HP 10751A

Air Sensor

Optics Type

Use the *OPT mnemonic to specify which optics are in use (linear, plane mirror or high resolution). A value of 0 corresponds to linear optics, 1 corresponds to plane mirror optics and a value of 2 corresponds to high resolution optics.

NOTE

*For proper range checking *OPT should be set before programming *AVS, *DPD, *PRD, or *RES.*

Direction Sense

Use the *DIR mnemonic to specify the direction sense of the optics installation. Set this mnemonic so the *POS value increases as the optics move away from each other. Do not use this mnemonic to swap the A/UP lines with the B/DOWN lines (Use the sign of *RES to swap A/Up and B/Down).

Measurement Receiver

Use either the *MEX or *MIN mnemonic to specify the receiver being used. Sending the *MEX mnemonic selects an external HP 10780 Receiver. Sending *MIN selects the internal receiver on an HP 5518A Laser Head.

Pulse Units Resolution and Direction

Use the *PUN and *RES mnemonics to specify the output units and resolution. *PUN 0 selects millimeters while *PUN 1 selects inches. The value sent to *RES specifies the output pulse resolution in the selected units. For example, sending *PUN 0 and *RES 1E-3 specifies the output pulse resolution as 1 micron. It also specifies that all distance information (*DPD, *MPO, *POF, *POS, *PRD, *PSE, and *RES) will be in millimeters. Set *RES to a negative value to swap the A/Up output lines with the B/Down lines.

NOTE

*For proper range checking *PUN should be set before programming *AVS, *DPD, *MPO, *PRD, or *RES.*

Output Pulse Rate and Format

The *KHZ and *QAD mnemonics specify the maximum output pulse rate and format respectively. Sending *QAD 1 selects quadrature outputs while *QAD 0 selects up/down pulse outputs. The value sent with *KHZ specifies the maximum transition rate (for A quad B format) or pulse rate (for up/down format).

Compensation Options

Use the *BCN, *CMD, *CMP, *CUR, *RCM, *WCM and *COF mnemonics to setup the desired compensation operation. See the "Using Compensation" section for details on how these mnemonics interact with the environmental compensation system. See the note on page 4E-11 to compensate for known offsets to a laser head's vacuum wavelength.

Preset and Deadpath Distances

Use the *PRD and *DPD mnemonics to tell the board where the optics will be when the laser system is reset (*Figure 4E-2*). Set the deadpath distance (*DPD) to the distance between the interferometer and the reflector when the laser system is normally reset (for additional details on deadpath, see Product Note 5527A/B-2 "Achieving Maximum Accuracy and Repeatability").

Set the preset distance (*PRD) to the distance between the normal reset location and either an alternate reset location or the end of travel location. The preset distance is used in conjunction with the PRESET_ENABLE– input to simplify error recovery (see the "Error Recovery" section for details). For both mnemonics, the units will be either mm or inches as selected by *PUN.

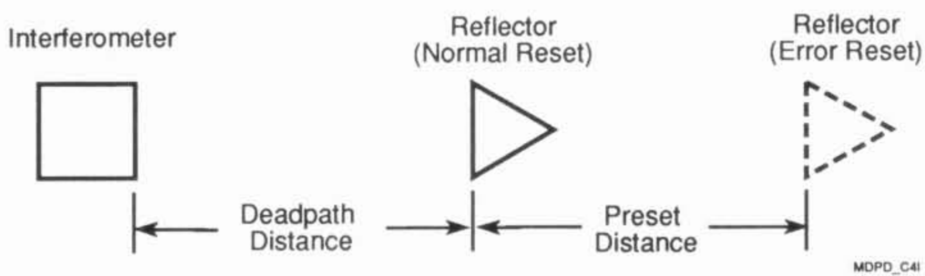

Figure 4E-2. Deadpath and Preset Distances

Maximum Position Offset Distance

Use the *MPO mnemonic to specify the maximum geometric correction value. Set *MPO to the maximum distance that will be sent to the position offset mnemonic (*POF) (see Geometric Correction section for additional information).

Error Mode

Use the *EMD mnemonic to select how the board will respond to errors. The default setting, 0, causes the board to disable the pulse outputs when this board has a measurement error. A value of 1 disables the pulses on any error on this axis. A value of 2 disables the pulses on any system measurement error. A value of 3 disables the pulses on any system error.

Sampling Options

Use the *MUR and *PUR mnemonics to set the rate for sampling position registers. Set the corresponding rate to zero if the *MUL and *POS values are not read during normal operation. See the *MUR and *PUR mnemonics in the reference section for details.

Use the *MEN and *PEN mnemonics to enable or disable external sampling of the position registers. Sending 0 disables external sampling, 1 enables it. See the *MLE and *PSE mnemonics in the reference section for additional details.

Serial Port Options

Use the *SAM, *SAR, *SBM, and *SBR mnemonics to set the serial port, modes and baud rates to those desired. See the descriptions of these mnemonics in the reference section for details.

SOFTWARE INTERFACE

After setup, the HP 10934A operates with only the hardware control and status lines connected. Additional capabilities are available by using one of the four data interfaces during operation. Position and status can be read, commands can be issued, and offsets can be adjusted. The serial and parallel ports have access to the entire HP 5507A/B, and respond to mnemonics for other boards as if they had been issued over HP-IB.

Commands

The HP 10934A responds to seven command mnemonics. They are:

- (*SGO) - start (initialize) all HP 10934A boards in the system
- (*AGO) - start (initialize) this axis only
- (*STP) - stop this axis only
- (*AER) - clear errors on this axis only
- (*CUR 0) - update the base compensation number
- (*PUR 0) - sample the position counter
- (*MUR 0) - sample the multiplier output

See these mnemonics in the reference section for details.

Position Information

The HP 10934A has several mnemonics that return position information. *Figure 4E-3*, HP 10934A Position Information Block Diagram illustrates what each mnemonic returns, the units associated with it, and when the value is sampled. The values in the two base registers are used to calculate the final output value in the selected units (mm or inch). See the *MLE, *MUL, *POS, *PSE, *RLP and *RSV mnemonics for additional information.

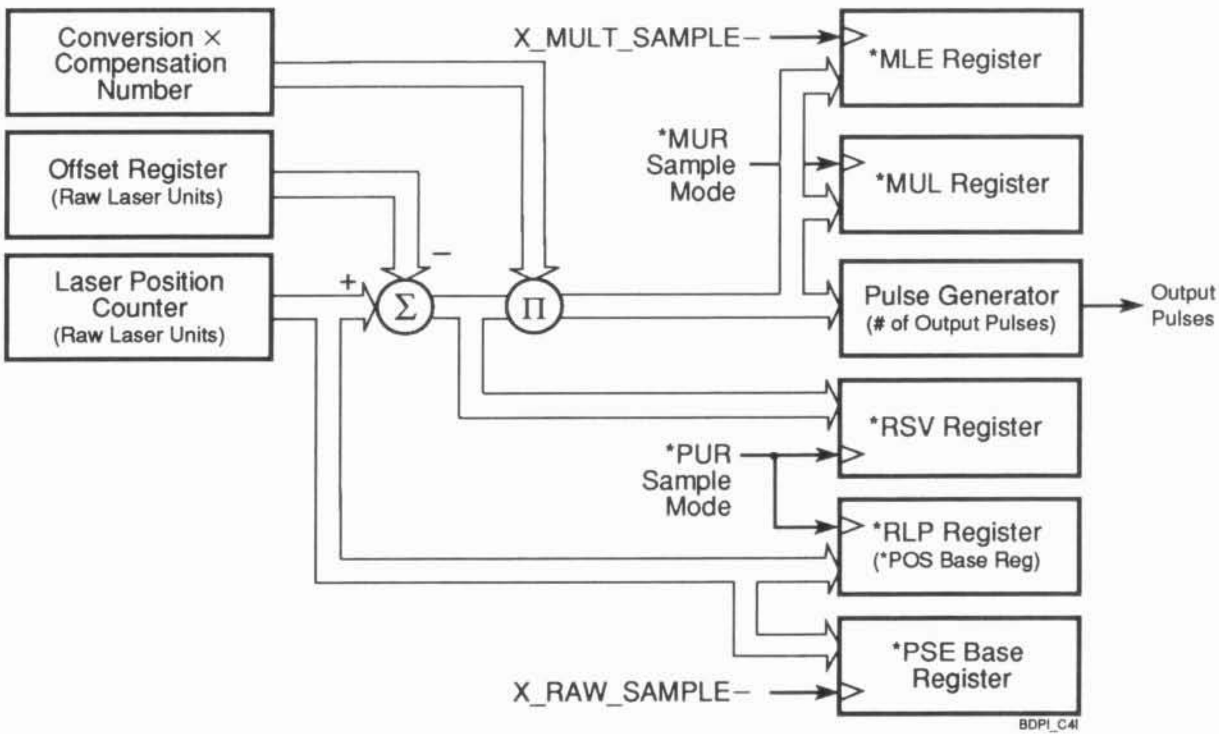

Figure 4E-3. HP 10934A Position Information Block Diagram

Offset Adjustments

The HP 10934A supports geometric correction through the position offset mnemonic. Refer to the Geometric Correction section and the *POF mnemonic for operational details.

