the register. For repeated access of the same register with MI199, MI198 only needs to be set once.

Example:

```
MS0,MI198=$79C03C       ; Set to Y:$C03C,0,24,S (PFM8 command value)
MS0,MI199               ; Request value of this register
0                       ; PMAC reports this value
MS0,MI199=65536         ; Set to new value through PMAC
```

MS{anynode},MI200 - MI899 (Reserved for future use)

4-Axis Board Global MI-variables

MI-variables in the range MI900 to MI909 control multi-channel aspects of the hardware setup using the ACC-2E stack boards or ACC-24E2x backplane boards on a MACRO Station. On an ACC-1 stack board, the jumper E1 determines whether the board is mapped as channels 1-4 or 5-8. On an ACC-24E2x backplane board, or an ACC-51E backplane interpolator board, SW1-1 and 2 determine whether the board is mapped as channels 1-4 or 5-8. If both the stack and backplane boards are present for a given set of channel numbers, these variables are used for the stack boards.

MS{anynode},MI900 PWM 1-4 Frequency Control

Range: 0 - 32767
Units: PWM Frequency = 117,964.8 kHz / [4*MI900+6]
Default: 6527
PWM Frequency = 117,964.8 / 26114 = 4.5173 kHz

MI900 controls the PWM frequency for MACRO Station machine interface channels 1-4. It does this by setting the limits of the PWM up-down counter, which increments and decrements at the PWMCLK frequency of 117,964.8 kHz (117.9648 MHz).

The PWM frequency determines the actual switching frequency of amplifiers connected to any of the MACRO Station's first four machine interface channels with the direct PWM command. It is only important if the direct PWM command signal format is used.

Generally, MI900 is set to the same value as MI992. If a different PWM frequency is desired for channels 1 to 4, MI900 should be set so that it is an odd-integer multiple (e.g. 3x, 5x, 7x) of MI992, or that MI992 is an odd-integer multiple of MI900. This will keep the PWM hardware on channels 1-4 in synchronization with the software algorithms driven by the PHASE clock.

The maximum value that can be written into the PWM command register without full saturation is MI900+1 on the positive end, and -MI900-2 on the negative end. Generally, the "PWM scale factor" Ix66 for Motor x, which determines the maximum PWM command magnitude, is set to MI900 + 10%.

To set MI900 for a desired PWM frequency, the following formula can be used:

MI900 = (117,964.8 kHz / [4*PWM Freq (kHz)]) - 1 (rounded down)

Example:

To set a PWM frequency of 10 kHz:

MI900 = (117,964.8 kHz / [4*10 kHz]) - 1 = 2948

To set a PWM frequency of 7.5 kHz:

MI900 = (117,964.8 kHz / [4*7.5 kHz]) - 1 = 3931

MS{anynode},MI903 Hardware Clock Control Channels 1-4

Range: 0 - 4095

Units: MI903 = Encoder SCLK Divider
+ 8 * PFM_CLK Divider
+ 64 * DAC_CLK Divider
+ 512 * ADC_CLK Divider

where:
Encoder SCLK Frequency = 39.3216 MHz / (2 ^ Encoder SCLK Divider)
PFM_CLK Frequency = 39.3216 MHz / (2 ^ PFM_CLK Divider)
DAC_CLK Frequency = 39.3216 MHz / (2 ^ DAC_CLK Divider)
ADC_CLK Frequency = 39.3216 MHz / (2 ^ ADC_CLK Divider)

Default: 2258 = 2 + (8 * 2) + (64 * 3) + (512 * 4)
Encoder SCLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz
PFM_CLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz
DAC_CLK Frequency = 39.3216 MHz / (2 ^ 3) = 4.9152 MHz
ADC_CLK Frequency = 39.3216 MHz / (2 ^ 4) = 2.4576 MHz

MI903 controls the frequency of four hardware clock frequencies -- SCLK, PFM_CLK, DAC_CLK, and ADC_CLK -- for channels 1-4 on a MACRO Station (on a 4-axis piggyback board with jumper E1 connecting 1-2). It is a 12-bit variable consisting of four independent 3-bit controls, one for each of the clocks. Each of these clock frequencies can be divided down from a starting 39.3216 MHz frequency by powers of 2, 2^N, from 1 to 128 times (N=0 to 7). This means that the possible frequency settings for each of these clocks are:

Frequency	Divide by	Divider N in $1/2^N$
39.3216 MHz	1	0
19.6608 MHz	2	1
9.8304 MHz	4	2
4.9152 MHz	8	3
2.4576 MHz	16	4
1.2288 MHz	32	5
611.44 kHz	64	6
305.72 kHz	128	7

Very few MACRO Station users will be required to change the setting of MI903 from the default value.

The encoder sample clock signal SCLK controls how often the MACRO Station's digital hardware looks at the encoder and flag inputs. The MACRO Station can take at most one count per SCLK cycle, so the SCLK frequency is the absolute maximum encoder count frequency.

SCLK also controls the signal propagation through the digital delay filters for the encoders and flags; the lower the SCLK frequency, the greater the noise pulse that can be filtered out. The SCLK frequency should optimally be set to the lowest value that can accept encoder counts at the maximum possible rate.
The pulse-frequency-modulation clock PFM_CLK controls the PFM circuitry that is commonly used for stepper drives. The maximum pulse frequency possible is 1/4 of the PFM_CLK frequency. The PFM_CLK frequency should optimally be set to the lowest value that can generate pulses at the maximum frequency required.
The DAC_CLK controls the serial data frequency into D/A converters. If these converters are on Delta Tau-provided accessories, the DAC_CLK setting should be left at the default value.
The ADC_CLK controls the serial data frequency from A/D converters. If these converters are on Delta Tau-provided accessories, the ADC_CLK setting should be left at the default value.
To determine the clock frequencies set by a given value of MI903, use the following procedure:

1. Divide MI903 by 512 and round down to the nearest integer. This value N1 is the ADC_CLK divider.
2. Multiply N1 by 512 and subtract the product from MI903 to get MI903'. Divide MI903' by 64 and round down to the nearest integer. This value N2 is the DAC_CLK divider.
3. Multiply N2 by 64 and subtract the product from MI903' to get MI903". Divide MI903" by 8 and round down to the nearest integer. This value N3 is the PFM_CLK divider.
4. Multiply N3 by 8 and subtract the product from MI903". The resulting value N4 is the SCLK divider.

Examples:
The maximum encoder count frequency in the application is 800 kHz, so the 1.2288 MHz SCLK frequency is chosen. A pulse train up to 500 kHz needs to be generated, so the 2.4576 MHz PFM_CLK frequency is chosen. The default serial DACs and ADCs provided by Delta Tau are used, so the default DAC_CLK frequency of 4.9152 MHz and the default ADC_CLK frequency of 2.4576 MHz are chosen. From the table:

SCLK Divider N: 5
PFM_CLK Divider N: 4
DAC_CLK Divider N: 3
ADC_CLK Divider N: 4
MI903 = 5 + (8 * 4) + (64 * 3) + (512 * 4) = 5 + 32 + 192 + 2048 = 2277

MI903 has been set to 3429. What clock frequencies does this set?

N1 = INT (3429/512) = 6	ADC_CLK = 611.44 kHz
MI903' = 3429 - (512*6) = 357	
N2 = INT (357/64) = 5	DAC_CLK = 1.2288 MHz
MI903" = 357 - (64*5) = 37	
N3 = INT (37/8) = 4	PFM_CLK = 2.4576 MHz
N4 = 37 - (8*4) = 5	SCLK = 1.2288 MHz

See Also: MI907, MI993

MS{anynode},MI904 PWM 1-4 Deadtime / PFM 1-4 Pulse Width Control

Range: 0 - 255
Units: PWM Deadtime = [16 / PWM_CLK (MHz)] * MI904 = 0.135 μsec * MI904
PFM Pulse Width = [1 / PFM_CLK (MHz)] * MI904
= PFM_CLK_period (μsec) * MI904
Default: 15
PWM Deadtime = 0.135 μsec * 15 = 2.03 μsec

PFM Pulse Width = [1 / 9.8304 MHz] * 15 = 1.526 μsec (with default MI903)

MI904 controls the deadtime period between top and bottom on-times in the MACRO Station's automatic PWM generation for machine interface channels 1-4. In conjunction with MI903, it also controls the pulse width for the MACRO Station's automatic pulse-frequency modulation generation for machine interface channels 1-4.

The PWM deadtime, which is the delay between the top signal turning off and the bottom signal turning on, and vice versa, is specified in units of 16 PWM_CLK cycles. This means that the deadtime can be specified in increments of 0.135 μsec. The equation for MI904 as a function of PWM deadtime is:

MI904 = Deadtime (μsec) / 0.135 μsec

The PFM pulse width is specified in PFM_CLK cycles, as defined by MI903. The equation for MI904 as a function of PFM pulse width and PFM_CLK frequency is:

MI904 = PFM_CLK Freq (MHz) / PFM pulse width (μsec)

In PFM pulse generation, the minimum off time between pulses is equal to the pulse width. This means that the maximum PFM output frequency is

PFM Max_Freq (MHz) = PFM_CLK Freq / (2 * MI904)

Examples:

A PWM deadtime of approximately 1 microsecond is desired:

MI904 ≅ 1 μsec / 0.135 μsec ≅ 7

With a 2.4576 MHz PFM_CLK frequency, a pulse width of 0.4 μsec is desired:

MI904 ≅ 2.4576 MHz * 0.4 μsec ≅ 1

See Also:

MI908, MI994

MS{anynode},MI905 DAC 1-4 Strobe Word

Range: $000000 - $FFFFFF
Units: Serial Data Stream (MSB first, starting on rising edge of phase clock)
Default: $7FFF00 (for 16-bit DAC data)

MI905 controls the DAC strobe signal for machine interface channels 1-4, present on a 4-axis piggyback board with E1 connecting pins 1 and 2. The 24-bit word set by MI905 is shifted out serially on lines DAC_STROB1-4, MSB first, one bit per DAC_CLK cycle starting on the rising edge of the phase clock. The value in the LSB is held until the next phase clock cycle.

ACC-2E stack axis-interface boards have 16-bit DACs installed on-board with Option 1 and accessed through the PMAC1-style 96-pin DIN connector; MI905 should be set to $7FFF00 for these boards. The DACs on ACC-8E analog breakout boards that can be interfaced to ACC-2E boards through the PMAC2-style 100-pin high-density connectors have 18-bit DACs; MI905 should be set to $7FFFC0 for these. ACC-24E2A backplane analog axis-interface boards have 18-bit DACs; MI905 should be set to $7FFFC0 for these boards.

See also: MI909, MI999

MS{anynode},MI906 PWM 5-8 Frequency Control

Range: 0 - 32767
Units: PWM Frequency = 117,964.8 kHz / [4*MI906+6]
Default: 6257

PWM Frequency = 117,964.8 / 26114 = 4.5163 kHz

MI906 controls the PWM frequency for machine interface channels 5-8. It does this by setting the limits of the PWM up-down counter, which increments and decrements at the PWMCLK frequency of 117,964.8 kHz (117.9648 MHz).

The PWM frequency determines the actual switching frequency of amplifiers connected to any of the MACRO Station's first four machine interface channels with the direct PWM command. The value of MI906 is only important if the direct PWM command signal format is used on channels 5 to 8.
Generally, MI906 is set to the same value as MI992. If a different PWM frequency is desired for channels 5 to 8, MI906 should be set so that it is an odd-integer multiple (e.g. 3x, 5x, 7x) of MI992, or that MI992 is an odd-integer multiple of MI906. This will keep the PWM hardware on channels 5-8 in synchronization with the software algorithms driven by the PHASE clock.
To set MI906 for a desired PWM frequency, the following formula can be used:

MI906 = (117,964.8 kHz / [4*PWM Freq (kHz)]) - 1 (rounded down)

Example:

A 30 kHz PWM frequency is desired for Channels 5-8:

MI906 = (117,964.8 / [4 * 30]) - 1 = 982

See Also:

MI900, MI992

MS{anynode},MI907 Hardware Clock Control Channels 5-8

Range: 0 - 4095
Units: MI907 = Encoder SCLK Divider
+ 8 * PFM_CLK Divider
+ 64 * DAC_CLK Divider
+ 512 * ADC_CLK Divider

where:
Encoder SCLK Frequency = 39.3216 MHz / (2 ^ Encoder SCLK Divider)
PFM_CLK Frequency = 39.3216 MHz / (2 ^ PFM_CLK Divider)
DAC_CLK Frequency = 39.3216 MHz / (2 ^ DAC_CLK Divider)
ADC_CLK Frequency = 39.3216 MHz / (2 ^ ADC_CLK Divider)

Default: 2258 = 2 + (8 * 2) + (64 * 3) + (512 * 4)
Encoder SCLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz
PFM_CLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz
DAC_CLK Frequency = 39.3216 MHz / (2 ^ 3) = 4.9152 MHz
ADC_CLK Frequency = 39.3216 MHz / (2 ^ 4) = 2.4576 MHz

MI907 controls the frequency of four hardware clock frequencies for the second group of four machine interface channels on the MACRO Station (channels 5-8). It is a 12-bit variable consisting of four independent 3-bit controls, one for each of the clocks. Each of these clock frequencies can be divided down from a starting 39.3216 MHz frequency by powers of 2, from 1 to 128 times. This means that the possible frequency settings for each of these clocks are:

Frequency	Divide by	Divider N in 1/2^N
39.3216 MHz	1	0
19.6608 MHz	2	1
9.8304 MHz	4	2
4.9152 MHz	8	3
2.4576 MHz	16	4
1.2288 MHz	32	5
611.44 kHz	64	6
305.72 kHz	128	7

Very few MACRO Station users will be required to change the setting of MI907 from the default value

The encoder sample clock signal SCLK controls how often the MACRO Station's digital hardware looks at the encoder and flag inputs. The MACRO Station can take at most one count per SCLK cycle, so the SCLK frequency is the absolute maximum encoder count frequency. SCLK also controls the signal propagation through the digital delay filters for the encoders and flags; the lower the SCLK frequency, the greater the noise pulse that can be filtered out. The SCLK frequency should optimally be set to the lowest value that can accept encoder counts at the maximum possible rate.

The pulse-frequency-modulation clock PFM_CLK controls the PFM circuitry that is commonly used for stepper drives. The maximum pulse frequency possible is 1/4 of the PFM_CLK frequency. The PFM_CLK frequency should optimally be set to the lowest value that can generate pulses at the maximum frequency required.

The DAC_CLK controls the serial data frequency into D/A converters. If these converters are on Delta Tau-provided accessories, the DAC_CLK setting should be left at the default value.

The ADC_CLK controls the serial data frequency from A/D converters. If these converters are on Delta Tau-provided accessories, the ADC_CLK setting should be left at the default value.

Example: See MI903 Example

See Also:

MI903, MI993

MS{anynode},MI908 PWM 5-8 Deadtime / PFM 5-8 Pulse Width Control

Range: 0 - 255

Units: PWM Deadtime = 0.135 μsec * MI908
PFM Pulse Width = [1 / PFM_CLK (MHz)] * MI908
= PFM_CLK_period (μsec) * MI908

Default: 15
PWM Deadtime = 0.135 μsec * 15 = 2.03 μsec
PFM Pulse Width = [1 / 9.8304 MHz] * 15 = 1.526 μsec (with default MI907)

MI908 controls the deadtime period between top and bottom on-times in the MACRO Station's automatic PWM generation for machine interface channels 5-8. In conjunction with MI907, it also controls the pulse width for the MACRO Station's automatic pulse-frequency modulation generation for machine interface channels 5-8.