Status Information

The *STA and *MSW mnemonics return the board's status. The definitions for each bit in *MSW are detailed in the reference section. The *STA Values and corresponding errors are described in the "Error Message" section.

HP 5507A/B Information and Control

All the interfaces support some system level commands and data requests. Operation is similar across all interfaces. Use the mnemonics reference section for this board and the HP-IB board for details on the CNFG, DFMT, ECHO, ERRM, HSMD, INST, and ISTA mnemonics.

Additionally, the serial and parallel ports support command numbers as well as mnemonics for faster operation. The ASGN and CNUM mnemonics are used to associate command numbers with specific mnemonics. Refer to the "Using Command Numbers" section and the reference section on these mnemonics for details.

ERROR RECOVERY

Due to the nature of laser interferometry, a momentary blockage of the laser beam causes the laser system to lose track of position and generate an error. For the A-Quad-B board, this condition turns off the pulse output section of the board, requiring that the axis involved be initialized again (*AGO).

To prevent an axis from underflowing the laser counter, the preset distance (when enabled) plus the deadpath distance must be greater than the distance the stage can physically move in the minus direction. Thus, to recover from a beam blockage error and return to the normal reset location, the preset distance (*PRD) and the ENABLE_PRESET– input line must be used.

The PRESET_ENABLE– line selects whether the preset distance is included (input true - low) or not (input false - high) during axis initialization. When included, the axis hardware is preset as if the board had been reset normally and the stage then moved to the preset position, however no pulses are output reflecting the simulated motion to the preset position.

The preset distance (*PRD) may be set by the machine operator to the approximate stage position, or set at the factory to the maximum stage position. After initialization with preset enabled, the stage will have full travel range without underflowing the laser counter. This operation allows the system to return to the normal reset location using laser feedback without generating additional errors.

To automatically control the PRESET_ENABLE– line, connect it to a switch that opens when the stage is close to the normal reset position.

USING COMPENSATION

The HP 10934A A-Quad-B Axis Board accommodates several methods of updating the environmental compensation number. (See Sections II and V for information on why environmental compensation may be required.)

- Manual
- Automatic Software Trigger
- Automatic External Trigger
- Automatic Internal Trigger (timed)
- Automatic Limit Trigger

Additionally, each axis can have its own compensation offset value for calibrating out known scaling factors. (See "Compensation Offset" below.)

The *CUR, *BCN, *COF, *CMP, *WCM, *RCM, and *CMD mnemonics are used to setup and control all these options. The latter four mnemonics access an internal table (or array) of mnemonics and data values (see *Table 4E-2* below) used in the compensation process. The *CMP value is the index into this table. The *RCM and *CMD values indicate what mnemonics and data values are in the table at the index specified by *CMP. The *WCM and *CMD mnemonics are used to change these values.

Table 4E-2. Internal Array of Compensation Mnemonics and Data

INDEX	MNEMONIC	DATA
***CMP Value**	***RCM String (Use *WCM Value; Value to Change)**	***CMD Value**
0	Mnemonic used to obtain a new Compensation Number. (*BCN is set to the value read.)	No Data Value Allowed
1	Mnemonic used to set the Compensation Limit Value.	The Compensation Limit Value
2	Mnemonic used to read the Initial Compensation Number. (Value read will be sent to the mnemonic at *CMP = 0.)	No Data Value Allowed
3	Mnemonic used to set the Filter Time Constant.	The Filter Time Constant Value
4	Mnemonic used to set the Coefficient of Thermal Expansion value.	The Coefficient of Thermal Expansion Value
5	1st mnemonic the *BCN value is sent to after updating. (If undefined, then rows 6 through 9 will be ignored.)	No Data Value Allowed
6	2nd mnemonic the *BCN value is sent to after updating. (If undefined, then rows 7 through 9 will be ignored.)	No Data Value Allowed
7	3rd mnemonic the *BCN value is sent to after updating. (If undefined, then rows 8 and 9 will be ignored.)	No Data Value Allowed
8	4th mnemonic the *BCN value is sent to after updating. (If undefined, then row 9 will be ignored.)	No Data Value Allowed
9	5th mnemonic the *BCN value is sent to after updating.	No Data Value Allowed

Note that only the mnemonic at *CMP = 2 can address this board. All others must access other boards in the system.

Manual Compensation

For manual compensation, none of the internal table needs to be filled. To change the compensation value, a new compensation number is written to the desired boards through the *BCN mnemonic.

For multi-axis systems, rows 5 through 9 of the internal table may be filled in to send the new *BCN value to other mnemonics in the system. For example, in a three axis system (X,Y, and Z), setup the X board by sending "XCMP5;XWCM22850;17230;XCMP6;XWCM23106;17230"CR/LF. This command puts "YBCN" in table row 5 and "ZBCN" in table row 6. Now, writing a value to XBCN causes the X board to send the same value to YBCN and ZBCN.

NOTE

*Do NOT have two or more boards update each other's *BCN value. Doing so causes an endless loop of writing *BCN values to each other's mailbox.*

Automatic Compensation

Use of the HP 10946C Automatic Compensation Board with an HP 10751A/B Air Sensor and an HP 10757A/B/C Material Temperature Sensor enables the automatic compensation features of the HP 10934A A-Quad-B Axis Board.

The HP 10946C has two sources for a compensation number. The first is the *CNV value calculated from the sensor readings. The second is the *WTC value determined from the HP 10717A Wavelength Tracker. One of these sources must be selected and the appropriate mnemonic entered into row zero of the table (use *CMP 0 and *WCM).

If the wavelength tracker is used, then rows two and three (*CMP 2 and *CMP 3) must also be filled in order to change the HP 10946C's default settings for initial compensation number and filter time constant.

To set the initial compensation number to some fixed value, save that value in this board's *BCN mnemonic, and also put this mnemonic in row 2 of the table.

If material temperature compensation is used, then row four of the table must also be filled.

To fill in a row in the table, send the *CMP mnemonic followed by the row number and then the *WCM mnemonic followed by two integers that contain the packed ASCII for the mnemonic being setup. If the row has data associated with it, then the *CMD mnemonic must also be sent followed by the desired data value.

For example, to set row 3 (filter time constant) to the VFTC 2.5 command, send the string "XCMP3;XWCM22086;21571;XCMD2.5"CR/LF. Note that the setup enable switch must be in the enable position to write mnemonics and data into this compensation system table. Sending the *SAV command stores the table entries in EEPROM.

Additional table rows may be filled in to send the compensation number to other boards in the system (see paragraph on Multi-axis systems under Manual Compensation).

Software Trigger

In the software trigger mode of operation, the base compensation number (*BCN) is updated by sending the *CUR0 command to the desired board. When the board receives this command, it executes a read of the mnemonic in row zero of the table, and sets *BCN equal to the returned value.

If row 5 has been setup, then the new *BCN value is sent to the mnemonic stored there and row 6 is checked. If row 5 has not been setup, then rows 6 through 9 are not checked. Likewise, rows 7,8 and 9 are checked and executed only if the preceding row has been setup. Adhere to the note under "Manual Compensation" when filling rows 5 through 9.

External Trigger

External trigger mode operates similar to the software trigger mode. To set the mode, send a *CUR0 command to turn off timed updates. Now the base compensation number will be updated when the COMP_UPDATE– line is brought low (true). The update process is identical to the process described in Software Trigger.

Timed Trigger

For the timed trigger mode, the periodic update rate is programmed using the *CUR mnemonic with a value less than –1 to specify number of seconds between updates (–5 indicates update every 5 seconds), or a value greater than 1 to specify a frequency (3 indicates to update at 3 Hz). Again, the update process is the same as in the Software Trigger mode except that an internal timer triggers the update instead of the software command.

Limit Trigger

The use of limit trigger mode requires that row 1 of the table be filled. When you send "XCMP1;XWCM22083;20044;XCMD1E-6"CR/LF, row 1 is set to the VCNL 0.000001 command, that will generate a compensation alert if the compensation number changes by more than 1 ppm (see *CNL in Section 4K). The A-Quad-B board will respond to a compensation alert when *CUR is set to –1. The response clears and resets the alert mode on the HP 10946C and also executes the update process described in Software Trigger.

Compensation Offset

The compensation offset (*COF) value is added to the base compensation number (*BCN) to form the total compensation number used in all calculations on the A-Quad-B axis board. Each board in the system can have a unique compensation offset value, while one base compensation number is used throughout the entire system.

This capability facilitates minor adjustments to the scaling of each axis in a system. For example, during calibration of a two axis machine, it is determined that each axis has a small cosine error. The X axis is off by 0.03 ppm and the Y axis is off by 0.06 ppm. Sending "XCOF0.03;YCOF0.06"CR/LF adjusts the compensation factor to correct for this error. Note that the *COF values are expressed in parts per million (ppm).

NOTE

*The wavelength constant used in the HP 10934A is 632.991370 nm. If the laser head being used differs from this vacuum wavelength value, then adjust the *BCN or *COF values to compensate (set *COF to –0.0253 for a vacuum wavelength of 632.991354 nm).*

GEOMETRIC CORRECTION

Many machines have repeatable errors that the built in scales cannot measure. Some CNC controllers have the ability to correct for these errors, but may not have a fine enough correction grid for some applications. The HP 10934A A-Quad-B Axis Board has the ability to add a correction term to the pulse output stream before it goes to the CNC controller, thus providing a correction grid of arbitrary resolution. The correction scheme is as follows:

- A computer is connected to the 8-bit parallel ports. (bus all boards together internally with a ribbon cable (see ordering information for cable part numbers).
- The machine is periodically calibrated throughout its working volume.
- The calibration data is loaded into the computer.
- As the machine moves, the computer continuously reads the positions of all axes, and computes a delta for each one based on the location of all axes.
- The calculated (or looked up) deltas are sent to each A-Quad-B board as position offsets (*POF) over the 8-bit parallel port.

The computer may be any device that can communicate over the parallel port (the serial port could also be used, however updates will be slower than with the parallel port). One could use a PC with a parallel interface card (NOT a parallel printer interface).

Alternately, one could build a dedicated computer on the HP 10941A Prototyping Board, and install it directly in the HP 5507A/B. When programming the computer, note that the *POS value includes the position offset correction, so the program must take this into account when computing the delta values.

COMMUNICATING THROUGH the SERIAL and PARALLEL PORTS

The serial and parallel ports use a superset of the data formats and syntax used by the HP-IB interface. Sending a command or data to a board in the HP 5507A/B consists of transmitting the four ASCII characters for the mnemonic and then optionally an ASCII representation of the data value. The transmission is ended with a carriage return/line feed sequence.

To read a value from a board in the HP 5507A/B, the four ASCII characters for the mnemonic are transmitted followed by a question mark and carriage return/line feed sequence. The board will then transmit the requested information in the data format selected for the interface being used (see the DFMT mnemonic and the data formats section below). For program clarity, the board ignores embedded spaces between mnemonics, data values, and item separators (semicolons and commas).

A sequence of commands, data writes and data requests can be strung together in one transmission by separating each mnemonic with either a semicolon or a comma. When there is more than one data request in the string, each request will generate an output from the board.

For example, the string "XOPT1;YOPT0;XDPD?;XPUN?"CR/LF sets the optics for both the X and Y boards and then requests the deadpath and units from the X board. The response will be the X deadpath value followed by a CR/LF sequence and then a 0 or 1 for the value of XPUN followed by another CR/LF sequence (assumes the ASCII output format).

Multiple reads and writes to the same mnemonic can be accomplished without re-sending the mnemonic. Simply separating the data values with a semicolon or comma after first sending the mnemonic writes multiple values. For example, sending "XWCM22083;20054"CR/LF writes both values to the XWCM mnemonic.

Sending the mnemonic followed by a sequence of question marks (separated by semicolons or commas) reads the same item multiple times. For example, sending "INST?;?;?;?"CR/LF reads four lines from the internal table of instructions.

Additionally, a copy of the last string sent is kept in memory. Sending just a question mark (or, for the parallel port only, requesting it to output additional data) after having sent a multiple query string, causes the multiple query string to be executed again. For example, first sending "XPOS?;YPOS?;ZPOS?"CR/LF causes the board to reply with three numbers separated by CR/LF. Now sending "?"CR/LF causes the board to again reply with three numbers separated by CR/LF. In both cases each number corresponds to one of the mnemonics in the initial string. Note that if a data write is embedded in the query string, then its value will also be included in the output string from the single question mark query. In the above example, if "XPUN0" was included in the string, then the first response would have three numbers, but the response to the "?"CR/LF would have four numbers.

To speed up communication, the serial and parallel ports also recognize command numbers. These numbers can be used instead of the four character ASCII string. When a command number is used, any data sent must be in internal format (see data formats) and must follow immediately after the command number (no intervening spaces). Additional commands, either as command numbers or mnemonics, may then follow in the same string. No item separator is required after a command number.

In all cases, unread data requests are lost when a new command string is received. For example, sending "XPOS?;YPOS?;ZPOS?"CR/LF and then only reading two numbers (the *XPOS* value and the *YPOS* value) leaves the *ZPOS* value in the output buffer until a new string arrives. If "XSTA?"CR/LF is then sent and one value read, the value returned will be the *XSTA* value, not the *ZPOS* value.

Should one of the data channels experience a communications error, it will return "Error"CR/LF in response to everything. Sending the control codes ESC (&H1B), ACK (&H06), and CAN (&H18), or BREAK for serial channels, clears and resets the input buffer.

Data Formats

The HP 10934A accepts and generates several data formats when communicating over the serial and parallel interfaces (see the HP-IB section for data formats for that interface). The following paragraphs detail the formats recognized for the different data types used in the HP 5507A/B.

ASCII Data Type

ASCII (or text) data items consist of a sequence of ASCII character bytes followed by a carriage return byte and then a line feed byte (the CR and LF are replaced by a comma or semicolon when sending multiple items to the HP 5507A/B).

Integer Data Type

Integer data items can have several representations depending on the integer type (byte or word). Both types can be represented as ASCII strings or a sequence of 1 to 2 bytes of binary data.

ASCII FORMAT

The ASCII representation consists of an optional space or minus sign followed by 1 to 5 digits. Byte types have a range of 0 to 255. Word types have a range of –32768 to +32767.

BYTE DATA (INTERNAL FORMAT)

The internal byte format is employed when command numbers are used. It consists of a single 8-bit byte of unsigned binary data.

WORD DATA (INTERNAL FORMAT)

The internal word format is employed when command numbers are used. It consists of two 8-bit bytes of signed binary data. The most significant byte is transmitted first.

Floating Point Data Type

Four representations are used for floating point data items. The first is ASCII text conforming to the IEEE-728 standard formats NR1, NR2, NR3. The next two are block formats corresponding to BDFA and BDFD of the IEEE-728 standard. The final format is the IEEE-754 double precision floating point format.

ASCII FORMAT

ASCII number representation consists of an optional sign digit, 1 to 10 number digits with an optional embedded decimal point, followed by an optional exponent. The optional exponent consists of an "e" or "E" followed by an optional sign digit and then 1 to 4 exponent digits.

BDFA FORMAT

The BDFA format consists of a four character prefix followed by eight bytes of binary data in the IEEE-754 double precision floating point format. The four character prefix consists of an ASCII number sign (#), an ASCII capital A (A), a binary 0 (NUL), and a binary 8 (BS).

BDFD FORMAT

The BDFD format consists of a two character prefix followed by eight bytes of binary data in the IEEE-754 double precision floating point format. The two character prefix consists of an ASCII number sign (#) and ASCII capital D (D).

IEEE-754 (INTERNAL FORMAT)

The IEEE-754 double precision format consists of a 64-bit data word sent as eight 8-bit bytes. The first byte sent is the most significant. The most significant bit is the sign, the next eleven bits are the exponent, and the remaining fifty-two bits are the fraction (first bit equals 0.5, next bit equals 0.25, ...). The value represented is found using the following rules:

if exponent = 0 then value = 0
if exponent = 2047 then value is undefined (generates error 527)
if $0 < \text{exponent} < 2047$ then value = $(-1)^{\text{sign}} * 2^{(\text{exponent} - 1023)} * (1 + \text{fraction})$

Using Command Numbers

Command numbers can be used as a shortcut for all mnemonics in an HP 5507A/B (serial and parallel port communication only). The command numbers are assigned at bootup according to what boards and addresses are being used in the HP 5507A/B. Thus each system may have different command numbers for the same mnemonic.