The PWM deadtime, which is the delay between the top signal turning off and the bottom signal turning on, and vice versa, is specified in units of 16 PWM_CLK cycles. This means that the deadtime can be specified in increments of 0.135 μsec.

The PFM pulse width is specified in PFM_CLK cycles, as defined by MI907.

In PFM pulse generation, the minimum off time between pulses is equal to the pulse width. This means that the maximum PFM output frequency is

PFM Max_Freq (MHz) = PFM_CLK Freq / (2 * MI908)

Example: See MI904 Example.
See Also: MI904, MI994

MS{anynode},MI909 DAC 5-8 Strobe Word

Range: $000000 - $FFFFFF
Units: Serial Data Stream (MSB first, starting on rising edge of phase clock)
Default: $7FFF00 (for 16-bit DAC data)

MI909 controls the DAC strobe signal for machine interface channels 5-8, present on a 4-axis board with jumper E1 connecting pins 2 and 3. The 24-bit word set by MI909 is shifted out serially on the DAC_STROB lines, MSB first, one bit per DAC_CLK cycle starting on the rising edge of the phase clock. The value in the LSB is held until the next phase clock cycle.
ACC-2E stack axis-interface boards have 16-bit DACs installed on-board with Option 1 and accessed through the PMAC1-style 96-pin DIN connector; MI909 should be set to $7FFF00 for these boards. The DACs on ACC-8E analog breakout boards that can be interfaced to ACC-2E boards through the PMAC2-style 100-pin high-density connectors have 18-bit DACs; MI909 should be set to $7FFFC0 for these. ACC-24E2A backplane analog axis-interface boards have 18-bit DACs; MI909 should be set to $7FFFC0 for these boards.
See Also: MI905, MI999

Node-Specific Gate Array MI-variables

MI-variables MI910 through MI919 on the MACRO station control the hardware setup of the hardware interface channel on the station associated a MACRO node. The matching of hardware interface channels to MACRO nodes is determined by the setting of the SW1 rotary switch on the CPU/Interface Board of the MACRO station.
These variables are accessed using the “MS” station auxiliary read and write commands. The number immediately after the “MS” specifies the node number, and therefore the channel number mapped to that node by the SW1 setting.

MS{node},MI910 Encoder/Timer n Decode Control

Range: 0 - 15
Units: None
Default: 7

MI910 controls how the input signal for the encoder mapped to the specified node is decoded into counts. As such, this defines the sign and magnitude of a "count". The following settings may be used to decode an input signal.

0: Pulse and direction CW
1: x1 quadrature decode CW
2: x2 quadrature decode CW
3: x4 quadrature decode CW
4: Pulse and direction CCW
5: x1 quadrature decode CCW
6: x2 quadrature decode CCW
7: x4 quadrature decode CCW
8: Internal pulse and direction
9: Not used
10: Not used
11: x6 hall format decode CW
12: MLDT pulse timer control
(internal pulse resets timer; external pulse latches timer)
13: Not used
14: Not used
15: x6 hall format decode CCW

In any of the quadrature decode modes, PMAC is expecting two input waveforms on CHAn and CHBn, each with approximately 50% duty cycle, and approximately one-quarter of a cycle out of phase with each other. "Times-one" (x1) decode provides one count per cycle; x2 provides two counts per cycle; and x4 provides four counts per cycle. The vast majority of users select x4 decode to get maximum resolution.

The "clockwise" (CW) and "counterclockwise" (CCW) options simply control which direction counts up. If you get the wrong direction sense, simply change to the other option (e.g. from 7 to 3 or vice versa).

Note:

If you change the direction sense of an encoder with a properly working servo without also changing the direction sense of the output, you can get destabilizing positive feedback to your servo and a dangerous runaway condition.

In the pulse-and-direction decode modes, PMAC is expecting the pulse train on CHAn, and the direction (sign) signal on CHBn. If the signal is unidirectional, the CHBn line can be allowed to pull up to a high state, or it can be hardwired to a high or low state.

If MI910 is set to 8, the decoder inputs the pulse and direction signal generated by Channel n's pulse frequency modulator (PFM) output circuitry. This permits the MACRO Station to create a phantom closed loop when driving an open-loop stepper system. *No jumpers or cables are needed to do this; the connection is entirely within the ASIC.* The counter polarity automatically matches the PFM output polarity.

If MI910 is set to 12, the timer circuitry is set up to read magnetostrictive linear displacement transducers (MLDTs) such as Temposonics™. In this mode, the timer is cleared when the PFM

circuitry sends out the excitation pulse to the sensor on PULSEn, and it is latched into the memory-mapped register when the excitation pulse is received on CHAn.
If MI910 is set to 11 or 15, the channel is set up to accept 3-phase “hall-effect” style inputs on the A, B, and C inputs, decoding 6 states per cycle.

MS{node},MI911 Position Compare n Channel Select

Range: 0 - 1
Units: None
Default: 0
0: Use channel n encoder counter for position compare function
1: Use first encoder counter on IC (encoder 1 for channels 1 to 4;
encoder 5 for channels 5 to 8) for position compare function

MI911 determines which encoder input that the position compare circuitry for the machine interface channel mapped to the specified node uses.
When MI911 is set to 0, the channel's position compare register is tied to the channel's own encoder counter, and the position compare signal appears only on the EQUn output.
When MI911 is set to 1, the channel's position compare register is tied to the first encoder counter on the ASIC -- Encoder 1 for channels 1-4, Encoder 5 for channels 5-8, or Encoder 9 for channels 9-10 -- and the position compare signal appears both on EQUn, and combined into the EQU output for the first channel on the IC (EQU1 or EQU5); executed as a logical OR.
MI911 for the first channel on an ASIC performs no effective function, so is always 1. It cannot be set to 0.

MS{node},MI912 Encoder n Capture Control

Range: 0 - 15
Units: none
Default: 1

This parameter determines which signal or combination of signals, and which polarity, triggers a position capture of the counter for the encoder mapped to the specified node. If a flag input (home, limit, or user) is used, MI913 for the node determines which flag. Proper setup of this variable is essential for a successful home search, which depends on the position-capture function. The following settings may be used:

0: Immediate capture
1: Capture on Index (CHCn) high
2: Capture on Flag high
3: Capture on (Index high AND Flag high)
4: Immediate capture
5: Capture on Index (CHCn) low
6: Capture on Flag high
7: Capture on (Index low AND Flag high)
8: Immediate capture
9: Capture on Index (CHCn) high
10: Capture on Flag low
11: Capture on (Index high AND Flag low)
12: Immediate capture
13: Capture on Index (CHCn) low
14: Capture on Flag low
15: Capture on (Index low AND Flag low)

The trigger is armed when the position capture register is read. After this, as soon as the MACRO Station sees that the specified input lines are in the specified states, the trigger will occur -- it is level-trigger, not edge-triggered.

MS{node},MI913 Capture n Flag Select Control

Range: 0 - 3
Units: none
Default: 0

This parameter determines which of the "Flag" inputs will be used for position capture (if one is used -- see MI912):

0: HMFLn (Home Flag n)
1: PLIMn (Positive End Limit Flag n)
2: MLIMn (Negative End Limit Flag n)
3: USERn (User Flag n)

This parameter is typically set to 0 or 3, because in actual use, the LIMn flags create other effects that usually interfere with what is trying to be accomplished by the position capture. If you wish to capture on the LIMn flags, you probably will want to disable their normal functions with Ix25, or use a channel n where none of the flags is used for the normal axis functions.

MS{node},MI914 Encoder n Gated Index Select

Range: 0 - 1
Units: none
0 = Use ungated index for encoder position capture
1 = Use index gated by quadrature channels for position capture
Default: 0

When MI914 is set to 0, the index channel input (CHCn) for the encoder mapped to the specified MACRO node is passed directly into the position capture circuitry.
When MI914 is set to 1, the encoder index channel input (CHCn) is logically combined with ("gated by") the quadrature signals of Encoder n before going to the position capture circuitry. The intent is to get a "gated index" signal exactly one quadrature state wide. This provides a more accurate and repeatable capture, and makes the use of the capture function to confirm the proper number of counts per revolution very straightforward.
In order for the gated index capture to work reliably, the index pulse must reliably span one, but only one, "high-high" or "low-low" AB quadrature state of the encoder. MI915 allows you to select which of these two possibilities is used.

MS{node},MI915 Encoder n Index Gate State

Range: 0 - 1
Units: none
0 = Gate index with "high-high" quadrature state (GI = A & B & C)
1 = Gate index with "low-low" quadrature state (GI = A/ & B/ & C)
Default: 0

When using the "gated index" feature of the MACRO Station for more accurate position capture (see MI914), MI915 specifies whether the raw index-channel signal for the encoder mapped to the specified MACRO node is passed through to the position capture signal only on the "high-high" quadrature state, or only on the "low-low" quadrature state. If MI915 is set to 0, it is passed through only on the "high-high" state; if MI915 is set to 1, it is passed through only on the "low-low" state.

MS{node},MI916 Output n Mode Select

Range:	0 - 3
Units:	none
	0 = Outputs A & B are PWM; Output C is PWM
	1 = Outputs A & B are DAC; Output C is PWM
	2 = Outputs A & B are PWM; Output C is PFM
	3 = Outputs A & B are DAC; Output C is PFM
Default:	3

MI916 controls what output formats are used on the command output signal lines for machine interface channel n. If a three-phase direct PWM command format is desired, MI916 should be set to 0. If signal outputs for (external) digital-to-analog converters are desired, MI916 should be set to 1 or 3. In this case, the C output can be used as a supplemental (non-servo) output in either PWM or PFM form. For example, it can be used to excite an MLDT sensor (e.g. Temposonics™) in PFM form.

MS{node},MI917 Output n Invert Control

Range: 0 - 3
Units: none
0 = Do not invert Outputs A & B; Do not invert Output C
1 = Invert Outputs A & B; Do not invert Output C
2 = Do not invert Outputs A & B; Invert Output C
3 = Invert Outputs A & B; Invert Output C
Default: 0

MI917 controls the polarity of the command output signals for Channel n. The default non-inverted outputs are high true. For PWM signals on Outputs A, B, and C, this means that the transistor-on signal is high. Delta Tau PWM-input amplifiers, and most other PWM-input amplifiers, expect this non-inverted output format. For such a 3-phase motor drive, MI917 should be set to 0.

For PFM signals on Output C, non-inverted means that the pulse-on signal is high (direction polarity is controlled by MI918). During a change of direction, the direction bit will change synchronously with the leading edge of the pulse, which in the non-inverted form is the rising edge. If the drive requires a set-up time on the direction line before the rising edge of the pulse, the pulse output can be inverted so that the rising edge is the trailing edge, and the pulse width (established by MI904 or MI908) is the set-up time.

For DAC signals on Outputs A and B, non-inverted means that a 1 value to the DAC is high. DACs used on Delta Tau accessory boards, as well as all other known DACs always expect non-inverted inputs, so MI917 should always be set to 0 or 2 when using DACs on Channel n.

MS{node},MI918 Output n PFM Direction Signal Invert Control

Range: 0 - 1
Units: none
0 = Do not invert direction signal (+ = low; - = high)
1 = Invert direction signal (- = low; + = high)
Default: 0

MI918 controls the polarity of the direction output signal in the pulse-and-direction format for Channel n. It is only active if MI916 has been set to 2 or 3 to use Output C as a pulse-frequency-modulated (PFM) output.

If MI918 is set to the default value of 0, a positive direction command provides a low output; if MI918 is set to 1, a positive direction command provides a high output.

MS{node},MI919 Reserved for future use

MS{node},MI920 Absolute Power-On Position (Read Only)

Range: $0 -$FFFFFFFFFFFF
Units: counts
Default:

This variable, when queried, reports the value of the absolute position for the specified MACRO node. MI11x for the motor node determines what type of feedback device at what address will be read when this variable is queried.

When the value of MI920 is queried, the encoder counter for the channel matched to the specified node is cleared (when the otherwise similar MI930 is queried, the counter is not cleared.)

MS{node},MI921 Flag Capture Position (Read Only)

Range: $0 - $FFFFFF
Units: counts
Default:

This variable, when queried, reports the value of the captured position for the machine interface channel mapped to the specified MACRO node by SW1. Refer to the Motor command/status flag registers for their relationship to this value.

MS{node},MI922 ADC A Input Value (Read Only)

Range: $000000 - $FFFFFF
Units: Bits of a 24-bit ADC

MI922 reports the value of the serial ADC input register A for the machine interface channel mapped to the specified MACRO node number. The value is reported as a 24-bit number, even though there are a maximum of 18 real bits in the register (the most significant bits) and existing hardware provides 12 or 16 bits of true input.

MS{node},MI923 Compare Auto-Increment Value

Range: -8,388,608 - 8,388,607
Units: Encoder counts
Default: 0

MI923 specifies the value of the position-compare auto-increment register for the machine interface channel mapped to the specified MACRO node number.

MS{node},MI924 ADC B Input Value (Read Only)

Range: -8,388,608 - 8,388,607
Units: Bits of a 24-bit ADC

MI924 reports the value of the serial ADC input register B for the machine interface channel mapped to the specified MACRO node number. The value is reported as a 24-bit number, even though there are a maximum of 18 real bits in the register (the most significant bits) and existing hardware provides 12 or 16 bits of true input.

MS{node},MI925 Compare A Position Value

Range: -8,388,608 - 8,388,607
Units: Encoder counts
Default: 0

MI925 specifies the value of the 'A' compare register of the position compare function for the machine interface channel mapped to the specified MACRO node number. The units are encoder counts, referenced to the position at the latest power-on or reset.

MS{node},MI926 Compare B Position Value

Range: -8,388,608 - 8,388,607
Units: Encoder counts

Default: 0

MI926 specifies the value of the 'B' compare register of the position compare function for the machine interface channel mapped to the specified MACRO node number. The units are encoder counts, referenced to the position at the latest power-on or reset.

MS{node},MI927 Encoder Loss Status Bit

Range: 0 – 1
Units: none
Default: 0

MI927 reports whether the Servo IC on the MACRO Station has detected loss of a differential encoder signal for the machine interface channel mapped to the specified MACRO node number. It is a single-bit variable that reports 0 if no loss has been detected, or a 1 if a loss has been detected. It will still report a value of 1 after a loss has been detected, even if the signal has been recovered, until a zero value has been written to MI927 to clear the bit.

For this bit to work properly, the A+, A-, B+, and B- encoder inputs must also be wired into the T, U, V, and W flags for the channel. Also, the resistor pack for the encoder channel must be reversed from the standard configuration so that pin 1 of the pack (marked with a dot) is at the opposite end from pin 1 of the socket (marked with a bold outline and square solder pin).

The shutdown function on encoder loss will work as long as the resistor pack has been reversed from factory default. However, proper reporting of the exactly where the loss occurred requires double wiring of the encoder into the flags so MI927 can detect the loss.

This encoder-loss status bit for each channel is copied into bit 8 of the flag status word of the matching MACRO node for reporting back to PMAC if MI16 for the MACRO Station is set to 1. If the T, U, V, and W flags are used for other purposes, such as Hall commutation sensors, or analog-encoder sub-count data, the status of MI927 should be ignored.

MS{node},MI928 Compare-State Write Enable

Range: 0 – 1
Units: none
Default: 0

When MI928 is set to 1, the value of MI929 if forced onto the position-compare output for the channel associated with the specified node. MI928 is automatically reset to 0 immediately after this occurs.

MS{node},MI929 Compare-Output Initial State

Range: 0 – 1
Units: none
Default: 0

The value of MI929 is forced onto the position-compare output for the channel associated with the specified node when MI928 is set to 1. After this, each time the channel's encoder-counter position matches the value of MI925 or MI926, the output state is toggled.

MS{node},MI930 Absolute Power-On Position (Read Only)

Range: 0 – $FFFFFFFFFFFF
Units: counts
Default:

This variable, when queried, reports the value of the absolute position for the specified MACRO node. MI11x for the motor node determines what type of feedback device at what address will be read when this variable is queried.
When the value of MI930 is queried, the encoder counter for the channel matched to the specified node is not cleared (when the otherwise similar MI920 is queried, the counter is cleared.)

MS{node},MI931-MI939 (Reserved for future use)

MS{anynode},MI940 ADC1-4 Strobe Word

Range: $000000 - $FFFFFF
Units: Individual Bits
Default: $FFFFFE

MI940 specifies the strobe word for the serial A/D converters connected to the first 4-axis interface board (ACC-2E or ACC-24E2). The bits of the strobe word are shifted out, one bit per ADC_CLK cycle, MSB first, starting on the rising edge of the phase clock. The default value is suitable both for current-feedback ADCs on ACC-8K boards or in most direct PWM amplifiers, and for ACC-28B general-purpose ADCs.

MS{anynode},MI941 ADC5-8 Strobe Word

Range: $000000 - $FFFFFF
Units: Individual Bits
Default: $FFFFFE

MI941 specifies the strobe word for the serial A/D converters connected to the second 4-axis interface board (ACC-2E or ACC-24E2). The bits of the strobe word are shifted out, one bit per ADC_CLK cycle, MSB first, starting on the rising edge of the phase clock. The default value is suitable both for current-feedback ADCs on ACC-8K boards or in most direct PWM amplifiers, and for ACC-28B general-purpose ADCs.

MS{anynode},MI942 ADC9-10 Strobe Word

Range: $000000 - $FFFFFF
Units: Individual Bits
Default: $FFFFFE

MI942 specifies the strobe word for the serial A/D converters interfaced to the ACC-1E 2-axis piggyback board through the PMAC2-style connectors. The bits of the strobe word are shifted out, one bit per ADC_CLK cycle, MSB first, starting on the rising edge of the phase clock. The default value is suitable both for current-feedback ADCs on ACC-8K boards or in most direct PWM amplifiers, and for ACC-28B general-purpose ADCs.

MS{anynode},MI943-MI973 (Reserved for future use)

MS{anynode},MI1974 Station Display Status (Read Only)

Range: $0 - $F
Units: none

This variable, when queried, reports the hexadecimal digit displayed on the MACRO Station's 7-segment display. The meaning of each digit is:

0: No motors enabled on Station
1: 1 motor enabled on Station
2: 2 motors enabled on Station
3: 3 motors enabled on Station
4: 4 motors enabled on Station
5: 5 motors enabled on Station
6: 6 motors enabled on Station
7: 7 motors enabled on Station
8: 8 motors enabled on Station
9: (reserved for future use)
A: Amplifier fault
B: Ring-break fault
C: Configuration change fault
D: Ring data-error fault
E: Loss-of-encoder fault
F: Other fault

Note

If the display itself is blank, this indicates that ring communications are not active, which means that this value cannot be reported back to the controller.

MS{anynode},MI975 I/O Node Enable

Range: $0000 - $FFFF
Units: none (individual bits)
Default: $0000

MI975 permits the enabling of MACRO I/O nodes on the MACRO Station. MI975 is a 16-bit value – bits 0 to 15 – with bit *n* controlling the enabling of MACRO node *n*. If the bit is set to 0, the node is disabled; if the bit is set to 1, the node is enabled. The I/O nodes on the MACRO Station are nodes 2, 3, 6, 7, 10, and 11, which can be enabled by MI975 bits of these numbers. Only bits 2, 3, 6, 7, 10, and 11 of MI975 should ever be set to 1.

MI975 is used at the power-on/reset of the MACRO Station in combination with rotary switch SW1 and MI976 to determine which MACRO nodes are to be enabled. The net result can be read in Station variable MI996. To get a value of MI975 to take effect, the value must be saved (**MSSAVE{node}**) and the Station reset (**MS$$${node}**)

Examples:

```
MS0,MI975=$4        ; Enable I/O Node 2 alone
MS0,MI975=$C        ; Enable I/O Nodes 2 & 3
MS0,MI975=$4C       ; Enable I/O Nodes 2, 3, & 6
MS0,MI975=$CC       ; Enable I/O Nodes 2, 3, 6, & 7
MS0,MI975=$4CC      ; Enable I/O Nodes 2, 3, 6, 7, & 10
MS0,MI975=$CCC      ; Enable I/O Nodes 2, 3, 6, 7, 10, & 11
MS4,MI975=$40       ; Enable I/O Node 6 alone
```

```
MS4,MI975=$C0          ; Enable I/O Nodes 6 & 7
MS8,MI975=$400         ; Enable I/O Node 10 alone
MS8,MI975=$C00         ; Enable I/O Nodes 10 & 11
```

MS{anynode},MI976 Motor Node Disable

Range: $0000 - $FFFF
Units: none (individual bits)
Default: $0000

MI976 permits the disabling of MACRO motor nodes that would be enabled by the setting of rotary switch SW1. MI976 is a 16-bit value – bits 0 to 15 – with bit *n* controlling the disabling of MACRO node *n*. If the bit is set to 0, the node may be enabled by SW1; if the bit is set to 1, the node is disabled, regardless of the setting of SW1. The motor nodes on the MACRO Station are nodes 0, 1, 4, 5, 8, 9, 12, & 13, which can be disabled by MI976 bits of these numbers. Only bits 0, 1, 4, 5, 8, 9, 12, & 13 of MI975 should ever be set to 1.

MI976 is used at the power-on/reset of the MACRO Station in combination with rotary switch SW1 and MI975 to determine which MACRO nodes are to be enabled. The net result can be read in Station variable MI996. To get a value of MI976 to take effect, the value must be saved (**MSSAVE{node}**) and the Station reset (**MS$$${node}**)

Examples:

```
MS0,MI976=$2           ; Disable Motor Node 1
MS0,MI976=$20          ; Disable Motor Node 5
MS0,MI976=$30          ; Disable Motor Nodes 4 & 5
MS8,MI976=$200         ; Disable Motor Node 9
MS12,MI976=$2000       ; Disable Motor Node 13
```

MS{anynode},MI977 Motor Nodes Reporting Ring Break

Range: $0000 - $FFFF
Units: none (individual bits)
Default: $0

MI977 permits the MACRO Station to enable additional motor nodes if it detects a ring break immediately upstream from it, and send out the "ring break" bit (Bit 13) in the flag word for these nodes. When the Station detects a ring break, it turns itself into a ring master, and sets the "ring break" bit on all active nodes. In this manner, other stations downstream of the break can be directly notified of the ring break, so they can shut down properly.

MI977 is a 16-bit value – bits 0 to 15 – with bit *n* controlling the enabling of MACRO node *n* on a ring break If the bit is set to 0, the node will not be enabled on a ring break; if the bit is set to 1, the node will be enabled on a ring break. The motor nodes on the MACRO Station are nodes 0, 1, 4, 5, 8, 9, 12, & 13, which can be enable on ring break by MI977 bits of these numbers. Only bits 0, 1, 4, 5, 8, 9, 12, & 13 of MI975 should ever be set to 1.

Examples:

```
MS0,MI977=$3300        ; Enable Motor Nodes 8, 9, 12, & 13 on ring break
MS8,MI977=$0033        ; Enable Motor Nodes 0, 1, 4, & 5 on ring break
```

MS{anynode},MI978-MI979 (Reserved for future use)

Direct I/O Control MI-Variables – Multi-Node Stations (V1.xxx)

Variables MI980 – MI986 have different meanings depending on whether they are used in a multi-node Station with Version 1.xxx firmware, or a single-node Station with V2.xxx firmware. This section documents their meanings for multi-node Stations; the next section documents their use for single-node Stations.

MS{anynode},MI980 JTHW Port A Data

Range: $00 - $FF
Units: Bits
Default: $00

On a multi-node Station, MI980 specifies the 8-bit read/write data for "Port A" of the JTHW connector (J6) on the CPU/Interface Board of the MACRO Station. This data is brought out on pins SEL0 - SEL7 of the JTHW connector. MI980 can be used to read from or write to this port when the connector is not used for multiplexed I/O.

MS{anynode},MI981 JTHW Port A Direction Control

Range: 0 – 1
Units: none
Default: 1

On a multi-node Station, MI981 specifies the direction for the 8 bits of I/O on "Port A" of the JTHW connector (J6) on the CPU/Interface Board of the MACRO Station. This data is brought out on pins SEL0 - SEL7 of the JTHW connector. A value of 0 specifies all inputs; a value of 1 specifies all outputs.
MI981 is not used until MI983 is set to 1. MI981 must be in its default setting for any multiplexed I/O accessories to be used on the port.

MS{anynode},MI982 JTHW Port A Inversion Control

Range: $00 - $FF
Units: Bits
Default: $00

On a multi-node Station, MI982 specifies the inversion for the 8 bits of I/O on "Port A" " of the JTHW connector (J6) on the CPU/Interface Board of the MACRO Station. This data is brought out on pins SEL0 - SEL7 of the JTHW connector. The inversion of each I/O point is individually controllable by the corresponding bit of MI982; a bit value of 0 specifies non-inverting (high true); a bit value of 1 specifies inverting (low true).
MI982 is not used until MI983 is set to 1. MI982 must be in its default setting for any multiplexed I/O accessories to be used on the port.

MS{anynode},MI983 JTHW Direct I/O Enable

Range: 0 – 1
Units: none
Default: 0

On a multi-node Station, MI983 applies the MI981, MI982, MI985, and MI986 direction and inversion control MI-variables when it is set to 1. After applying these variables, the MACRO Station automatically sets MI983 back to 0.

MS{anynode},MI984 JTHW Port B Data

Range: $00 - $FF
Units: Bits
Default: $00

On a multi-node Station, MI984 specifies the 8-bit read/write data for "Port B" of the JTHW connector (J6) on the CPU/Interface Board of the MACRO Station. This data is brought out on pins DAT0 - DAT7 of the JTHW connector. MI984 can be used to read from or write to this port when the connector is not used for multiplexed I/O.

MS{anynode},MI985 JTHW Port B Direction Control

Range: 0 – 1
Units: none
Default: 1

On a multi-node Station, MI985 specifies the direction for the 8 bits of I/O on "Port B" of the JTHW connector (J6) on the CPU/Interface Board of the MACRO Station. This data is brought out on pins DAT0 - DAT7 of the JTHW connector. A value of 0 specifies all inputs; a value of 1 specifies all outputs.
MI985 is not used until MI983 is set to 1. MI985 must be in its default setting for any multiplexed I/O accessories to be used on the port.

MS{anynode},MI986 JTHW Port B Inversion Control

Range: $00 - $FF
Units: Bits
Default: $00

On a multi-node Station, MI986 specifies the inversion for the 8 bits of I/O on "Port B" of the JTHW connector (J6) on the CPU/Interface Board of the MACRO Station. This data is brought out on pins DAT0 - DAT7 of the JTHW connector. The inversion of each I/O point is individually controllable by the corresponding bit of MI986; a bit value of 0 specifies non-inverting (high true); a bit value of 1 specifies inverting (low true).
MI986 is not used until MI983 is set to 1. MI986 must be in its default setting for any multiplexed I/O accessories to be used on the port.

Direct I/O Control MI-Variables – Single-Node Stations (V2.xxx)

Variables MI980 – MI986 have different meanings depending on whether they are used in a multi-node Station with Version 1.xxx firmware, or a single-node Station with V2.xxx firmware. This section documents their meanings for single-node Stations; the previous section documents their use for multi-node Stations.

MS{anynode},MI980 IO24-27 Input State (Read-Only)

Range: $0 - $F
Units: Bits
Default:

On a single-node Station, MI980 reports the 4 bits of input data read on I/O points 24 – 27 of the MACRO IC, mapped into Bits 0 – 3, respectively, of Y:$C081.

MS{anynode},MI981 (Reserved for future use)

MS{anynode},MI982 IO24-27 Input Inversion Control

Range: $0 – $F
Units: Bits
Default: $0

On a single-node Station, MI982 specifies the inversion for the 4 bits of input read on I/O points 24 – 27 of the MACRO IC, which can be read with MI980. The inversion of each input is individually controllable by the corresponding bit of MI982; a bit value of 0 specifies non-inverting (high true); a bit value of 1 specifies inverting (low true).
MI982 is only used at Station power-up/reset, or when MI983 is set (momentarily) to 1.

MS{anynode},MI983 IO24-31 Initialize/Set

Range: 0 - 1
Units: none
Default: 1 -> 0

On a single-node Station, MI983 applies the, MI982 and MI986 inversion-control MI-variables, and sets the MI984 outputs equal to the MI985 value when it is set to 1. After applying these variables, the MACRO Station automatically sets MI983 back to 0. These actions are automatically executed at the Station power-up/reset.

MS{anynode},MI984 IO28-31 Output Status

Range: $0 – $F
Units: Bits
Default: (from MI985)

On a single-node Station, MI984 reports the present values of the outputs on I/O points 28 – 31 of the MACRO IC, mapped into bits 4 – 7, respectively, of Y:$C081.

MS{anynode},MI985 IO28-31 Output Commands

Range: $0 – $F
Units: Bits
Default: 0

On a single-node Station, MI985 specifies the values that will be forced onto the outputs on I/O points 28 – 31 of the MACRO IC, mapped into bits 4 – 7, respectively of Y:$C081, when MI983 is set to 1.

MS{anynode},MI986 IO28-31 Output Inversion Control

Range: $00 - $FF
Units: Bits
Default: $00

On a single-node Station, MI986 specifies the inversion for the 4 bits of output read on I/O points 28 – 31 of the MACRO IC, which can be set with MI985 and MI983, and read with MI984. The inversion of each output is individually controllable by the corresponding bit of MI986; a bit value of 0 specifies non-inverting (high true); a bit value of 1 specifies inverting (low true). MI986 is only used at Station power-up/reset, or when MI983 is set (momentarily) to 1.

A/D Converter Demultiplex Control

MS{anynode},MI987 A/D Input Enable

Range: 0 - 1
Units: none
Default: 0

MI987 controls whether the MACRO Station will read its optional on-board A/D converters (on the ACC-1E 2-Axis or ACC-6E A/D piggyback board). If MI987 is set to 1, the Station will read these A/D converters at a high rate, copying new data every phase cycle into each of the Y-registers $0200 to $0207. If MI987 is set to 0, the Station will ignore the A/D converters, even if they are physically present on the Station.

Note:

If an I/O board set to address $FFC0 (ACC-3E 144-I/O Board with E1 ON, or ACC-4E 48-Isolated I/O Board with E15 ON) is present on the Station even in the absence of the A/D converters, setting MI987 to 1 may interfere with the I/O on the board.

MS{anynode},MI988 A/D Unipolar/Bipolar Control

Range: $00 - $FF
Units: none
Default: $00

MI988 controls whether the optional on-board A/D converters (on the ACC-1E 2-axis or ACC-6E A/D piggyback board) are set up for unipolar (0 to +5V) or bipolar (-2.5 to +2.5V) inputs. MI988 consists of 8 bits; each bit controls the setup of a pair of A/D converters. A value of 0 in the bit sets up the A/D converters for unipolar inputs; a value of 1 in the bits sets up the A/D converters for bipolar inputs.