To find the command number for a specific mnemonic in a system, the ASGN and CNUM mnemonics are employed. For example, to find the command number for the VWTC mnemonic, the string "ASGN VWTC;CNUM?"CR/LF is sent to the communication board. The board returns an integer number that is the base command number for the mnemonic specified by the ASGN mnemonic (a zero value indicates that the mnemonic does not exist in this particular HP 5507A/B).

This command number must be prefixed by a command number read/execute byte (&H80) or a command number write byte (&HC0). If the command number is less than 256, then it is the second and final byte of the command number sequence. If the command number is greater than 255, then &H00 is the 2nd byte of the sequence and the command number minus 255 is the final byte of the sequence.

For example, if the command number returned is 135 (&H0087), and one wants to send an integer data value (2000 = &H07D0) to this mnemonic, then the sequence of data bytes sent would be &HC0, &H87, &H07, &HD0.

If the command number returned is 295 (&H0127), and one wants to read the value of this mnemonic, the sequence of data bytes sent would be &H80, &H00, &H28.

Mnemonics for the communication board also have a one-byte command number that does not require a command number Prefix byte. These numbers are given in the mnemonics reference section and the mnemonics summary. Note that the internal data format must be used when sending data to any board using a command number.

To re-execute a sequence of command numbers, use the CR/LF sequence to delimit the desired command number sequence. Then send the "?"CR/LF to repeat the commands as described on page 4E-13.

Using Handshake Mode

Handshake mode allows synchronization between an external sample trigger source and the data collection process. The basic operation is as follows:

1) The boards involved are set to external sample mode.

2) The communication channel board is set to handshake on.

3) The position information is requested (one semicolon separated string of mnemonics/command numbers).

4) The control computer attempts to read the data (the communication board will not respond with data until it receives a signal indicating that data is ready).

5) The external sample signal arrives at all boards involved.

6) The boards process the sample and pulse a backplane line when they have updated the position information in their mailboxes.

7) The communication channel board detects the first pulse on the backplane line, reads the requested data, and outputs it to the control computer.

8) Steps 3 though 7 are repeated for each data sample (note that in step 3 a single question mark may replace the string for all but the first data request).

The above sequence is optimized for a two-axis system due to a single line informing the communication board that data is ready (it is ambiguous what data is ready in systems with three or more axes). To use this mode with more than two axes, the communication board's position should be requested first in the semicolon separated string. This setup gives the other boards additional time to update their mailbox before the communication board reads it to send to the control computer.

A-QUAD-B AXIS BOARD MNEMONICS REFERENCE

The following mnemonics are recognized by the A-Quad-B Axis Board. The "*" preceding them denotes the address of the selected board, which can be "S", "T", "U", "V", "W", "X", "Y", or "Z".

Note that the first eleven mnemonics do not have a preceding "*". These mnemonics are "system"mnemonics and are recognized only over the serial and parallel interfaces (some are also recognized by the HP-IB board, but their operation may vary slightly from the descriptions given below - see the corresponding descriptions under the HP-IB board when using them over that data channel).

Mnemonics noted as "Read-only" or "Read/Write" can be read over all data channels by issuing the mnemonic followed by a "?" (e.g., XREV?).

Values can be written to "Read/Write" mnemonics by adding a number suffix to the mnemonic (e.g., XDPD 3.8). Some mnemonics accept floating point numbers while others accept only integers.

Mnemonics noted as "Commands" are not associated with a value that can be read or written. Issuing the command mnemonic causes some action to be taken.

ASGN

Description: Assign Command Number

Type: Read/Write ASCII Data

Range: ASCII Character Set   Power-up Default: " – – – –"

Command Number: read – 179 (&HB3), write – 243 (&HF3)

Internal Format Data Length: 4 Bytes

This mnemonic is used in conjunction with the CNUM mnemonic to determine a command number for a specified mnemonic. When written to, the board tests the four characters sent against an internal list of mnemonics. If a match is found, then CNUM is set to the command number associated with the four character mnemonic.

If no match is found, an error is generated and the value of CNUM is set to zero. The reverse operation is also possible. Writing a value to CNUM will cause the associated mnemonic string to be placed in the ASGN mnemonic for reading.

BOOT

Description: Hard Reset

Type: Command

Command Number: 180 (&HB4)

The BOOT command causes this board to assert the backplane Hard_Reset– line. All boards restart as if power was just applied. See "Hard Reset" on page 4E-40.

CNFG

Description: Configuration

Type: Read-Only ASCII Data

Command Number: read – 181 (&HB5)

The CNFG read-only mnemonic will return a list of the boards currently installed and functional in the HP 5507. Each board's address will be followed by either its name or four dashes (indicates power-up self test failure).

CNUM

Description: Command Number

Type: Read/Write Integer Data

Range: 0 to 360 Power-up Default: 0

Command Number: read – 182 (&HB6), write – 246 (&HF6)

Internal Format Data Length: 2 Bytes

This mnemonic is used in conjunction with the ASGN mnemonic to determine a command number for a specified mnemonic. When written to, the board checks its list of command numbers. If the number sent is found, then the ASGN mnemonic is set to the four character string of the associated mnemonic.

If the number is not found, an error is generated and the value of ASGN is set to "– – – –". The reverse operation is also possible. Writing a value to ASGN will cause the associated command number to be placed in the CNUM mnemonic for reading.

DFMT

Description: I/O Channel Data Format

Type: Read/Write Byte Data

Range: 0 to 5 Factory Setting: 0

Command Number: read – 187 (&HB7), write – 247 (&HF7)

Internal Format Data Length: 1 Byte

This mnemonic specifies what data format this I/O channel will use when transmitting data. The available formats and corresponding format mode value are specified below. Each interface may have a different mode programmed. See "Data Formats" for details on each format.

0 - ASCII for everything.
1 - BLOCK #A for Floating Point, ASCII for everything else.
2 - BLOCK #D for Floating Point, ASCII for everything else.
3 - Internal format for Floating Point, ASCII for everything else.
4 - ASCII for text requests, Internal format for command number requests.
5 - ASCII for ASCII data items, Internal format for everything else.

ECHO

Description: Echo State (serial ports only)

Range:0 to 255 Factory Setting: 1

Command Number: read – 184 (&HB8), write – 248 (&HF8)

Internal Format Data Length: 1 Byte

The ECHO mnemonic turns off (0) or on (non-zero) this I/O channel's echo operation (useful when connected to a serial terminal).

ERML

Descrition: Reserved

Type: Read/Write Byte Data

Range: 0 to 9 Power-up Default: 0

Command Number: read – 185 (&HB9), write – 249 (&HF9)

Internal Format Data Length: 1 Byte

This mnemonic is reserved for future use.

ERRM

Description: System Error Message

Type: Read-Only ASCII Data

Command Number: read – 186 (&HBA)

The ERRM read-only mnemonic will return a sequence of comma (,) separated strings terminated by CR LF. One string for each board in the HP 5507 that has an error condition. A CR LF instead of a comma will be output after the final error message string.

ERST

Description: Soft Reset

Type: Command

Command Number: 187 (&HBB)

This command causes the board to hold the backplane Soft_Reset– line low for ~500 ms. Each A-Quad-B Axis board will respond to this line as if it had received an Axis Error Reset command (*AER). See "Soft Reset" on page 4E-40.

HSMD

Description: Handshake Mode

Type: Read/Write Byte Data

Range: 0 or 1 Power-up Default: 0

Command Number: read – 188 (&HBC), write – 252 (&HFC)

Internal Format Data Length: 1 Byte

The HSMD mnemonic turns on/off the handshake feature which can be used to synchronize external samples with data transfers over the serial and parallel data channels. HSMD 0 (off) is the power-up default mode. See the "Using Handshake Mode" section for details.

INST

Description: System Mnemonic List

Type: Read-Only ASCII Data

Command Number: 189 (&HBD)

This read-only mnemonic returns a list of the mnemonics associated with all the boards currently installed and operational in this HP 5507. Each request returns an 80-character string of the next group of 15 mnemonics. A soft reset will set the internal list pointer to the first mnemonic.

ISTA

Description: System Status Byte

Type: Read-Only Byte Data

Command Number: read – 190 (&HBE)

Internal Format Data Length: 1 Byte

This read-only mnemonic returns an 8-bit status value as shown below. The bits correspond to those of the HP-IB board's ISTA mnemonic with the exception of bits 0, 1 and 6.