The following table shows which bits of MI988 control which A/D converters:

MI998 Bit #	Hex Bit Value	1st ADC	2nd ADC
0	1	ANAI00	ANAI08
1	2	ANAI01	ANAI09
2	4	ANAI02	ANAI10
3	8	ANAI03	ANAI11
4	10	ANAI04	ANAI12
5	20	ANAI05	ANAI13
6	40	ANAI06	ANAI14
7	80	ANAI07	ANAI15

MS{anynode},MI989 A/D Source Address

Range: $0000 - $FFFF
Units: Station Y-addresses
Default: $0 (specifies $FFC0)

This variable specifies the source address of the multiplexed A/D converters acted on by the de-multiplexing algorithms of MI987 and MI988. These multiplexed A/D converters can be on ACC-1E or ACC-6E stack boards, or ACC-36E or ACC-59E backplane boards.

The A/D converters on an ACC-1E or ACC-6E stack boards are located at address Y:$FFC0, so MI989 should be set to $FFC0 to use these A/D converters.
The A/D converters on an ACC-36E or ACC-59E backplane board are located at 1 of 4 addresses, depending on the DIP-switch setting of the board. These addresses are Y:$FFE0, Y:$FFE8, Y:$FFF0, and Y:$FFF8.
In station firmware revisions 1.114 and older, this variable was not active (fixed at 0), and the A/D demultiplexing algorithm automatically used register Y:$FFC0. For backward compatibility, a setting of 0 for MI989 specifies the use of Y:$FFC0.

Global & 2-Axis Board I-Variables

MI-Variables numbered in the MI990s control hardware aspects of the "DSPGATE2" ASIC. This IC controls operation of the MACRO ring on all MACRO Stations. This IC also controls the frequency of the clock signals for the 2-axis piggyback board (machine interface channels 9 & 10).

MS{anynode},MI992 MaxPhase and PWM 9-10 Frequency Control

Range: 0 - 32767
Units: MaxPhase Frequency = 117,964.8 kHz / [2*MI992+3]
PWM Frequency = 117,964.8 kHz / [4*MI992+6]
Default: 6527
MaxPhase Frequency = 117,964.8 / 13057 = 9.0346 kHz
PWM Frequency = 117,964.8 / 26114 = 4.5173 kHz

MI992 controls the "maximum phase" clock frequency for the MACRO Station, and the PWM frequency for supplementary machine interface channels 9 and 10. It does this by setting the limits of the PWM up-down counter, which increments and decrements at the PWMCLK frequency of 117,964.8 kHz (117.9648 MHz).
The actual phase clock frequency is divided down from the maximum phase clock according to the setting of MI997. The phase clock frequency must be the same as the ring update frequency as set by the ring controller - usually a PMAC or PMAC2. If the ring controller is a PMAC2 Ultralite, MI992 and MI997 on the MACRO Station should be set to the same values as MI992 and MI997 on the PMAC2 Ultralite. If the ring controller is a PMAC2 that is not "Ultralite", MI992 and MI997 on the MACRO Station should be set to the same values as MI900 and MI901 on the PMAC2. If the ring controller is a PMAC(1), MI992 and MI997 on the MACRO Station should be set to obtain the same frequency as that set on the PMAC(1) by jumpers E98 and E29-E33.
To set MI992 for a desired "maximum phase" clock frequency, the following formula can be used:

MI992 = (117,964.8 kHz / [2*MaxPhase (kHz)]) - 1 (rounded down)

Examples:

To set a PWM frequency of 10 kHz and therefore a MaxPhase clock frequency of 20 kHz:

MI992 = (117,964.8 kHz / [4*10 kHz]) - 1 = 2948

To set a PWM frequency of 7.5 kHz and therefore a MaxPhase clock frequency of 15 kHz:

MI992 = (117,964.8 kHz / [4*7.5 kHz]) - 1 = 3931

MS{anynode},MI993 Hardware Clock Control Channels 9-10

Range: 0 - 4095
Units: MI993 = Encoder SCLK Divider
+ 8 * PFM_CLK Divider

+ 64 * DAC_CLK Divider
+ 512 * ADC_CLK Divider

where:
Encoder SCLK Frequency = 39.3216 MHz / (2 ^ Encoder SCLK Divider)
PFM_CLK Frequency = 39.3216 MHz / (2 ^ PFM_CLK Divider)
DAC_CLK Frequency = 39.3216 MHz / (2 ^ DAC_CLK Divider)
ADC_CLK Frequency = 39.3216 MHz / (2 ^ ADC_CLK Divider)

Default: 2258 = 2 + (8 * 2) + (64 * 3) + (512 * 4)
Encoder SCLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz
PFM_CLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz
DAC_CLK Frequency = 39.3216 MHz / (2 ^ 3) = 4.9152 MHz
ADC_CLK Frequency = 39.3216 MHz / (2 ^ 4) = 2.4576 MHz

MI993 controls the frequency of three hardware clock frequencies -- SCLK, PFM_CLK, DAC_CLK and ADC_CLK -- for the machine interface channels 9 and 10 on the 2-axis piggyback board of a MACRO Station. It is a 12-bit variable consisting of four independent 3-bit controls, one for each of the clocks. Each of these clock frequencies can be divided down from a starting 39.3216 MHz frequency by powers of 2, 2^N, from 1 to 128 times (N=0 to 7). This means that the possible frequency settings for each of these clocks are:

Frequency	Divide by	Divider N in $1/2^N$
39.3216 MHz	1	0
19.6608 MHz	2	1
9.8304 MHz	4	2
4.9152 MHz	8	3
2.4576 MHz	16	4
1.2288 MHz	32	5
611.44 kHz	64	6
305.72 kHz	128	7

Very few MACRO Station users will be required to change the setting of MI993 from the default value.

The encoder sample clock signal SCLK controls how often 2-axis board's digital hardware looks at the encoder inputs. PMAC2 can take at most one count per SCLK cycle, so the SCLK frequency is the absolute maximum encoder count frequency. SCLK also controls the signal propagation through the digital delay filters for the encoders and flags; the lower the SCLK frequency, the greater the noise pulse that can be filtered out. The SCLK frequency should optimally be set to the lowest value that can accept encoder counts at the maximum possible rate.

The pulse-frequency-modulation clock PFM_CLK controls the PFM circuitry on the 2-axis board that can create pulse and direction outputs. The maximum pulse frequency possible is 1/4 of the PFM_CLK frequency. The PFM_CLK frequency should optimally be set to the lowest value that can generate pulses at the maximum frequency required.

The ADC_CLK controls the serial data frequency from A/D converters, either for digital current loop closure, or from an ACC-28B A/D converter board.

The DAC-CLK controls the serial data frequency to D/A converters for the 2-axis board, either the on-board converters that come with Option A, or the external converters on an ACC-8E board.

To determine the clock frequencies set by a given value of MI993, use the following procedure:

1. Divide MI993 by 512 and round down to the nearest integer. This value N1 is the ADC_CLK divider.
2. Multiply N1 by 512 and subtract the product from MI993 to get MI993'. Divide MI993' by 64 and round down to the nearest integer. This value N2 is the DAC_CLK divider (not relevant here).
3. Multiply N2 by 64 and subtract the product from MI993' to get MI993". Divide MI993" by 8 and round down to the nearest integer. This value N3 is the PFM_CLK divider.
4. Multiply N3 by 8 and subtract the product from MI993". The resulting value N4 is the SCLK divider.

Examples:
The maximum encoder count frequency in the application is 800 kHz, so the 1.2288 MHz SCLK frequency is chosen. A pulse train up to 500 kHz needs to be generated, so the 2.4576 MHz PFM_CLK frequency is chosen. ADCs and DACs are not used, so the default DAC_CLK frequency of 4.9152 MHz and the default ADC_CLK frequency of 2.4576 MHz are chosen. From the table:

SCLK Divider N: 5

PFM_CLK Divider N: 4
DAC_CLK Divider N: 3
ADC_CLK Divider N: 4
MI993 = 5 + (8 * 4) + (64 * 3) + (512 * 4) = 5 + 32 + 192 + 2048 = 2277

MI993 has been set to 3429. What clock frequencies does this set?

N1 = INT (3429/512) = 6 ADC_CLK = 611.44 kHz
MI993' = 3429 - (512*6) = 357
N2 = INT (357/64) = 5 DAC_CLK = 1.2288 MHz
MI993" = 357 - (64*5) = 37
N3 = INT (37/8) = 4 PFM_CLK = 2.4576 MHz
N4 = 37 - (8*4) = 5 SCLK = 1.2288 MHz

MS{anynode},MI994 PWM 9-10 Deadtime / PFM 9-10 Pulse Width Control

Range: 0 - 255
Units: PWM Deadtime = [16 / PWM_CLK (MHz)] * MI994 = 0.135 μsec * MI994
PFM Pulse Width = [1 / PFM_CLK (MHz)] * MI994
= PFM_CLK_period (μsec) * MI994
Default: 15
PWM Deadtime = 0.135 μsec * 15 = 2.03 μsec
PFM Pulse Width = [1 / 9.8304 MHz] * 15 = 1.526 μsec (with default MI993)

MI994 controls the deadtime period between top and bottom on-times in the MACRO Station's automatic PWM generation for machine interface channels 9 and 10 on the 2-axis piggyback board. In conjunction with MI993, it also controls the pulse width for PMAC2's automatic pulse-frequency modulation generation for these machine interface channels.
The PWM deadtime, which is the delay between the top signal turning off and the bottom signal turning on, and vice versa, is specified in units of 16 PWM_CLK cycles. This means that the deadtime can be specified in increments of 0.135 μsec. The equation for MI994 as a function of PWM deadtime is:

MI994 = Deadtime (μsec) / 0.135 μsec

The PFM pulse width is specified in PFM_CLK cycles, as defined by MI993. The equation for MI994 as a function of PFM pulse width and PFM_CLK frequency is:

MI994 = PFM_CLK Freq (MHz) / PFM pulse width (μsec)

In PFM pulse generation, the minimum off time between pulses is equal to the pulse width. This means that the maximum PFM output frequency is

PFM Max_Freq (MHz) = PFM_CLK Freq / (2 * MI994)

Examples:
A PWM deadtime of approximately 1 microsecond is desired:

MI994 ≅ 1 μsec / 0.135 μsec ≅ 7

With a 2.4576 MHz PFM_CLK frequency, a pulse width of 0.4 μsec is desired:

MI994 ≅ 2.4576 MHz * 0.4 μsec ≅ 1

MS{anynode},MI995 MACRO Ring Configuration/Status

Range: $0000 - $FFFF (0 - 65,535)
Units: none
Default: $0080

MI995 contains configuration and status bits for MACRO ring operation of the MACRO Station. There are 11 configuration bits and 5 status bits, as follows:

Bit #	Value	Type	Function
0	1($1)	Status	Data Overrun Error (cleared when read)
1	2($2)	Status	Byte Violation Error (cleared when read)
2	4($4)	Status	Packet Parity Error (cleared when read)
3	8($8)	Status	Packet Underrun Error (cleared when read)
4	16($10)	Config	Master Station Enable
5	32($20)	Config	Synchronizing Master Station Enable
6	64($40)	Status	Sync Node Packet Received (cleared when read)
7	128($80)	Config	Sync Node Phase Lock Enable
8	256($100)	Config	Node 8 Master Address Check Disable
9	512($200)	Config	Node 9 Master Address Check Disable
10	1024($400)	Config	Node 10 Master Address Check Disable
11	2048($800)	Config	Node 11 Master Address Check Disable
12	4096($1000)	Config	Node 12 Master Address Check Disable
13	8192($2000)	Config	Node 13 Master Address Check Disable
14	16384($4000)	Config	Node 14 Master Address Check Disable
15	32768($8000)	Config	Node 15 Master Address Check Disable

A MACRO Station is a slave on the ring in all normal operation, so configuration bits 4 and 5 are set to 0. It should synchronize itself to the sync node, so configuration bit 7 should be set to 1. In most applications, it will only accept packets from its own master so bits 8 to 15 are all set to 0. All other bits are status bits that are normally 0. This makes the usual setting of MI995 equal to $0080.

MS{anynode},MI996 MACRO Node Activate Control

Range: $000000 to $FFFFFF (0 to 8,388,607)
Units: none
Default: $0 (all nodes de-activated)

MI996 controls which of the 16 MACRO nodes on the MACRO Station are activated. It also controls the master station number, and the node number of the packet that creates a synchronization signal.

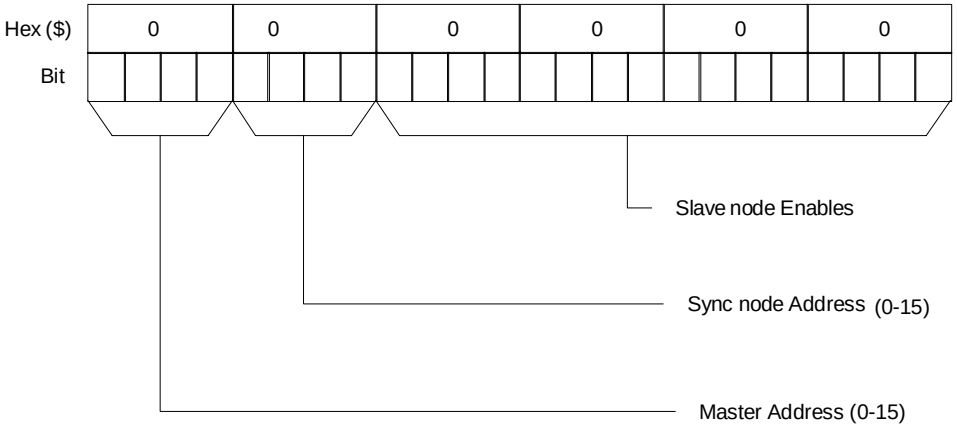
On a power-up or reset of the MACRO Station, MI996 is set automatically by Station firmware as a function of SW1 and SW2 switch settings, plus the saved values of MI975 and MI976.

The bits of MI996 are arranged as follows:

Bit #	Value	Type	Function
0	1($1)	Config	Node 0 Activate
1	2($2)	Config	Node 1 Activate
2	4($4)	Config	Node 2 Activate
3	8($8)	Config	Node 3 Activate
4	16($10)	Config	Node 4 Activate
5	32($20)	Config	Node 5 Activate
6	64($40)	Config	Node 6 Activate
7	128($80)	Config	Node 7 Activate

8	256($100)	Config	Node 8 Activate
9	512($200)	Config	Node 9 Activate
10	1024($400)	Config	Node 10 Activate
11	2048($800)	Config	Node 11 Activate
12	4096($1000)	Config	Node 12 Activate
13	8192($2000)	Config	Node 13 Activate
14	16384($4000)	Config	Node 14 Activate
15	32768($8000)	Config	Node 15 Activate
16-19	$X0000	Config	Packet Sync Node Slave Address (0 - 15)
20-23	$X00000	Config	Master Station Number (0-15)

Bits 0 to 15 are individual control bits for the matching node number 0 to 15. If the bit is set to 1, the node is activated; if the bit is set to 0, the node is de-activated. On power-up reset, these bits are set as defined by the SW1 setting, with some motor nodes possibly disabled by MI976, and some I/O nodes possibly enabled by MI975. Node 15 should always be activated to support the Type 1 auxiliary communications.
Bits 16-19 specify the slave number of the packet which will generate the “sync pulse” on the MACRO Station. This is always set to 15 ($F) on the MACRO Station.
Bits 20-23 specify the master number (0-15) for the MACRO Station. At power-up/reset, these bits get the value set by SW2. The number must be specified whether the card is a master station or a slave station.



MS{anynode},MI997 Phase Clock Frequency Control

Range: 0 - 15
Units: PHASE Clock Frequency = MaxPhase Frequency / (MI997+1)
Default: 0
PHASE Clock Frequency = 9.0346 kHz / 1 = 9.0346 kHz
(with default value of MI992)

MI997, in conjunction with MI992, determines the frequency of the PHASE clock on a MACRO Station. Each cycle of the PHASE clock, a set of MACRO ring information is expected, and any data transfers between MACRO nodes and interface circuitry are performed. The PHASE clock cycle on the MACRO Station should match that of the PMAC commanding it as closely as possible.
Specifically, MI997 controls how many times the PHASE clock frequency is divided down from the "maximum phase" clock, whose frequency is set by MI992. The PHASE clock frequency is

equal to the "maximum phase" clock frequency divided by (MI997+1). MI997 has a range of 0 to 15, so the frequency division can be by a factor of 1 to 16. The equation for MI997 is:

MI997 = (MaxPhase Freq / PHASE Clock Freq) - 1

The ratio of MaxPhase Freq. to PHASE Clock Freq. must be an integer.

Example:

With a 20 kHz MaxPhase Clock frequency established by MI992, and a desired 6.67 kHz PHASE clock frequency, the ratio between MaxPhase and PHASE is 3:

MI997 = (20 / 6.67) - 1 = 3 -1 = 2

MS{anynode},MI998 Servo Clock Frequency Control

Range: 0 - 15
Units: Servo Clock Frequency = PHASE Clock Frequency / (MI998+1)
Default: 0
PHASE Clock Frequency = 9.0346 kHz / (0+1) = 9.0346 kHz
(with default values of MI992 and MI997)

Note:

There is currently no software use of the SERVO clock on the MACRO Station. However, it is needed to capture certain encoder values in the DSPGATEx Servo ICs.

MI998, in conjunction with MI997 and MI992, determines the frequency of the SERVO clock on the MACRO Station.
Specifically, MI998 controls how many times the SERVO clock frequency is divided down from the PHASE clock, whose frequency is set by MI992 and MI997. The SERVO clock frequency is equal to the PHASE clock frequency divided by (MI998+1). MI998 has a range of 0 to 15, so the frequency division can be by a factor of 1 to 16. The equation for MI998 is:

MI998 = (PHASE Clock Freq. / SERVO Clock Freq) - 1

The ratio of PHASE Clock Freq. to SERVO Clock Freq. must be an integer. On the MACRO Station, MI998 should always be set to 0 so the servo clock frequency is equal to the phase clock frequency.

MS{anynode},MI999 DAC 9-10 Strobe Word

Range: $000000 - $FFFFFF
Units: Serial Data Stream (MSB first, starting on rising edge of phase clock)
Default: $7FFF00 (for 16-bit DAC data)

MI999 controls the DAC strobe signal for machine interface channels 9-10, the channels on the 2-axis piggyback board. The 24-bit word set by MI999 is shifted out serially on the DAC_STROB lines for these channels, MSB first, one bit per DAC_CLK cycle starting on the rising edge of the phase clock. The value in the LSB is held until the next phase clock cycle.
The default MI999 value of $7FFF00 is proper for the 16-bit DACs installed on the ACC-1E board with Option 1 accessed through the PMAC1-style 96-pin DIN connector. A value of $7FFFC0 is suitable for the 18-bit DACs on the ACC-8E Analog Interface Board connected through the PMAC2-style 100-pin connector. MI909 should not be changed from the default unless different DACs are used.
See Also: MI905, MI999

MACRO STATION SERIAL COMMANDS

The MACRO Station can accept ASCII text commands directly through the serial port at connector J7 on the CPU/Interface Board, or in auxiliary mode from a Turbo PMAC over the MACRO ring using MACSTASCII commands. Serial communications is at 9600 baud (CPU board jumper E3 connecting pins 1 & 2) or 38400 baud (E3 connecting pins 2 & 3), 8 bits, 1 stop bit, no parity. These commands are intended for basic setup and troubleshooting. Most users will not utilize this port, instead sending commands only through the MACRO ring.
The following commands can be sent to the MACRO Station through the serial port or over the MACRO ring.

$$$ Station Reset

The **$$$** command will reset the MACRO Station and restore all station MI-variables to their last saved values.

*$$$*** Station Re-initialize*

The **$$$*** ** command will reset the MACRO Station and restore all station MI-variables to their factory default values.

CHN Report Channel Number

The **CHN** command causes the MACRO Station to report its present channel number.

CID Report Card ID Number

The **CID** command causes the MACRO Station CPU to report its part number: 602804.

CLRF Clear Station Faults

The **CLRF** command will clear all faults on the MACRO Station and prepare it for further operation.

DATE Report Firmware Date

The **DATE** command causes the MACRO Station to report the date of its firmware.
Example:

```
DATE
07/10/97
```

MI{constant} Report Station MI-Variable Value

The **MI{constant}** command causes the MACRO Station to report the current value of the specified MI-variable.

MI{constant}={constant} Set Station MI-Variable Value

The **MI{constant}={constant}** command causes the MACRO Station to set the value of the specified MI-variable to the specified value.

R{address} Read Station Address

The **R[H]{address}[,{count}]** command causes the MACRO Station to report the value stored at the specified address[es]. If **H** is used, the contents of the register[s] are reported back in hexadecimal; otherwise, they are reported back in decimal form **{address}** consists of a register type (X, Y, L, or P), and the numerical address of the register. The optional {count} value specifies the number of registers to be reported, starting at the specified address and counting up. If no **{count}** value is specified in the command, one register value is reported.

Examples:

```
RX:$20              ; Read X register $20
64                  ; CMS responds in decimal
RHX:$20             ; Read X register $20 in hex
40                  ; CMS responds in hex
RHY:$FFC0,3         ; Read Y registers $FFC0, $FFC1, $FFC2
FFFFA4 FFFF01 FFFFC7  ; CMS responds in hex
```

SAVE Save Station MI-variables

The **SAVE** command causes the MACRO Station to copy its MI-variable values from volatile active memory to the non-volatile flash memory. On the next power-up or reset, these values will be copied back from flash memory to active memory.

VERS Report Firmware Version

The **VERS** command causes the MACRO Station to report its firmware version number.

Example:

```
VERS
1.106
```

VID Report Vendor ID Number

The **VID** command causes the MACRO Station to report its vendor identification number: for Delta Tau, this number is '1'.

PMAC TYPE 1 MACRO STATION COMMANDS

The following commands from the PMAC and Turbo PMAC controllers can be used for Type 1 auxiliary communication with the MACRO Station. These commands require PMAC firmware version V1.16C or newer.

On-Line Commands

MS Command

Syntax: **MS{command}{node #}**

where:

{command} is one of the following text strings:

$$$	normal station reset
$$$* **	station reset and re-initialize
CLRF	station fault clear for
CONFIG	report station configuration value
DATE	report station firmware date
SAVE	save station setup
VER	report station firmware version

{node #} is a constant representing the number of the node to be commanded (if the command affects the entire station, the number of any active node on the station may be used)

This PMAC command causes PMAC to issue the specified command to a Type 1 MACRO slave station.

The **MS CONFIG** command allows the user to set and report a user-specified configuration value. This provides any easy way for the user to see if the MACRO station has already been configured to the user's specifications. The factory default configuration value is 0. It is recommended that after the user finishes the software configuration of the station, a special number be given to the configuration value with the **MS CONFIG{node #}={constant}** command. This number will be saved to the non-volatile memory with the **MS SAVE** command. Subsequently, when the system is powered up, the station can be polled with the **MS CONFIG {node #}** command. If the expected value is returned, the station can be assumed to have the proper software setup. If the expected value is not returned (for instance, when a replacement station has just been installed) then the setup will have to be transmitted to the station.

Examples:

```
MS $$$0          ; Resets MACRO station which has active node 0
MS $$$***4       ; Reinitializes MACRO station which has active node 4
MS CLRF8         ; Clears fault on Node 8 of MACRO station
MS CONFIG12      ; Causes MACRO station to report its configuration number
37               ; PMAC reports MACRO station configuration number to host
MS CONFIG12=37   ; Sets MACRO station configuration number
MS DATE 0        ; Causes MACRO station to report its firmware date
03/27/97         ; PMAC reports MACRO station firmware date to host
```

```
MS SAVE 4          ; Causes MACRO station to save setup variables
MS VER 8           ; Causes MACRO station to report its firmware version
1.104              ; PMAC reports MACRO station firmware version to host
```

MS Variable Read

Syntax: **MACROSLAVE{node #},{slave MI-variable}**
MS{node #},{slave MI-variable}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be read (if the variable is not node-specific, the number of any active node on the station may be used)

{slave MI-variable} is the name of the MI-variable on the slave station whose value is to be reported

This command causes PMAC to query the MACRO slave station at the specified node # and report back the value of the specified slave station MI-variable to the host computer.

Examples:

```
MS0,MI910   ; Causes MACRO station to report value of Node 0 variable MI910
7           ; PMAC reports this value back to host
MS1,MI997   ; Causes MACRO station to report value global variable MI997
6258        ; PMAC reports this value back to host
```

MS Variable Write

Syntax: **MACROSLAVE{node #},{slave variable}={constant}**
MS{node #},{slave variable}={constant}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be written to (if the variable is not node-specific, the number of any active node on the station may be use)

{slave variable} is the name of the MI-variable or C-command on the slave station whose value is to be set

{constant} is a number representing the value to be written to the specified MI-variable

This command causes PMAC to write the specified constant value to the MACRO slave station MI-variable, or if a C-command is specified, it causes the station to execute the specified command number (in which case the constant value does not matter).

The valid C-commands are:

C1	Clear station faults
C2	Reset station, loading saved station MI-variables
C3	Re-initialize station, loading default station MI-variables
C4	Save station MI-variables to non-volatile memory

Examples:

```
MS0,MI910=7        ;sets Node 0 variable MI910 to 7
MS8,C4=0           ; Clears faults on MACRO station with active node 8
```

MS Variable Read Copy

Syntax: **MACROSLVREAD{node #},{slave MI-variable},{PMAC variable}**
MSR{node #},{slave MI-variable},{PMAC variable}

where

{node #} is a constant (0-14) representing the number of the node whose variable is to be read (if the variable is not node-specific, the number of any active node on the station may be used)

{slave MI-variable} is the name of the MI-variable on the slave station whose value is to be reported

{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied

This command copies the value of the specified MI-variable on the MACRO slave station into the specified variable on PMAC.
The MI-variable on the MACRO slave station can be global to the station, or node-specific.
The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variable on the card.
If this command is issued to a PMAC while a PLC buffer is open, it will be stored in the buffer as a PLC command, not executed as an on-line command.

Examples:

```
MS0,MI910,P1        ; Copies value of MACRO station Node 0 variable MI910 into
                    ; PMAC variable P1

MS1,MI997,M10       ; Copies value of MACRO station global variable MI997 into
                    ; PMAC variable M10
```

MS Variable Write Copy

Syntax: **MACROSLVWRITE{node #},{slave variable},{PMAC variable}**
MSW{node #},{slave MI-variable},{PMAC variable}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be read (if the variable is not node-specific, the number of any active node on the station may be used)

{slave variable} is the name of the MI-variable or C-command on the slave station whose value is to be reported

{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied

This command copies the value of the specified variable on PMAC into the specified MI-variable on the MACRO slave station, or if a slave station C-command number is specified, it executes that command (in which case the PMAC variable value is not really used).
The valid C-commands are:

C1	Clear station faults
C2	Reset station, loading saved station MI-variables
C3	Re-initialize station, loading default station MI-variables
C4	Save station MI-variables to non-volatile memory

The MI-variable on the MACRO slave station can be global to the station, or node-specific.
The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variable on the card.
If this command is issued to a PMAC while a PLC buffer is open, it will be stored in the buffer as a PLC command, not executed as an on-line command.

Examples:

```
MSW0,MI910,P35      ; Copies value of PMAC P35 into MACRO station
                    ; node 0 variable MI910
MSW4,C4,P0          ; Causes MACRO station with active node 4 to save its
                    ; MI-variable values to non-volatile memory
                    ; (P0 is a dummy variable here)
```

PMAC PLC Commands for Type 1 MACRO Stations

MS Variable Read Copy

Syntax: **`MACROSLVREAD{node #},{slave MI-variable},{PMAC variable}`**
`MSR{node #},{slave MI-variable},{PMAC variable}`

where

`{node #}` is a constant (0-14) representing the number of the node whose variable is to be read (if the variable is not node-specific, the number of any active node on the station may be used)

`{slave MI-variable}` is the name of the MI-variable on the slave station whose value is to be reported

`{PMAC variable}` is the name of the variable on the PMAC into which the value of the slave station variable is to be copied

This command copies the value of the specified MI-variable on the MACRO slave station into the specified variable on PMAC.
The MI-variable on the MACRO slave station can be global to the station, or node-specific.
The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variable on the card.
If this command is issued to a PMAC while no PLC buffer is open, it will be executed as an on-line command, not stored in the buffer as a PLC command.

Examples:

```
MS0,MI910,P1      ; Copies value of MACRO station Node 0 variable MI910 into
                  ; PMAC variable P1

MS1,MI997,M10     ; Copies value of MACRO station global variable MI997 into
                  ; PMAC variable M10
```

MS Variable Write Copy

Syntax: **`MACROSLVWRITE{node #},{slave variable},{PMAC variable}`**
`MSW{node #},{slave MI-variable},{PMAC variable}`

where:

`{node #}` is a constant (0-14) representing the number of the node whose variable is to be read (if the variable is not node-specific, the number of any active node on the station may be used)

`{slave variable}` is the name of the MI-variable or C-command on the slave station whose value is to be reported

{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied

This command copies the value of the specified variable on PMAC into the specified MI-variable on the MACRO slave station, or if a slave station C-command number is specified, it executes that command (in which case the PMAC variable value is not really used).

The valid C-commands are:

- **C1** Clear station faults
- **C2** Reset station, loading saved station MI-variables
- **C3** Re-initialize station, loading default station MI-variables
- **C4** Save station MI-variables to non-volatile memory

The MI-variable on the MACRO slave station can be global to the station, or node-specific.

The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variable on the card.

If this command is issued to a PMAC while no PLC buffer is open, it will be executed as an on-line command, not stored in the buffer as a PLC command.

Examples:

```
MSW0,MI910,P35    ; Copies value of PMAC P35 into MACRO station
                  ; node 0 variable MI910
MSW4,C4,P0        ; Causes MACRO station with active node 4 to save its
                  ; MI-variable values to non-volatile memory
                  ; (P0 is a dummy variable here)
```

MACRO STATION MEMORY AND I/O MAP

In the listing below, the hexadecimal address is listed first, followed by the decimal address in parentheses.

Global Servo Calculation Registers

`X/Y:$0000-$000F`

Encoder Conversion (Interpolation) Table

`X:$0010-$002F` (1824-1855)		Converted encoder and time base data
`Y:$0010-$002F`		Encoder conversion source and format

The format of the conversion table is:

	Bits	
`Y:`word	16-23	Conversion format: `$00` = 1/T incremental encoder `$10` = A/D register conversion `$20` = Unfiltered parallel Y word source* `$30` = Filtered parallel Y word source** `$40` = Time base* `$50` = Integrated A/D register conversion* `$60` = Unfiltered parallel X word source* `$70` = Filtered parallel X word source** `$80` = parallel interp. of incremental `$90` = Triggered time base; frozen* `$A0` = Triggered time base; running* `$B0` = Triggered time base; armed* `$C0` = no interp. of incremental encoder `$D0` = Exponential filter** `$E0` = Sum or difference of two entries
	0-15	Address of source data

* Next Y word contains user-set constant for conversion (this is a double-entry conversion).
** Next two Y words contain user-set constants for conversion (this is a triple-entry conversion).