Compensation Alert	Not Used Always 0	Error	Ready	Not Used Always 0	Path Error	Reserved (Ref. Absent)	Reserved (Sys. Null)
D7 128	D6 64	D5 32	D4 16	D3 08	D2 04	D1 02	D0 01
MSB ←							→ LSB

*AER

Description: Axis Error Reset (this axis only)

Type: Command

Command Number: 129 (&H81)

This command clears any errors on this axis only. If a Path Error was present, then the pulse output electronics will be initialized as if the *AGO command had been issued. Note that only this board is affected. For complete details, see "Soft Reset" on page 4E-40.

*AGO

Description: Axis Go

Type: Command

Command Number: 130 (&H82)

This command causes the board to initialize its hardware and then enable counting. If there is an inconsistency in a board's setup parameters, then an error message will be generated and output pulses will not be enabled. See "Axis Initialize" for additional information.

*AVS

Description: Reserved

Type: Read/Write Floating Point Data

Range: TBD Factory Setting: 0

Command Number: read – 131 (&H83), write – 195 (&HC3)

Internal Format Data Length: 8 Bytes

This mnemonic is reserved for future use. The setup enable switch must be in the enable position to change this mnemonic's value.

*BCN

Description: Base Compensation Number

Type: Read/Write Floating Point Data

Range: 0.99 to 1.01 Factory Setting: 0.999728766

Command Number: read – 132 (&H84), write – 196 (&HC4)

Internal Format Data Length: 8 Bytes

This mnemonic reads or writes the base compensation number. This base number is set by the user through this mnemonic, or updated through the hardware update line and other compensation update modes (see "Using Compensation" section).

If mnemonics are stored in the compensation mnemonic array at *CMP = 5 through 9, then this board will send any value written to *BCN to the mnemonics specified.

*CMD

Description: Compensation Mnemonic Data Value

Type: Read/Write Floating Point Data

Range: $\pm 10^{308}$ Factory Setting: 0

Command Number: read – 133 (&H85), write – 197 (&HC5)

Internal Format Data Length: 8 Bytes

This mnemonic sets any data values that are associated with the compensation mnemonic set using the *WCM mnemonic. The *CMP mnemonic specifies which mnemonic the data is associated with. See the "Using Compensation" section for additional details.

The setup enable switch must be in the enable position to change this mnemonic's value.

*CMP

Description: Compensation Mnemonic Pointer

Type: Read/Write Byte Data

Range: 0 to 9 Power-up Default: 0

Command Number: read – 134 (&H86), write – 198 (&HC6)

Internal Format Data Length: 1 Byte

This mnemonic works in conjunction with the *WCM, *CMD and *RCM mnemonics. The value written to this mnemonic selects which compensation mnemonic and data are accessed by the above mnemonics as follows:

- 0 - Compensation Number Mnemonic - HP 5507A/B mnemonic used to read/write the compensation number - mnemonic only, no data value.
- 1 - Compensation Limit Mnemonic - HP 5507A/B mnemonic used to write the limit value to the compensation board when the update mode is set to minus 1 - mnemonic and floating point data.
- 2 - Get Initial Compensation Number Mnemonic - HP 5507A/B mnemonic used to read the initial compensation number from a compensation board - mnemonic only, no data value.
- 3 - Compensation Filter Time Constant Mnemonic - HP 5507A/B mnemonic used to write the filter time constant to the compensation board - mnemonic and floating point data.
- 4 - Coefficient of Thermal Expansion Mnemonic - HP 5507A/B mnemonic used to write the coefficient of thermal expansion to the compensation board - mnemonic and floating point data.
- 5 through 9 - Compensation Value Write Mnemonics - HP 5507A/B mnemonics that will be sent the new compensation value after an update compensation operation - mnemonics only, no data values.

Refer to the "Using Compensation" section for additional details.

*COF

Description: Compensation Offset (in ppm)

Type: Read/Write Floating Point Data

Range: ±100 Factory Setting: 0

Command Number: read – 135 (&H87), write – 199 (&HC7)

Internal Format Data Length: 8 Bytes

The *COF mnemonic specifies the compensation offset value to add to the base compensation number for this axis (each axis can have a different value). The resulting number is then used as the total compensation number for all calculations on this board. Its value is expressed in parts per million (ppm). For example, a value of –10.3 would change the base compensation number of 0.9997380 to a total compensation number of 0.9997277.

*CUR

Description: Compensation Update Rate/Mode/Command

Type: Read/Write Integer

Range: –327 to 100 Factory Setting: 0

Command Number: read – 136 (&H88), write – 199 (&HC8)

Internal Format Data Length:

The compensation number (*BCN) may be updated in several ways (see the "Using Compensation" section). This mnemonic's value specifies when it will be updated as follows:

- Manual: A value of zero specifies manual updates and also updates the compensation number using the specified mnemonic (see *WCM and *CMP = 0).
- Limit: A value of minus one specifies that the compensation number be updated only when it exceeds a specified limit (see *WCM, *CMD, *CMP = 0, 1 and the HP 10946B/C documentation).
- Periodic: Values less than minus one specify how many seconds to wait between updates using the specified mnemonic (see *WCM and *CMP = 0). For example, a value of –5 indicates to update the compensation number every five seconds. The maximum wait time is 327 seconds (5:27).
- Frequent: Values greater than zero specify the frequency at which to update the compensation number using the specified mnemonic (see *WCM and *CMP = 0). For example, a value of 3 indicates to update the compensation number three times per second. Maximum value is 100 Hz.

*DIR

Description: Direction Sense

Type: Read/Write Byte Data

Range: 0 or 1 Factory Setting: 0

Command Number: read – 137 (&H89), write – 201 (&HC9)

Internal Format Data Length: 1 Byte

The orientation of the optics and the laser head determine the direction sense for the laser position counter. Some configurations cause this direction sense to be backwards (position decreasing as optics get further apart). The *DIR mnemonic corrects this condition. Change the value of *DIR if *POS decreases as the optics move further apart. Failing to set this value correctly causes the deadpath correction to be invalid.

The setup enable switch must be in the enable position to change this mnemonic's value.

*DPD

Description: Deadpath Distance

Type: Read/Write Floating Point Data

Range (plane mirror optics): 0 to 10500 mm Factory Setting: 0
0 to 410 inches

Command Number: read – 138 (&H8A), write – 202 (&HCA)

Internal Format Data Length: 8 Bytes

This mnemonic sets the deadpath distance in the selected I/O units (inches for *PUN = 1, mm for *PUN = 0). The deadpath is defined as the measurement path length minus the reference path length at the normal reset position (see *Figure 4E-2*, Deadpath & Preset Distance).

Note that the given range is for plan mirror optics. The range doubles for linear optics, and is cut in half for high resolution optics.

The setup enable switch must be in the enable position to change this mnemonic's value.

*EMD

Description: Error Mode

Type: Read/Write Byte Data

Range: 0, 1, 2, or 3 Factory Setting: 0

Command Number: read – 139 (&H8B), write – 203 (&HCB)

Internal Format Data Length: 1 Byte

The *EMD mnemonic specifies how the board responds to various error conditions as follows:

0 - Only Measurement Errors on this board will disable the pulse outputs.
1 - Any error (measurement or programming) on this board will disable the pulse outputs.
2 - Measurement errors on any board will disable this board's pulse outputs.
3 - Any HP 5507A/B error will disable this board's pulse outputs.

*KHZ

Description: Pulse Output Rate (in kHz)

Type: Read/Write Integer Data

Range: 781 to 5154 Factory Setting: 781

Command Number: read – 140 (&H8C), write – 204 (&HCC)

Internal Format Data Length: 2 Bytes

The *KHZ mnemonic specifies the output pulse rate in kHz. The board, however, has a limited set of data rates. It will pick the one from this set that is closest to that specified, but not greater than it. Reading the value of this mnemonic after writing to it will return the value the board will use (781, 805, 831, 859, 889, 920, 955, 991, 1031, 1074, 1121, 1171, 1227, 1289, 1356, 1432, 1516, 1611, 1718, 1841, 1982, 2148, 2343, 2577, 2864, 3221, 3682, 4295, or 5154).

The setup enable switch must be in the enable position to change this mnemonic's value.

*LBV

Description: Last Bad Value

Type: Read Only Floating Point Data

Command Number: read – 141 (&H8D)

Internal Format Data Length: 8 Bytes

The *LBV mnemonic returns the data value that generated the last range error message. Please note that the ASGN and CNUM errors do not change the *LBV value.