X:word	0-4	Fractional bits of converted data
	5-23	Integer bits of converted data (if last entry in conversion) Intermediate value if not last entry in conversion

Refer to the detailed description of the encoder conversion table under "Feedback Features."

DSPGATE1 Registers

Note:

The MACRO Station can support with its automatic servo functions up to 8 servo interface channels on 2 4-channel DSPGATE1 ICs. Two ACC-2E

boards with DSPGATE1 ICs can be installed on the "stack", and two ACC-24E2 or 24E2A boards with DSPGATE1 ICs can be installed on the "backplane". Registers on boards not used by automatic servo functions can be used with Station I/O copying operations.

- Stack channels 1-4 are present on an ACC-2E 4-axis piggyback board with jumper E1 connecting pins 1 & 2.
- Stack channels 5-8 are present on an ACC-2E 4-axis piggyback board with jumper E1 connecting pins 2 & 3.
- Backplane channels 1-4 are present on an ACC-24E2x 2/4-axis backplane board with SW1-1 and –2 closed (ON).
- Backplane channels 5-8 are present on an ACC-24E2x 2/4-axis backplane board with SW1-1 and –2 open (OFF).

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C000**	**$C008**	**$C010**	**$C018**	**$C020**	**$C028**	**$C030**	**$C038**
Backplane Address	**$C040**	**$C048**	**$C050**	**$C058**	**$C060**	**$C068**	**$C070**	**$C078**

`Y:$Cxxx` Channel n Time between last two encoder counts (SCLK cycles)
`X:$Cxxx` Channel n Status Word

Bits:
- 0-2 Capture Hall Effect Device State
- 3 Invalid demultiplex of C, U, V, and W
- 4-7 Not used (reports as 0)
- 8 Encoder Count Error (0 on counter reset, 1 on illegal transition) (MS{node},MI927)
- 9 Position Compare (EQUn) output value
- 10 Position-Captured-On-Gated-Index Flag (=0 on read of captured position register, =1 on trigger capture)
- 11 Position-Captured Flag (on any trigger) (=0 on read of captured position register, =1 on trigger capture)
- 12 Encoder Channel A (CHAn) Input Value
- 13 Encoder Channel B (CHBn) Input Value
- 14 Encoder Channel C (Index, CHCn) Input Value (ungated)
- 15 Amplifier Fault (FAULTn) Input Value
- 16 Home Flag (HMFLn) Input Value
- 17 Positive End Limit (PLIMn) Input Value
- 18 Negative End Limit (MLIMn) Input Value
- 19 User Flag (USERn) Input Value
- 20 FlagWn Input Value
- 21 FlagVn Input Value
- 22 FlagUn Input Value
- 23 FlagTn Input Value

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C001**	**$C009**	**$C011**	**$C019**	**$C021**	**$C029**	**$C031**	**$C039**
Backplane Address	**$C041**	**$C049**	**$C051**	**$C059**	**$C061**	**$C069**	**$C071**	**$C079**

`Y:$Cxxx` Channel n Time since last encoder count (SCLK cycles)
`X:$Cxxx` Channel n Encoder phase position (counts)

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C002**	**$C00A**	**$C012**	**$C01A**	**$C022**	**$C02A**	**$C032**	**$C03A**
Backplane Address	**$C042**	**$C04A**	**$C052**	**$C05A**	**$C062**	**$C06A**	**$C072**	**$C07A**

Y:$Cxxx Channel n Output A Command Value
Bits: 8-23: PWM Command Value
6-23: Serial DAC Command Value
0-5: Not Used
X:$Cxxx Channel n Encoder Servo Position Capture Register
Bits: 0: Direction of last count (0=up, 1=down)
1-23: Position counter (units of counts)

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C003**	**$C00B**	**$C013**	**$C01B**	**$C023**	**$C02B**	**$C033**	**$C03B**
Backplane Address	**$C043**	**$C04B**	**$C053**	**$C05B**	**$C063**	**$C06B**	**$C073**	**$C07B**

Y:$Cxxx Channel n Output B Command Value
Bits: 8-23: PWM Command Value
6-23: Serial DAC Command Value
0-5: Not used
X:$Cxxx Channel n Flag Position Capture Value; 24 bits, in counts (MS{node}, MI921)

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C004**	**$C00C**	**$C014**	**$C01C**	**$C024**	**$C02C**	**$C034**	**$C03C**
Backplane Address	**$C044**	**$C04C**	**$C054**	**$C05C**	**$C064**	**$C06C**	**$C074**	**$C07C**

Y:$Cxxx Channel n Output C Command Value
Bits: 8-23: PWM Command Value
0-23: PFM Command Value
X:$Cxxx IC Global Control Word
Stack Channel 1: X:$C004; Stack Channel 5: X:$C024;
Backplane Channel 1: X:$C044; Backplane Channel 5: X:$C064:
Clock Control Word
(X:$C004 controls stack channels 1-4; X:$C024 controls stack channels 5-8)
(X:$C044 controls backplane channels 1-4; X:$C064 controls backplane channels 5-8)
Bits (X:$C004 bits 0-11 is I903; X:$C024 bits 0-11 is I907)
0-2: SCLK Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
3-5: PFM Clock Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
6-8: DAC Clock Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
9-11: ADC Clock Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
12: Phase Clock Direction (0=output, 1=input)
(This must be 0 in X:$C004; 1 in X:$C024--if 2nd ASIC is used)
13: Servo Clock Direction (0=output, 1=input)
(This must be 0 in X:$C004; 1 in X:$C024--if 2nd ASIC is used)
14-15: Reserved for future use (report as zero)

(X:$C004 bits 16-19 is I901)
16-19: Phase Clock Frequency Control n (f=MAXPHASE / [n+1], n=0-15)
(value in X:$C024 not used)
(X:$C004 bits 20-23 is I902)
20-23: Servo Clock Frequency Control n (f=PHASE / [n+1], n=0--15)
(value in X:$C024 not used)

Stack Channel 2: X:$C00C; Stack Channel 6: X:$C02C;
Backplane Channel 2: X:$C04C; Backplane Channel 6: X:$C06C:
DAC Strobe Word, 24 bits
(X:$C00C controls stack channels 1-4;
X:$C02C controls stack channels 5-8)
(X:$C04C controls backplane channels 1-4;
X:$C06C controls backplane channels 5-8)

(Shifted out MSB first one bit per DAC_CLK cycle, starting on rising edge of phase clock)
Stack Channel 3: X:$C014; Stack Channel 7: X:$C034;
Backplane Channel 11: X:$C054; Backplane Channel 15: X:$C074:
ADC Strobe Word, 24 bits
(X:$C014 controls stack channels 1-4; X:$C034 controls stack channels 5-8)
(X:$C054 controls backplane channels 1-4; X:$C074 controls backplane channels 5-8)
(Shifted out MSB first one bit per ADC_CLK cycle, starting on rising edge of phase clock)
Stack Channel 4: X:$C01C; Stack Channel 8: X:$C03C;
Backplane Channel 4: X:$C01C; BackplaneChannel 8: X:$C03C:
PWM, PFM, MaxPhase Control Word
(X:$C01C controls stack channels 1-4; X:$C03C controls stack channels 5-8)
(X:$C05C controls backplane channels 1-4; X:$C07C controls backplane channels 5-8)
(X:$C01C bits 0-7 is I904; X:$C03C bits 0-7 is I908)
Bits: 0-7: PWM Dead Time (16*PWM CLK cycles)
also PFM pulse width (PFM CLK cycles)
(X:$C01C bits 8-23 is I900; X:$C03C bits 8-23 is I906)
8-23: PWM MaxCount Value
PWM Frequency = 117.9648 MHz / [4*MaxCount + 6]
"MaxPhase" Frequency = 2*PWM Frequency
= 117.9648 MHz / [2*MaxCount + 3]

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C005**	**$C00D**	**$C015**	**$C01D**	**$C025**	**$C02D**	**$C035**	**$C03D**
Backplane Address	**$C045**	**$C04D**	**$C055**	**$C05D**	**$C065**	**$C06D**	**$C075**	**$C07D**

`Y:$Cxxx` Channel n ADC A Input Value (MS{node},MI922}
Bits: 6-23: Serial ADC Value
0-5: Not used
`X:$Cxxx` Channel n Control Word
(Bits 0-3: MS{node},MI910)
Bits 0-1: Encoder Decode Control
00: Pulse and direction decode
01: x1 quadrature decode

10: x2 quadrature decode
11: x4 quadrature decode

2-3: Direction & Timer Control
00: Standard timer control, external signal source, no inversion
01: Standard timer control, external signal source, invert direction
10: Standard timer control, internal PFM source, no inversion
11: Alternate timer control, external signal source

4-5: Position Capture Control (MS{node},MI912)
00: Software capture (by setting bit 6)
01: Use encoder index alone
10: Use capture flag alone
11: Use encoder index and capture flag

6: Index Capture Invert Control (0=no inversion, 1=inversion)

7: Flag Capture Invert Control (0=no inversion, 1=inversion)

8-9: Capture Flag Select Control (MS{node},MI913)
00: Home Flag (HMFLn)
01: Positive End Limit (PLIMn)
10: Negative End Limit (MLIMn)
11: User Flag (USERn)

10: Encoder Counter Reset Control (1=reset)

11: Position Compare Initial State Write Enable (MS{node},MI928)

12: Position Compare Initial State Value (MS{node},MI929)

13: Position Compare Channel Select (MS{node},MI911)
(0 = use this channel's encoder; 1 = use first encoder on IC)

14: AENAn output value

15: Gated Index Select for Position Capture (MS{node},MI914)
(0=ungated index, 1=gated index)

16: Invert AB for Gated Index (MS{node},MI915)
(0: Gated Signal=A&B&C; 1: Gated Signal=A/&B/&C)

17: Index channel demultiplex control (0=no demux, 1=demux)

18: Reserved for future use (reports as 0)

19: Invert PFM Direction Control (0=no inversion, 1=invert)
(MS{node},MI918)
(Bits 20-21: MS{node},MI917)

20: Invert A & B Output Control (0=no inversion, 1=invert)

21: Invert C Output Control (0=no inversion, 1=invert)
(Bits 22-23: MS{node},MI916)

22: Output A & B Mode Select (0=PWM, 1=DAC)

23: Output C Mode Select (0=PWM, 1=PFM)

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C006**	**$C00E**	**$C016**	**$C01E**	**$C026**	**$C02E**	**$C036**	**$C03E**
Backplane Address	**$C046**	**$C04E**	**$C056**	**$C05E**	**$C066**	**$C06E**	**$C076**	**$C07E**

Y:$Cxxx Channel n ADC B Input Value (MS{node},MI924)
Bits: 6-23: Serial ADC Value
0-5: Not used

X:$Cxxx Channel n Encoder Compare Auto-increment value (24 bits, units of counts)
(MS{node},MI923)

Channel #	1	2	3	4	5	6	7	8
Stack Address	**$C007**	**$C00F**	**$C017**	**$C01F**	**$C027**	**$C02F**	**$C037**	**$C03F**
Backplane Address	**$C047**	**$C04F**	**$C057**	**$C05F**	**$C067**	**$C06F**	**$C077**	**$C07F**

`Y:$Cxxx` Channel n Encoder Compare A Value (24 bits, units of counts)
(MS{node},MI925)

`X:$Cxxx` Channel n Encoder Compare B Value (24 bits, units of counts)
(MS{node},MI926)

DSPGATE2 Registers

`Y:$C080` General I/O Data Register

Note

The pins associated with this register are used for other purposes on the MACRO Station.

Bits: 0 I/O00 Data Value
...
23 I/O23 Data Value

`X:$C080` General I/O Data Direction Control

Note

The pins associated with this register are used for other purposes on the MACRO Station.

Bits: 0 I/O00 Direction Control
...
23 I/O23 Direction Control
(All bits: 0=Input; 1=Output)

`Y:$C081` General I/O Data Register
Bits: 0 I/O24 (SEL0 pin) Data Value
...
7 I/O31 (SEL7 pin) Data Value
8 I/O24 Latched Data Value
...
15 I/O31 Latched Data Value
16-23 Not used

`X:$C081` General I/O Direction Control
Bits: 0 I/O24 (SEL0 pin) Direction Control
...
7 I/O31 (SEL7 pin) Direction Control
(All bits: 0=Input; 1=Output)
8-23 Not used

`Y:$C082` General I/O Data Register

Note:

The pins associated with this register are used for other purposes on the MACRO Station.

Bits:
0 DAT0 Data Value
...
7 DAT7 Data Value
8 SEL0 Data Value
...
15 SEL7 Data Value
16-23 Not used

`X:$C082` General I/O Data Direction Control Register

Note:

The pins associated with this register are used for other purposes on the MACRO Station.

Bits:
0 DAT0 Direction Control
...
7 DAT7 Direction Control
8 SEL0 Direction Control
...
15 SEL7 Direction Control
(All bits: 0=Input; 1=Output)
16-23 Not used

`Y:$C083` General I/O Port Data Register

Bits:
0 DISP0 (DAT0 pin) Data Value
...
7 DISP7 (DAT7 pin) Data Value
8 CTRL0 Data Value
...
11 CTRL3 Data Value
12-23 Not used

`X:$C083` General I/O Port Data Direction Control Register

Bits:
0 DISP0 Direction Control
...
7 DISP7 Direction Control
8 CTRL0 Direction Control
...
11 CTRL3 Direction Control (Must be 1 for ??? to function)
(All bits: 0=Input; 1=Output)
12-23 Not used

`Y:$C084` Data Type Control Register

Bits:
0 I/O00 Data Type Control (0=FlagW9; 1=I/O00)
1 I/O01 Data Type Control (0=FlagV9; 1=I/O01)
2 I/O02 Data Type Control (0=FlagU9; 1=I/O02)
3 I/O03 Data Type Control (0=FlagT9; 1=I/O03)
4 I/O04 Data Type Control (0=USER9; 1=I/O04)

5 I/O05 Data Type Control (0=MLIM9; 1=I/O05)
6 I/O06 Data Type Control (0=PLIM9; 1=I/O06)
7 I/O07 Data Type Control (0=HMFL9; 1=I/O07)
8 I/O08 Data Type Control (0=PWM_B_BOT9; 1=I/O08)
9 I/O09 Data Type Control (0=PWM_B_TOP9; 1=I/O09)
10 I/O10 Data Type Control (0=PWM_A_BOT9; 1=I/O10)
11 I/O11 Data Type Control (0=PWM_A_TOP9; 1=I/O11)
12 I/O12 Data Type Control (0=PWM_B_BOT10; 1=I/O12)
13 I/O13 Data Type Control (0=PWM_B_TOP10; 1=I/O13)
14 I/O14 Data Type Control (0=PWM_A_BOT10; 1=I/O14)
15 I/O15 Data Type Control (0=PWM_A_TOP10; 1=I/O15)
16 I/O16 Data Type Control (0=HMFL10; 1=I/O16)
17 I/O17 Data Type Control (0=PLIM10; 1=I/O17)
18 I/O18 Data Type Control (0=MLIM10; 1=I/O18)
19 I/O19 Data Type Control (0=USER10; 1=I/O19)
20 I/O20 Data Type Control (0=FlagT10; 1=I/O20)
21 I/O21 Data Type Control (0=FlagU10; 1=I/O21)
22 I/O22 Data Type Control (0=FlagV10; 1=I/O22)
23 I/O23 Data Type Control (0=FlagW10; 1=I/O23)
(All bits: 0=dedicated hardware I/O; 1=general I/O)
(All bits must be 0 for use with ACC-1E 2-axis piggyback board)

`X:$C084` Data Inversion Control Register (when used as general I/O; see Y:$C084)

Bits:
0 I/O00 Inversion Control
...
23 I/O23 Inversion Control
(All bits: 0=Non-inverting; 1=Inverting)

`Y:$C085` General I/O Data Type Control Register

Bits:
0 I/O24 Data Type Control
...
7 I/O31 Data Type Control
(These bits are always 1; there is no alternate mode for these lines)
8-23 Not used

`X:$C085` General I/O Data Inversion Control

Bits:
0 I/O24 Inversion Control
...
7 I/O31 Inversion Control
(All bits: 0=Non-inverting; 1=Inverting)
8-23 Not used

`Y:$C086` Data Type Control Register

Bits:
0 DAT0 Data Type Control (0=ENCC9; 1 =DAT0)
1 DAT1 Data Type Control (0=ENCC10; 1 =DAT1)
2 DAT2 Data Type Control (0=Fault9; 1 =DAT2)
3 DAT3 Data Type Control (0=Fault10; 1 =DAT3)
4 DAT4 Data Type Control (0=EQU9; 1 =DAT4)
5 DAT5 Data Type Control (0=EQU10; 1 =DAT5)
6 DAT6 Data Type Control (0=AENA9; 1 =DAT6)
7 DAT7 Data Type Control (0=AENA10; 1 =DAT7)
8 SEL0 Data Type Control (0=ADC_STROB; 1=SEL0)
9 SEL1 Data Type Control (0=ADC_CLK; 1=SEL1)

10 SEL2 Data Type Control (0=ADC_A9; 1=SEL2)
11 SEL3 Data Type Control (0=ADC_B9; 1=SEL3)
12 SEL4 Data Type Control (0=ADC_A10; 1=SEL4)
13 SEL5 Data Type Control (0=ADC_B10; 1=SEL5)
14 SEL6 Data Type Control (0=SCLK; 1=SEL6)
15 SEL7 Data Type Control (0=SCLK_DIR*; 1=SEL7)
(All bits: 0=dedicated hardware I/O; 1=general I/O)
(All bits must be 0 for use with ACC-1E 2-axis piggyback board)
16-23 Not used

`X:$C086` JTHW Port Data Inversion Control Register (when used as general I/O; see Y:$C086)

Bits:
0 DAT0 Inversion Control
...
7 DAT7 Inversion Control
8 SEL0 Inversion Control
...
15 SEL7 Inversion Control
(All bits: 0=Non-inverting; 1=Inverting)
(All bits must be 0 to use standard port accessories)
16-23 Not used

`Y:$C087` Data Type Control Register

Bits:
0 DISP0 Data Type Control
...
7 DISP7 Data Type Control
8 CTRL0 Data Type Control
...
11 CTRL3 Data Type Control
(These bits are always 1; there is no alternate mode for these pins)
12-23 Not used

`X:$C087` Data Inversion Control Register

Bits:
0 DISP0 Inversion Control
...
7 DISP7 Inversion Control
8 CTRL0 Inversion Control
...
11 CTRL3 Inversion Control
(All bits: 0=Non-inverting; 1=Inverting)
(All bits must be 0 to use standard port accessories)
12-23 Not used

`Y:$C088-$C08B` Not used

`X:$C088-$C08B` Not used

`Y:$C08C` Pure binary conversion from gray code input on I/O00 to I/O23

Note:

The pins associated with this register are used for other purposes on the MACRO Station.

`X:$C08C` DAC Strobe Word, 24 bits
(Shifted out MSB first, one bit per DACCLK cycle, starting on rising edge of phase clock)

`Y:$C08D` Gray-to-binary conversion bit-length control

Note:

The pins associated with this register are used for other purposes on the MACRO Station.

Bits: 0-3 Bit length of less significant word portion (I/O00 - I/Onn)
4 =1 specifies 16-bit lower / 8-bit upper conversion
5-23 Not used

`X:$C08D` Not used

`Y:$C08E` MACRO Node Enable Control (I996)

Bits: 0 Node 0 enable control
...
15 Node 15 enable control
(0=node disable; 1=node enable)
16-19 Sync packet slave node number control
20-23 Master number control

`X:$C08E` Not used

`Y:$C08F` MACRO Ring Status and Control

Bits: 0 Data overrun error (cleared when read)
1 Byte violation error (cleared when read)
2 Packet parity error (cleared when read)
3 Data underrun error (cleared when read)
4 Master station enable
5 Synchronizing master station enable
6 Sync packet received (cleared when read)
7 Sync packet phase lock enable
8 Node 8 master address check disable
9 Node 9 master address check disable
10 Node 10 master address check disable
11 Node 11 master address check disable
12 Node 12 master address check disable
13 Node 13 master address check disable
14 Node 14 master address check disable
15 Node 15 master address check disable

`X:$C08F` DSPGATE2 clock control register

Bits (Bits 0-11 comprise I993)

0-2: SCLK Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
3-5: PFM Clock Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
6-8: DAC Clock Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
9-11: ADC Clock Frequency Control n (f=39.3216MHz / 2^n, n=0-7)
12: Phase Clock Direction (0=output, 1=input)
(This must be 1)
13: Servo Clock Direction (0=output, 1=input)
(This must be 1)
14-15: Not used (report as zero)
16-19: Phase Clock Frequency Control n (I997)
(f=MAXPHASE / [n+1], n=0-15)

20-23: Servo Clock Frequency Control n
(f=PHASE / [n+1], n=0-15)

Chan #	9	10
Hex	**[$C090]**	**[$C098]**

`Y:$C09x` Channel n Time between last two encoder counts (SCLK cycles)
`X:$C09x` Channel n Status Word

Bits:
- 0-2 Captured Hall Effect Device (UVW) State
- 3 Invalid demultiplex of C, U, V, and W
- 4-7 Not used (reports as 0)
- 8 Encoder Count Error (0 on counter reset, 1 on illegal transition)
- 9 Position Compare (EQUn) output value
- 10 Position-Captured-On-Gated-Index Flag (=0 on read of captured position register, =1 on trigger capture)
- 11 Position-Captured Flag (on any trigger) (=0 on read of captured position register, =1 on trigger capture)
- 12 Handwheel 1 Channel A (HWAn) Input Value
- 13 Handwheel 1 Channel B (HWBn) Input Value
- 14 Handwheel 1 Channel C (Index, HWCn) Input Value (ungated)
- 15 Amplifier Fault (FAULTn) Input Value
- 16 Home Flag (HMFLn) Input Value
- 17 Positive End Limit (PLIMn) Input Value
- 18 Negative End Limit (MLIMn) Input Value
- 19 User Flag (USERn) Input Value
- 20 FlagWn Input Value
- 21 FlagVn Input Value
- 22 FlagUn Input Value
- 23 FlagTn Input Value

Chan #	9	10
Hex	**[$C091]**	**[$C099]**

`Y:$C09x` Channel n Encoder Time Since Last Encoder Count (SCLK cycles)
`X:$C09x` Channel n Encoder Phase Position Capture Register (counts)

Chan #	9	10
Hex	**[$C092]**	**[$C09A]**

`Y:$C09x` Channel n Output A Command Value

Bits:
- 8-23: PWM Command Value
- 6-23: Serial DAC Command Value
- 0-5: Not Used

`X:$C09x` Channel n Encoder Servo Position Capture Register

Bits:
- 0: Direction of last count (0=up, 1=down)
- 1-23: Position counter (units of counts)

Chan #	9	10
Hex	**[$C093]**	**[$C09B]**

Y:$C09x Channel n Output B Command Value
Bits: 8-23: PWM Command Value
6-23: Serial DAC Command Value
0-5: Not used
X:$C09x Channel n Flag Position Capture Value; 24 bits, units of counts

Chan #	9	10
Hex	**[$C094]**	**[$C09C]**

Y:$C09x Channel n Output C Command Value
Bits: 8-23: PWM Command Value
0-23: PFM Command Value
X:$C094 Channel 9-10 ADC Strobe Word, 24 bits
(Shifted out MSB first one bit per DAC_CLK cycle, starting on rising edge of phase clock)
X:$C09C Channel 9-10 PWM, PFM, MaxPhase Control Word
Bits: 0-7: PWM Dead Time (16*PWM CLK cycles)
also PFM pulse width (PFM CLK cycles)
8-23: PWM Max Count Value
PWM Frequency = 117.96MHz / [10(MaxCount+1)]
"MaxPhase" Frequency = 2*PWM* Frequency

Chan #	9	10
Hex	**[$C095]**	**[$C09D]**

Y:$C09x Supplementary Channel n* ADC A Input Value
Bits: 6-23: Serial ADC Value
0-5: Not used
X:$C09x Channel n Control Word
Bits 0-1: Encoder Decode Control
00: Pulse and direction decode
01: x1 quadrature decode
10: x2 quadrature decode
11: x4 quadrature decode
2-3: Direction & Timer Control
00: Standard timer control, external signal source, no inversion
01: Standard timer control, external signal source, invert direction
10: Standard timer control, internal PFM source, no inversion
11: Alternate timer control, external signal source
4-5: Position Capture Control
00: Software capture (by setting bit 6)
01: Use encoder index alone
10: Use capture flag alone
11: Use encoder index and capture flag
6: Index Capture Invert Control (0=no inversion, 1=inversion)
7: Flag Capture Invert Control (0=no inversion, 1=inversion)
8-9: Capture Flag Select Control
00: Home Flag (HMFLn)

01: Positive Limit (PLIMn)
10: Negative Limit (MLIMn)
11: User Flag (USERn)

10: Encoder Counter Reset Control (1=reset)
11: Position Compare Initial State Write Enable
12: Position Compare Initial State Value
13: Position Compare Channel Select
(0= use this channel's encoder; 1=use first encoder on IC)
14: AENAn output value
15: Gated Index Select for Position Capture
(0=ungated index, 1=gated index)
16: Invert AB for Gated Index
(0: Gated Signal=A&B&C; 1: Gated Signal=A/&B/&C)
17: Index channel demultiplex control (0=no demux, 1=demux)
18: Reserved for future use (reports as 0)
19: Invert PFM Direction Control (0=no inversion, 1=invert)
20: Invert A & B Output Control (0=no inversion, 1=invert)
21: Invert C Output Control (0=no inversion, 1=invert)
22: Output A & B Mode Select (0=PWM, 1=DAC)
23: Output C Mode Select (0=PWM, 1=PFM)

Chan #	9	10
Hex	**[$C096]**	**[$C09E]**

`Y:$C09x` Supplementary Channel n* ADC B Input Value (uses SEL3 in dedicated mode)
Bits: 6-23: Serial ADC Value
0-5: Not used

`X:$C09x` Channel n Encoder Compare Auto-increment value (24 bits, units of counts)

Chan #	9	10
Hex	**[$C097]**	**[$C09F]**

`Y:$C09x` Channel n Encoder Compare A Value (24 bits, units of counts)
`X:$C09x` Channel n Encoder Compare B Value (24 bits, units of counts)
`Y:$C0A0` MACRO Node 0 24-bit command(write) and feedback (read) register
`X:$C0A0` MACRO Node 2 24-bit command(write) and feedback (read) register
`Y:$C0A1` MACRO Node 0 1st 16-bit command(write) and feedback (read) register
(bits 8-23; bits 0-7 not used)
`X:$C0A1` MACRO Node 2 1st 16-bit command(write) and feedback (read) register
(bits 8-23; bits 0-7 not used)
`Y:$C0A2` MACRO Node 0 1st 16-bit command(write) and feedback (read) register
(bits 8-23; bits 0-7 not used)
`X:$C0A2` MACRO Node 2 2nd 16-bit command(write) and feedback (read) register
(bits 8-23; bits 0-7 not used)
`Y:$C0A3` MACRO Node 0 3rd 16-bit command(write) and feedback (read) register
(bits 8-23; bits 0-7 not used)
`X:$C0A3` MACRO Node 2 3rd 16-bit command(write) and feedback (read) register
(bits 8-23; bits 0-7 not used)
`Y:$C0A4` MACRO Node 1 24-bit command(write) and feedback (read) register

X:$C0A4	MACRO Node 3 24-bit command(write) and feedback (read) register
Y:$C0A5	MACRO Node 1 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0A5	MACRO Node 3 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0A6	MACRO Node 1 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0A6	MACRO Node 3 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0A7	MACRO Node 1 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0A7	MACRO Node 3 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0A8	MACRO Node 4 24-bit command(write) and feedback (read) register
X:$C0A8	MACRO Node 4 24-bit command(write) and feedback (read) register
Y:$C0A9	MACRO Node 6 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0A9	MACRO Node 6 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0AA	MACRO Node 4 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0AA	MACRO Node 6 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0AB	MACRO Node 4 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0AB	MACRO Node 6 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0AC	MACRO Node 5 24-bit command(write) and feedback (read) register
X:$C0AC	MACRO Node 7 24-bit command(write) and feedback (read) register
Y:$C0AD	MACRO Node 5 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0AD	MACRO Node 7 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0AE	MACRO Node 5 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0AE	MACRO Node 7 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0AF	MACRO Node 5 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0AF	MACRO Node 7 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B0	MACRO Node 8 24-bit command(write) and feedback (read) register
X:$C0B0	MACRO Node 10 24-bit command(write) and feedback (read) register
Y:$C0B1	MACRO Node 8 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0B1	MACRO Node 10 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B2	MACRO Node 1 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)

X:$C0B2	MACRO Node 10 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B3	MACRO Node 8 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0B3	MACRO Node 10 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B4	MACRO Node 9 24-bit command(write) and feedback (read) register
X:$C0B4	MACRO Node 11 24-bit command(write) and feedback (read) register
Y:$C0B5	MACRO Node 9 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0B5	MACRO Node 11 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B6	MACRO Node 9 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0B6	MACRO Node 11 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B7	MACRO Node 9 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0B7	MACRO Node 11 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0B8	MACRO Node 12 24-bit command(write) and feedback (read) register
X:$C0B8	MACRO Node 14 24-bit command(write) and feedback (read) register
Y:$C0B9	MACRO Node 12 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0B9	MACRO Node 14 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0BA	MACRO Node 12 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0BA	MACRO Node 14 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0BB	MACRO Node 12 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0BB	MACRO Node 14 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0BC	MACRO Node 13 24-bit command(write) and feedback (read) register
X:$C0BC	MACRO Node 15 24-bit command(write) and feedback (read) register
Y:$C0BD	MACRO Node 13 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0BD	MACRO Node 15 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0BE	MACRO Node 13 1st 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0BE	MACRO Node 15 2nd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
Y:$C0BF	MACRO Node 13 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)
X:$C0BF	MACRO Node 15 3rd 16-bit command(write) and feedback (read) register (bits 8-23; bits 0-7 not used)