*LBV (Continued)

Also note that block-data format errors (caused by serial and parallel port communications) do not change the *LBV value returned through the HP-IB port, but they do change the *LBV value returned through the serial and parallel ports. If the output data format is ASCII, then improper floating point numbers will return "-?" and values corresponding to ± ∞ will return "– – –". Setting the output mode to internal format causes the actual data bytes to be returned.

*LHF

Description: (Reserved)

Type: Read Only Integer Data

Command Number: read – 142 (&H8E)

Internal Format Data Length: 2 Bytes

This mnemonic is reserved for future use.

*MEN

Description: External Multiplier Sample Enable

Type: Read/Write Byte Data

Range: 0 or 1 Factory Setting: 0

Command Number: read – 143 (&H8F), write – 207 (&HCF)

Internal Format Data Length: 1 Byte

This mnemonic enables (set to 1) or disables (set to 0) the external multiplier sample line: (X_MULT_SAMPLE–).

*MEX

Description: Select External Measurement Source

Type: Command

Command Number: 144 (&H90)

This mnemonics selects the HP 10780 receiver as the input to the measurement electronics (factory setting). See *MIN.

The setup enable switch must be in the enable position for this command to have any effect.

*MIN

Description: Select Internal Measurement Source

Type: Command

Command Number: 145 (&H91)

This mnemonics selects the receiver in the HP 5518A Laser Head as the input to the measurement electronics (single axis systems only). See *MEX.

The setup enable switch must be in the enable position for this command to have any effect.

*MLE

Description: Multiplier Output (external sample)

Type: Read Only Floating Point Data

Command Number: read – 146 (&H92)

Internal Format Data Length: 8 Bytes

This read-only mnemonic is like the *MUL mnemonic, however it is only updated by external multiplier samples initiated by a falling edge on the X_MULT_SAMPLE– line. The units and data format are the same as *MUL. The external sample line must first be enabled with a *MEN1 command.

*MPO

Description: Maximum Position Offset

Type: Read/Write Floating Point Data

Range: 0 - 100 mm Factory Setting: 0.1
0 - 4 in

Command Number: read – 147 (&H93), write – 211 (&HD3)

Internal Format Data Length: 8 Bytes

The *MPO mnemonic sets the maximum position offset value the board should allow. Setting this value maximizes the electronic's range while still allowing for adequate geometric correction. The *PUN mnemonic specifies the units as mm (0) or inches (1). (Refer to the "Geometric Correction" section for further details.)

The setup enable switch must be in the enable position to change this mnemonic's value.

*MSW

Description: Board Miscellaneous Status Word

Type: Read Only Integer Data

Command Number: read – 148 (&H94)

Internal Format Data Length: 2 Bytes

The board's 16-bit status register is returned by this read-only mnemonic. Each bit in the register is defined as follows:

ERROR STATUS							
No Measure Signal	Laser Counter Overflow	Slew Rate Exceeded	Measure Signal Glitch	Hardware Circuits Underflow	Pulse Circuits Error	System Path Error	System Error
D15	D14	D13	D12	D11	D10	D9	D8
–32768	16384	8192	4096	2048	1024	512	256
MSB →							

BOARD SETUP STATUS							
External Sample Enabled		Measurement Source Internal	Pulse Outputs Disabled	Sample Mode External		Data Channel Code	
*MUL	*POS			*MUL	*POS		
D7	D6	D5	D4	D3	D2	D1	D0
128	64	32	16	08	04	02	01
←							LSB

A 1 in any bit position indicates a "true" for the corresponding status. For example, a *MSW value of 84 has 1s in bit positions D6, D4 and D2, indicating *POS sample mode is external (D2), the pulse outputs are disabled (D4), and the *POS external sample is enabled (D6). The 0s in this example indicate that there are no errors, the *MUL sample mode is not external, the measurement source is not internal, and external *MUL sampling is disabled.

The Data Channel Code indicates which data channel generated the last error. The code is 0 for no errors, 1 for serial port A, 2 for serial port B, and 3 for the parallel port.

*MUL

Description: Multiplier Output (internal sample)

Type: Read Only Floating Point Data

Command Number: read – 149 (&H95)

Internal Format Data Length: 8 Bytes

The *MUL read-only mnemonic returns the value that is currently being sent to the pulse conversion electronics. It is updated as specified by the *MUR mnemonic. Its units equal the output pulse units as set by the *PUN and *RES mnemonics.

The value returned includes environmental compensation, deadpath, preset distance (if selected), position offset and motion. For example, if deadpath = 15 mm, preset = 20 mm, position offset = 0.01 mm, pulse resolution = 0.00001, and the optics have moved 8 mm in the minus direction, then *MUL would equal

$$\frac{(15 + 20 - 0.01 - 8)}{0.00001} = 269900$$

*MUR

Description: Multiplier (*MUL) Update Rate/Mode/Command

Type: Read/Write Integer Data

Range: -327 to 1000 Factory Setting: 0

Command Number: read – 150 (&H96), write – 214 (&HD6)

Internal Format Data Length: 2 Bytes

The value returned with the *MUL mnemonic may be updated in several ways. The *MUR mnemonic's value specifies which one as follows:

- Manual: A value of zero specifies update once immediately.
- External: A value of minus one specifies that the *MUL value be updated only by the external sample input line (must be enabled with *MEN1).
- Periodic: Values less than minus one specify how many seconds to wait between updates. For example, a value of –5 indicates to update every five seconds.
- Frequent: Values greater than zero specify the frequency at which to update the *MUL value. For example, a value of 3 indicates to update three times per second.
- Internal: Values greater than 100 specify to use an internal software loop to update the *MUL value. The rate is about 800 Hz.

*NAM

Description: Board Name

Type: Read Only ASCII Data

Command Number: read – 151 (&H97)

This read-only mnemonic returns the four ASCII characters corresponding to the board type. For this board they are "QUAD".

*OPT

Description: Optics Type

Type: Read/Write Byte Data

Range: 0 to 2 Factory Setting: 0

Command Number: read – 152 (&H098), write – 216 (&HD8)

Internal Format Data Length: 1 Byte

The *OPT mnemonic specifies which optics are used with this axis. A value of zero corresponds to linear and single beam optics. A value of one selects plane mirror optics. A value of two selects high resolution optics.

The setup enable switch must be in the enable position to change this mnemonic's value.

*PEN

Description: External Position Sample Enable

Type: Read/Write Byte Data

Range: 0 or 1 Factory Setting: 0

Command Number: read – 153 (&H99), write – 217 (&HD9)

Internal Format Data Length: 1 Byte

This mnemonic enables (set to 1) or disables (set to 0) the external position sample line: (X_RAW_SAMPLE–).

*POF

Description: Position Offset

Type: Read/Write Floating Point Data

Range: ± *MPO   Power-up Default: 0

Command Number: read – 154 (&H9A), write – 218 (&HDA)

Internal Format Data Length: 8 Bytes

The *POF mnemonic sets the current geometric correction value for this axis. This value will be added to the position (*POS) and to the output pulse stream. The units are selected by the *PUN mnemonic. (Refer to the "Geometric Correction" section for further details.)

*POS

Description: Position Output Variable (internal sample)

Type: Read Only Floating Point Data

Command Number: read – 155 (&H9B)

Internal Format Data Length: 8 Bytes

The *POS read-only mnemonic returns the current position in the units specified by *PUN. If *PUN equals 1, then the value is in inches. If *PUN equals 0 then the value is in mm. The *POS value is updated either at a specified rate, by the external sample line or by an internal software loop. The *PUR mnemonic selects the update mode. The value returned includes environmental compensation, deadpath correction, and geometric correction.

*PRD

Description: Preset Distance

Type: Read/Write Floating Point Data

Range (plane mirror optics) : ±10500 mm ±410 inches   Factory Setting: 0

Command Number: read – 156 (&H9C), write – 220 (&HDC)

Internal Format Data Length: 8 Bytes

The *PRD mnemonic reads or writes the preset value in the units specified by the *PUN mnemonic. This value is used in conjunction with the PRESET_ENABLE– input line to facilitate initializing the laser system when the stage is not at its normal reset location (should there be a path error for example).

A mechanical/optical switch that closes when the stage is away from the normal reset position can be connected to the PRESET_ENABLE– input line to automatically enable the preset value. The *PRD value can be set by the operator to the approximate stage location, or at the factory to the maximum stage position.

When the laser system is initialized with the enable line activated, the *PRD value is included in the *POS and *MUL values. The net result being that the stage can now move back to the normal reset position without generating a laser underflow (the laser system generates an underflow when the stage moves in the minus direction a distance greater than the programmed deadpath value - plus preset value if it was enabled).

*PSE

Description: Position Output Variable (external sample)

Type: Read Only Floating Point Data

Command Number: read – 157 (&H9D)

Internal Format Data Length: 8 Bytes

This read-only mnemonic is like *POS, except that it is only updated by an external sample operation. The units are the same as *POS. The external sample line must first be enabled with a *PEN1 command.

*PUN

Description: Pulse Output Units (0=mm, 1=inches)

Type: Read/Write Byte Data

Range: 0 or 1 Factory Setting: 0

Command Number: read – 158 (&H9E), write – 224 (&HDE)

Internal Format Data Length: 1 Byte

The *PUN mnemonic specifies the I/O units and what units to use for the output pulses. A value of zero selects millimeters and a value of one selects inches. Mnemonics that use *PUN are *AVS, *DPD, *MLE, *MPO, *MUL, *POF, *POS, *PRD, *PSE, and *RES.

The setup enable switch must be in the enable position to change this mnemonic's value.

*PUR

Description: Position (*POS) Update Rate/Mode/Command

Type: Read/Write Integer Data

Range: -327 to 1000 Factory Setting: 1000

Command Number: read – 159 (&H9F), write – 223 (&HDF)

Internal Format Data Length: 2 Bytes

The value returned with the *POS mnemonic may be updated in several ways. The *PUR mnemonic's value specifies which one as follows:

- Manual: A value of zero specifies update once right now.
- External: A value of minus one specifies that the *POS value be updated only by the external sample input line (must be enabled with *PEN1).
- Periodic: Values less than minus one specify how many seconds to wait between updates. For example, a value of –5 indicates to update every five seconds. Maximum wait time is 327 seconds (5:27).
- Frequent: Values greater than zero specify the frequency at which to update the *POS value. For example, a value of 3 indicates to update three times per second.
- Internal: Values greater than 100 specify to use an internal software loop to update the *POS value. The rate is about 800 Hz.

*QAD

Description: Quadrature Output Format Select

Type: Read/Write Byte Data

Range: 1 or 0 Factory Setting: 0

Command Number: read – 160 (&HA0), write – 223 (&HE0)

Internal Format Data Length: 1 Byte

The *QAD mnemonic specifies what output format to use for the "pulses". A value of 0 selects up and down pulses. A value of 1 selects A-quad-B signals.

The setup enable switch must be in the enable position to change this mnemonic's value.

*RCM

Description: Read Compensation Mnemonic

Type: Read Only ASCII Data

Command Number: read – 161 (&HA1)

The *RCM read-only mnemonic outputs the ASCII mnemonic associated with the current value of the *CMP mnemonic. These mnemonics are specified using the *WCM mnemonic to customize each system to the compensation system in use. See the "Using Compensation" section for additional details.

*RES

Description: Pulse Resolution

Type: Read/Write Floating Point Data

Range: 0.0000001 to 0.1 or –0.0000001 to –0.1 Factory Setting: 0.00001

Command Number: read – 162 (&HA2), write – 226 (&HE2)

Internal Format Data Length: 8 Bytes

This mnemonic specifies the output pulse resolution in the units selected by the *PUN mnemonic. For example, a value of 0.00001 specifies 10 μinch pulses when *PUN is set to one, and 0.01 micron pulses when *PUN is set to zero. Changing the sign of *RES swaps A/up output lines with the B/down output lines, reversing the direction as sensed by the CNC/NC controller.

The setup enable switch must be in the enable position to change this mnemonic's value.

*REV

Description: Software Revision Date Code

Type: Read Only Integer Data

Command Number: read – 163 (&HA3)

Internal Format Data Length: 2 Bytes

This read-only mnemonic returns a four digit date code indicating the year and week the firmware was last updated. The base year is 1960, so a date code of 3040 indicates week 40 of 1990.

*RLP

Description: Raw Laser Position

Type: Read Only Floating Point Data

Command Number: read – 164 (&HA4)

Internal Format Data Length: 8 Bytes

This read-only mnemonic returns the value in the laser position counter. No compensation, or units conversion takes place. The units are $\lambda/64$ for linear optics, $\lambda/128$ for plane mirror optics, and $\lambda/256$ for high-resolution optics. The value is intended for testing only.

*RSV

Description: Raw Subtraction Value

Type: Read Only Floating Point Data

Command Number: read – 165 (&HA5)

Internal Format Data Length: 8 Bytes

This read-only mnemonic returns the value that the hardware sends to the multiplier. No compensation or units conversion takes place. The units are the same as *RLP. The value is intended for testing only.

*SAM

Description: Serial Port A Parity Mode and Stop Bits

Type: Read/Write Byte Data

Range: 0 to 255 Factory Setting: 24

Command Number: read – 166 (&HA6), write – 230 (&HE6)

Internal Format Data Length: 1 Byte

This mnemonic specifies parity and stop bits for serial port A. Changes are not effective until the *SAV command executes, followed by a hard reset (BOOT) operation (see "Setup Enable" in Section III). When the setup enable switch is in the "enable" position, serial port A is set to 8-bits per character, even parity, 1.5 stop bits and echo on. A "1" in a given bit selects the corresponding option. The available settings are defined as follows:

No Echo	No Parity	Odd Parity	Eight Bit Characters	Code For Number of Stop Bits			
D7 128	D6 64	D5 32	D4 16	D3 8	D2 4	D1 2	D0 1
MSB ←							→ LSB

The stop-bit code (0 through F Hex) selects the number of stop bits as follows:

0 - 0.563	4 - 0.813	8 - 1.563	C - 1.813
1 - 0.625	5 - 0.875	9 - 1.625	D - 1.875
2 - 0.688	6 - 0.938	A - 1.688	E - 1.938
3 - 0.750	7 - 1.000	B - 1.750	F - 2.000

The ECHO mnemonic may be used to immediately turn echo off and on.

*SAR

Description: Serial Port A Baud Rate

Type: Read/Write Integer Data

Range: 110 to 19200 Factory Setting: 2400

Command Number: read – 167 (&HA7), write – 231 (&HE7)

Internal Format Data Length: 2 Bytes

This mnemonic specifies the serial port A baud rate in Hz. The board will select the closest one it has and then set the baud rate to that value. The baud rate does not change until after a *SAV command and then a hard reset (BOOT).

Note that the baud rate for serial port A is set by switches when the setup enable switch is in the enable position (see "Setup Enable" in Section III).

*SAV

Description: Save Setup Values

Type: Command

Command Number: 168 (&HA8)

The *SAV mnemonic saves setup information in the EEPROM. The Board will not accept this command unless the setup enable switch is in the enable position, or the *SAV command immediately follows a *TST 254 command.

When enabled, the *SAV command first verifies compatibility between all setup parameters. If they are OK, then it stores them in an on-board EEPROM. If there is an inconsistency, then an error message will be generated and nothing will be stored. The parameters saved are:

- *AVS - Reserved
- *BCN - Base Compensation Number (sets default on power-up)
- *COF - Compensation Offset (sets default offset on power-up)
- *CUR - Compensation Update Rate/Mode
- *DIR, *OPT - Direction Sense and Optics Type
- *DPD, *PRD - Deadpath and Preset distances
- *EMD - Error Mode (sets default error mode on power-up)
- *KHZ, *PUR, *MUR - Output Pulse Rate, Position and Multiplier Update Rates
- *MEN, *PEN - External Sample Enables
- *MEX, *MIN - Measurement Source
- *MPO - Maximum Position Offset
- *QAD - Output Pulse Format
- *PUN, *RES - Pulse Units and Resolution (also sets default I/O units)
- *SAM, *SAR, *SBM, *SBR - Serial Ports Mode and Data Rates
- *WCM (*RCM), *CMD - Compensation Mnemonics and Data Values

*SBM

Description: Serial Port B Parity Mode and Stop Bits

Type: Read/Write Byte Data

Range: 0 to 255   Factory Setting: 24

Command Number: read – 169 (&HA9), write – 233 (&HE9)

Internal Format Data Length: 1 Byte

This mnemonic specifies parity and stop bits for serial port B. Changes are not effective until the *SAV command executes, followed by a hard reset (BOOT) operation. A "1" in a given bit selects the corresponding option. The available settings are defined as follows:

No Echo	No Parity	Odd Parity	Eight Bit Characters	Code For Number of Stop Bits			
D7 128	D6 64	D5 32	D4 16	D3 8	D2 4	D1 2	D0 1
MSB ←							→ LSB

The stop-bit code (0 through F Hex) selects the number of stop bits as follows:

0 - 0.563	4 - 0.813	8 - 1.563	C - 1.813
1 - 0.625	5 - 0.875	9 - 1.625	D - 1.875
2 - 0.688	6 - 0.938	A - 1.688	E - 1.938
3 - 0.750	7 - 1.000	B - 1.750	F - 2.000

The ECHO mnemonic may be used to immediately turn echo off and on.