Y:$FFC0	ACC-1E/6E Analog-to-Digital Converters (low 12 bits) Write operation: channel select (Channels 0-7) and mode; Read operation: converted value of selected channel ACC-1E/6E Analog-to-Digital Converters (high 12 bits) Write operation: channel select (Channels 8-15) and mode; Read operation: converted value of selected channel	
Y:$FFC0	ACC-4E Board with Jumper E1 selected	
Y:$FFC8	ACC-4E Board with Jumper E2 selected	
Y:$FFD0	ACC-4E Board with Jumper E3 selected	
Y:$FFD8	ACC-4E Board with Jumper E4 selected	
Bits:	0	IN00 (J4 pin 1)
	1	IN01 (J4 pin 3)
	2	IN02 (J4 pin 5)
	3	IN03 (J4 pin 7)
	4	IN04 (J4 pin 9)
	5	IN05 (J4 pin 11)
	6	IN06 (J4 pin 13)
	7	IN07 (J4 pin 15)
Y:$FFC0	ACC-3E Board with Jumper E1 selected	
Y:$FFC8	ACC-3E Board with Jumper E2 selected	
Y:$FFD0	ACC-3E Board with Jumper E3 selected	
Y:$FFD8	ACC-3E Board with Jumper E4 selected	
Bits:	0	I/O00 (J4 pin 47; requires Option A)
	1	I/O01 (J4 pin 45; requires Option A)
	2	I/O02 (J4 pin 43; requires Option A)
	3	I/O03 (J4 pin 41; requires Option A)
	4	I/O04 (J4 pin 39; requires Option A)
	5	I/O05 (J4 pin 37; requires Option A)
	6	I/O06 (J4 pin 35; requires Option A)
	7	I/O07 (J4 pin 33; requires Option A)
	8	I/O48 (J6 pin 47; requires Option B)
	9	I/O49 (J6 pin 45; requires Option B)
	10	I/O50 (J6 pin 43; requires Option B)
	11	I/O51 (J6 pin 41; requires Option B)
	12	I/O52 (J6 pin 39; requires Option B)
	13	I/O53 (J6 pin 37; requires Option B)
	14	I/O54 (J6 pin 35; requires Option B)
	15	I/O55 (J6 pin 33; requires Option B)
	16	I/O96 (J8 pin 47; requires Option C)
	17	I/O97 (J8 pin 45; requires Option C)
	18	I/O98 (J8 pin 43; requires Option C)
	19	I/O99 (J8 pin 41; requires Option C)
	20	I/O100 (J8 pin 39; requires Option C)
	21	I/O101 (J8 pin 37; requires Option C)
	22	I/O102 (J8 pin 35; requires Option C)
	23	I/O103 (J8 pin 33; requires Option C)
Y:$FFC1	ACC-4E Board with Jumper E1 selected	
Y:$FFC9	ACC-4E Board with Jumper E2 selected	
Y:$FFD1	ACC-4E Board with Jumper E3 selected	
Y:$FFD9	ACC-4E Board with Jumper E4 selected	

Bits:	0	IN08 (J4 pin 21)
	1	IN09 (J4 pin 23)
	2	IN10 (J4 pin 25)
	3	IN11 (J4 pin 27)
	4	IN12 (J4 pin 29)
	5	IN13 (J4 pin 31)
	6	IN14 (J4 pin 33)
	7	IN15 (J4 pin 35)

`Y:$FFC1` ACC-3E Board with Jumper E1 selected
`Y:$FFC9` ACC-3E Board with Jumper E2 selected
`Y:$FFD1` ACC-3E Board with Jumper E3 selected
`Y:$FFD9` ACC-3E Board with Jumper E4 selected

Bits:	0	I/O08 (J4 pin 31; requires Option A)
	1	I/O09 (J4 pin 29; requires Option A)
	2	I/O10 (J4 pin 27; requires Option A)
	3	I/O11 (J4 pin 25; requires Option A)
	4	I/O12 (J4 pin 23; requires Option A)
	5	I/O13 (J4 pin 21; requires Option A)
	6	I/O14 (J4 pin 19; requires Option A)
	7	I/O15 (J4 pin 17; requires Option A)
	8	I/O56 (J6 pin 31; requires Option B)
	9	I/O57 (J6 pin 29; requires Option B)
	10	I/O58 (J6 pin 27; requires Option B)
	11	I/O59 (J6 pin 25; requires Option B)
	12	I/O60 (J6 pin 23; requires Option B)
	13	I/O61 (J6 pin 21; requires Option B)
	14	I/O62 (J6 pin 19; requires Option B)
	15	I/O63 (J6 pin 17; requires Option B)
	16	I/O104 (J8 pin 31; requires Option C)
	17	I/O105 (J8 pin 29; requires Option C)
	18	I/O106 (J8 pin 27; requires Option C)
	19	I/O107 (J8 pin 25; requires Option C)
	20	I/O108 (J8 pin 23; requires Option C)
	21	I/O109 (J8 pin 21; requires Option C)
	22	I/O110 (J8 pin 19; requires Option C)
	23	I/O111 (J8 pin 17; requires Option C)

`Y:$FFC2` ACC-4E Board with Jumper E1 selected
`Y:$FFCA` ACC-4E Board with Jumper E2 selected
`Y:$FFD2` ACC-4E Board with Jumper E3 selected
`Y:$FFDA` ACC-4E Board with Jumper E4 selected

Bits:	0	IN16 (J4 pin 41)
	1	IN17 (J4 pin 43)
	2	IN18 (J4 pin 45)
	3	IN19 (J4 pin 47)
	4	IN20 (J4 pin 49)
	5	IN21 (J4 pin 51)
	6	IN22 (J4 pin 53)
	7	IN23 (J4 pin 55)

`Y:$FFC2` ACC-3E Board with Jumper E1 selected
`Y:$FFCA` ACC-3E Board with Jumper E2 selected

Y:$FFD2	ACC-3E Board with Jumper E3 selected	
Y:$FFDA	ACC-3E Board with Jumper E4 selected	
Bits:	0	I/O16 (J4 pin 15; requires Option A)
	1	I/O17 (J4 pin 13; requires Option A)
	2	I/O18 (J4 pin 11; requires Option A)
	3	I/O19 (J4 pin 09; requires Option A)
	4	I/O20 (J4 pin 07; requires Option A)
	5	I/O21 (J4 pin 05; requires Option A)
	6	I/O22 (J4 pin 03; requires Option A)
	7	I/O23 (J4 pin 01; requires Option A)
	8	I/O64 (J6 pin 15; requires Option B)
	9	I/O65 (J6 pin 13; requires Option B)
	10	I/O66 (J6 pin 11; requires Option B)
	11	I/O67 (J6 pin 09; requires Option B)
	12	I/O68 (J6 pin 07; requires Option B)
	13	I/O69 (J6 pin 05; requires Option B)
	14	I/O70 (J6 pin 03; requires Option B)
	15	I/O71 (J6 pin 01; requires Option B)
	16	I/O112 (J8 pin 15; requires Option C)
	17	I/O113 (J8 pin 13; requires Option C)
	18	I/O114 (J8 pin 11; requires Option C)
	19	I/O115 (J8 pin 09; requires Option C)
	20	I/O116 (J8 pin 07; requires Option C)
	21	I/O117 (J8 pin 05; requires Option C)
	22	I/O118 (J8 pin 03; requires Option C)
	23	I/O119 (J8 pin 01; requires Option C)
Y:$FFC3	ACC-4E Board with Jumper E1 selected	
Y:$FFCB	ACC-4E Board with Jumper E2 selected	
Y:$FFD3	ACC-4E Board with Jumper E3 selected	
Y:$FFDB	ACC-4E Board with Jumper E4 selected	
Bits:	0	OUT00 (J4 pin 2)
	1	OUT01 (J4 pin 4)
	2	OUT02 (J4 pin 6)
	3	OUT03 (J4 pin 8)
	4	OUT04 (J4 pin 10)
	5	OUT05 (J4 pin 12)
	6	OUT06 (J4 pin 14)
	7	OUT07 (J4 pin 16)
Y:$FFC3	ACC-3E Board with Jumper E1 selected	
Y:$FFCB	ACC-3E Board with Jumper E2 selected	
Y:$FFD3	ACC-3E Board with Jumper E3 selected	
Y:$FFDB	ACC-3E Board with Jumper E4 selected	
Bits:	0	I/O24 (J5 pin 47; requires Option A)
	1	I/O25 (J5 pin 45; requires Option A)
	2	I/O26 (J5 pin 43; requires Option A)
	3	I/O27 (J5 pin 41; requires Option A)
	4	I/O28 (J5 pin 39; requires Option A)
	5	I/O29 (J5 pin 37; requires Option A)
	6	I/O30 (J5 pin 35; requires Option A)
	7	I/O31 (J5 pin 33; requires Option A)

	8	I/O72 (J7 pin 47; requires Option B)
	9	I/O73 (J7 pin 45; requires Option B)
	10	I/O74 (J7 pin 43; requires Option B)
	11	I/O75 (J7 pin 41; requires Option B)
	12	I/O76 (J7 pin 39; requires Option B)
	13	I/O77 (J7 pin 37; requires Option B)
	14	I/O78 (J7 pin 35; requires Option B)
	15	I/O79 (J7 pin 33; requires Option B)
	16	I/O120 (J9 pin 47; requires Option C); with Option C1/C2, also OUT120 (J10 pin 1)
	17	I/O121 (J9 pin 45; requires Option C); with Option C1/C2, also OUT121 (J10 pin 2)
	18	I/O122 (J9 pin 43; requires Option C); with Option C1/C2, also OUT122 (J10 pin 3)
	19	I/O123 (J9 pin 41; requires Option C); with Option C1/C2, also OUT123 (J10 pin 4)
	20	I/O124 (J9 pin 39; requires Option C); with Option C1/C2, also OUT124 (J10 pin 5)
	21	I/O125 (J9 pin 37; requires Option C); with Option C1/C2, also OUT125 (J10 pin 6)
	22	I/O126 (J9 pin 35; requires Option C); with Option C1/C2, also OUT126 (J10 pin 7)
	23	I/O127 (J9 pin 33; requires Option C); with Option C1/C2, also OUT127 (J10 pin 8)
`Y:$FFC4`	ACC-4E Board with Jumper E1 selected	
`Y:$FFCC`	ACC-4E Board with Jumper E2 selected	
`Y:$FFD4`	ACC-4E Board with Jumper E3 selected	
`Y:$FFDC`	ACC-4E Board with Jumper E4 selected	
Bits:	0	OUT08 (J4 pin 22)
	1	OUT09 (J4 pin 24)
	2	OUT10 (J4 pin 26)
	3	OUT11 (J4 pin 28)
	4	OUT12 (J4 pin 30)
	5	OUT13 (J4 pin 32)
	6	OUT14 (J4 pin 34)
	7	OUT15 (J4 pin 36)
`Y:$FFC4`	ACC-3E Board with Jumper E1 selected	
`Y:$FFCC`	ACC-3E Board with Jumper E2 selected	
`Y:$FFD4`	ACC-3E Board with Jumper E3 selected	
`Y:$FFDC`	ACC-3E Board with Jumper E4 selected	
Bits:	0	I/O32 (J5 pin 31; requires Option A)
	1	I/O33 (J5 pin 29; requires Option A)
	2	I/O34 (J5 pin 27; requires Option A)
	3	I/O35 (J5 pin 25; requires Option A)
	4	I/O36 (J5 pin 23; requires Option A)
	5	I/O37 (J5 pin 21; requires Option A)
	6	I/O38 (J5 pin 19; requires Option A)
	7	I/O39 (J5 pin 17; requires Option A)
	8	I/O40 (J7 pin 31; requires Option B)
	9	I/O41 (J7 pin 29; requires Option B)

	10	I/O42 (J7 pin 27; requires Option B)
	11	I/O43 (J7 pin 25; requires Option B)
	12	I/O44 (J7 pin 23; requires Option B)
	13	I/O45 (J7 pin 21; requires Option B)
	14	I/O46 (J7 pin 19; requires Option B)
	15	I/O47 (J7 pin 17; requires Option B)
	16	I/O128 (J9 pin 31; requires Option C); with Option C1/C2, also IN128 (J10 pin 9)
	17	I/O129 (J9 pin 29; requires Option C); with Option C1/C2, also IN129 (J10 pin 10)
	18	I/O130 (J9 pin 27; requires Option C); with Option C1/C2, also IN130 (J10 pin 11)
	19	I/O131 (J9 pin 25; requires Option C); with Option C1/C2, also IN131 (J10 pin 12)
	20	I/O132 (J9 pin 23; requires Option C); with Option C1/C2, also IN132 (J10 pin 13)
	21	I/O133 (J9 pin 21; requires Option C); with Option C1/C2, also IN133 (J10 pin 14)
	22	I/O134 (J9 pin 19; requires Option C); with Option C1/C2, also IN134 (J10 pin 15)
	23	I/O135 (J9 pin 17; requires Option C); with Option C1/C2, also IN135 (J10 pin 16)
`Y:$FFC5`	ACC-4E Board with Jumper E1 selected	
`Y:$FFCD`	ACC-4E Board with Jumper E2 selected	
`Y:$FFD5`	ACC-4E Board with Jumper E3 selected	
`Y:$FFDD`	ACC-4E Board with Jumper E4 selected	
Bits:	0	OUT16 (J4 pin 42)
	1	OUT17 (J4 pin 44)
	2	OUT18 (J4 pin 46)
	3	OUT19 (J4 pin 48)
	4	OUT20 (J4 pin 50)
	5	OUT21 (J4 pin 52)
	6	OUT22 (J4 pin 54)
	7	OUT23 (J4 pin 56)
`Y:$FFC5`	ACC-3E Board with Jumper E1 selected	
`Y:$FFCD`	ACC-3E Board with Jumper E2 selected	
`Y:$FFD5`	ACC-3E Board with Jumper E3 selected	
`Y:$FFDD`	ACC-3E Board with Jumper E4 selected	
Bits:	0	I/O40 (J5 pin 15; requires Option A)
	1	I/O41 (J5 pin 13; requires Option A)
	2	I/O42 (J5 pin 11; requires Option A)
	3	I/O43 (J5 pin 09; requires Option A)
	4	I/O44 (J5 pin 07; requires Option A)
	5	I/O45 (J5 pin 05; requires Option A)
	6	I/O46 (J5 pin 03; requires Option A)
	7	I/O47 (J5 pin 01; requires Option A)
	8	I/O88 (J7 pin 15; requires Option B)
	9	I/O89 (J7 pin 13; requires Option B)
	10	I/O90 (J7 pin 11; requires Option B)
	11	I/O91 (J7 pin 09; requires Option B)

12 I/O92 (J7 pin 07; requires Option B)
13 I/O93 (J7 pin 05; requires Option B)
14 I/O94 (J7 pin 03; requires Option B)
15 I/O95 (J7 pin 01; requires Option B)

16 I/O136 (J9 pin 15; requires Option C);
with Option C1/C2, also IN136 (J10 pin 17)
17 I/O137 (J9 pin 13; requires Option C);
with Option C1/C2, also IN137 (J10 pin 18)
18 I/O138 (J9 pin 11; requires Option C);
with Option C1/C2, also IN138 (J10 pin 19)
19 I/O139 (J9 pin 09; requires Option C);
with Option C1/C2, also IN139 (J10 pin 20)
20 I/O140 (J9 pin 07; requires Option C);
with Option C1/C2, also IN140 (J10 pin 21)
21 I/O141 (J9 pin 05; requires Option C);
with Option C1/C2, also IN141 (J10 pin 22)
22 I/O142 (J9 pin 03; requires Option C);
with Option C1/C2, also IN142 (J10 pin 23)
23 I/O143 (J9 pin 01; requires Option C);
with Option C1/C2, also IN143 (J10 pin 24)

`Y:$FFE0` ACC-9E, 10E, 11E, 12E Board with Jumper E1 selected
ACC-14E board with SW1-1 ON, SW1-2 ON

`Y:$FFE8` ACC-9E, 10E, 11E, 12E Board with Jumper E2 selected
ACC-14E board with SW1-1 OFF, SW1-2 ON

`Y:$FFF0` ACC-9E, 10E, 11E, 12E Board with Jumper E3 selected
ACC-14E board with SW1-1 ON, SW1-2 OFF

`Y:$FFF8` ACC-9E, 10E, 11E, 12E Board with Jumper E4 selected
ACC-14E board with SW1-1 OFF, SW1-2 OFF