*SBR

Description: Serial Port B Baud Rate

Type: Read/Write Integer Data

Range: 110 to 19200   Factory Setting: 2400

Command Number: read – 170 (&HAA), write – 234 (&HEA)

Internal Format Data Length: 2 Bytes

This mnemonic specifies the serial port B baud rate in Hz. The board will select the closest one it has and then set the baud rate to that value. The baud rate does not change until after a *SAV command and then a hard reset (BOOT).

*SGO

Description: System GO (Start)

Type: Command

Command Number: 171 (&HAB)

This command causes the board to assert the backplane Sync_Zero– line for ~500 ms. Each board will initialize its hardware and then enable counting when the Sync_Zero– line is released. If there is an inconsistency in a board's setup parameters, then that board will generate an error message and will not enable the pulse outputs (see "Axis Initialize").

*STA

Description: Board Status Byte

Type: Read Only Byte Data

Command Number: read – 172 (&HAC)

Internal Format Data Length: 1 Byte

This read-only mnemonic returns an unsigned integer indicating the board's status. The value of the integer corresponds to the error numbers associated with the board (see A-Quad-B Axis Board error messages). A status of zero indicates no errors.

NOTE

During the laser head warmup period, the serial and parallel ports will return a non-zero value while the HP-IB port will return zero (the warmup period is the time between a hard reset and the first Soft Reset– issued by the HP-IB board when it detects a stable reference signal).

*STP

Description: Stop

Type: Command

Command Number: 173 (&HAD)

The STOP command (*STP) causes the board to disable its pulse output section. It will continue to track position, which can be read with the *POS mnemonic, but no pulses will be output. The only way to re-enable pulse output is to issue a Start command with either the *SGO or the *AGO mnemonics, or the hardware START– line.

*TST

Description: Test Mode

Type: Read/Write Byte Data

Range: 0 to 7, 254, 255   Power-up Default: 0

Command Number: read – 174 (&HAE), write – 238 (&HEE)

Internal Format Data Length: 1 Byte

This mnemonic replaces the selected measurement signal with a test signal derived from the system clock. The test signal frequency is set to 6/(N+1) MHz. A value of zero selects the specified measurement signal again.

The value of 254 is used to enable the *SAV command when the setup enable switch is in the disable position. The *SAV command must immediately follow the *TST 254 mnemonic (do not eliminate the required item separator though). Any intervening mnemonics will cancel the enable. The *TST mnemonic will retain its prior value after sending *TST 254, and any programmed test frequency will not be changed.

The value of 255 is used to "turn off" the axis circuits on this board. No measurment errors will be generated and pulses will be disabled until a *TST 0 is sent.

*WCM

Description: Write Compensation Mnemonic

Type: Read/Write Integer Data

Range: two packed ASCII characters   Factory Setting: 0

Command Number: read – 175 (&HAF), write – 239 (&HEF)

Internal Format Data Length: 2 Bytes (Requires two data values)

The *WCM mnemonic specifies what mnemonic to use for the various compensation functions selected by *CMP. The mnemonic requires that two data items be written to it. Each data item consists of two ASCII characters packed into one integer value.

For example, the mnemonic VCNV would be converted into the two numbers 22083 and 20054 by multiplying the ASCII code for V by 256 and adding it to the ASCII code for C (see *Table 4E-5* on page 4E-52 for list of ASCII characters and their code values). Then doing the same for N and V. Use the *RCM query to verify the proper mnemonic was set. If the mnemonic is not available in the HP 5507A/B, an error will be generated. See the "Using Compensation" section for additional details.

The setup enable switch must be in the enable position to change this mnemonic's value.

A-QUAD-B BOARD RESET RESPONSE

Hard Reset

The variables and mode conditions listed in *Table 4E-3* are set when the system is forced into the "hard reset" cycle. The user may initialize hard reset by either:

- Power-cycling the HP 5507A/B Laser Transducer (i.e., cycling the ac power switch from on-to-off-to-on).
- Sending the BOOT command (mnemonic or command number) to one of the communication channels.
- Toggling the Internal Master Reset Switch located on the HP-IB board.

Soft Reset

The "soft reset" cycle may be initialized by performing one of the following:

- Depressing the front panel RESET key.
- Sending the ERST command (mnemonic or command number) to one of the interface channels.
- Sending the HP-IB device independent commands Device Clear (DCL) or Selected Device Clear (SDC).
- Asserting the rear panel ERROR_RESET– control line.

The "soft reset" performs the following on any axis without an error:

- Initializes the compensation system (only if not already initialized).
- Instruction Mnemonic Output pointers for INST? query are reset to top (beginning) of list.
- Word Counter for the *WCM mnemonic is reset to the first word.

The "soft reset" performs the following on any axis with an error:

- Status byte (*STA) is set to zero.
- Axis initialized (only if measurement error on this axis).
- Instruction Mnemonic Output pointers for INST? query are reset to top (beginning) of list.
- Word Counter for the *WCM mnemonic is reset to the first word.
- Input and Output Buffers on data channels with data transmission errors are cleared.
- Measurement errors are cleared.
- Programming errors are cleared.
- Data I/O errors are cleared.
- Rear-panel status lines are cleared.

Table 4F-3. A-Quad-B Axis Board Hard Reset Defaults

VARIABLE OR MODE	MNEMONICS	HARD RESET SETS VARIABLE OR MODE TO:
Optics Type	*OPT0,*OPT1,*OPT2	Programmable (*OPT0)
Direction Sense	*DIR0,*DIR1	Programmable (*DIR0)
Measurement Input Source	*MEX,*MIN	Programmable (*MEX)
Pulse and I/O Units	*PUN0,*PUN1	Programmable (*PUN0)
Output Pulse Resolution and Direction	*RES ± Distance	Programmable (*RES0.00001)
Output Pulse Rate	*KHZ Value	Programmable (*KHZ781)
Output Pulse Type	*QAD0,*QAD1	Programmable (*QAD0)
Preset Distance	*PRD Distance	Programmable (*PRD0)
Deadpath Distance	*DPD Distance	Programmable (*DPD0)
Maximum Position Offset	*MPO Distance	Programmable (*MPO0.1)
Starting Compensation Number	*BCN Value	Programmable (*BCN0.999728766)
Compensation Offset	*COF Value	Programmable (*COF0)
Compensation Mode/Rate	*CUR Value	Programmable (*CUR0)
Compensation Mnemonic	*CMP0;*WCM Value;Value	Programmable (undefined)
Compensation Limit Mnemonic	*CMP1;*WCM Value;Value;*CMD Value	Programmable (undefined)
Initial Compensation Mnemonic	*CMP2;*WCM Value;Value	Programmable (undefined)
Filter Time Constant	*CMP3;*WCM Value;Value;*CMD Value	Programmable (undefined)
Coefficent of Thermal Expansion	*CMP4;*WCM Value;Value;*CMD Value	Programmable (undefined)
Error Mode	*EMD0,*EMD1,*EMD2,*EMD3	Programmable (*EMD0)
*MUL Update Rate/Mode	*MUR Value	Programmable (*MUR0)
*POS Update Rate/Mode	*PUR Value	Programmable (*PUR1000)
Multiplier External Sample Enable	*MEN0,*MEN1	Programmable (*MEN0)
Position External Sample Enable	*PEN0,*PEN1	Programmable (*PEN0)
Serial Ports' Mode	*SAM Value, *SBM Value, ECHO Value	Programmable (24), (1)
Serial Ports' Baud Rate	*SAR Value, *SBR Value	Programmable (2400)
Reserved	*AVS Value	Programmable (Zero)
Position	*POS	Zero
Multiplier Output	*MUL	Zero
Position Value – Externally Sampled	*PSE	Zero
Multiplier Value – Externally Sampled	*MLE	Zero
Raw Laser Position	*RLP	Zero
Raw Subtraction Value	*RSV	Zero
Position Offset	*POF Distance	Zero
Compensation Mnemonic Pointer	*CMP Value	Zero
Last Bad Value	*LBV	Zero
Status	*STA	Zero
Miscellaneous Status Word	*MSW	Varies with power-up status
Test Frequency	*TST	Zero
Input Buffers	- - - -	Cleared
Output Buffers	- - - -	Cleared
I/O Data Formats	DFMT	Zero
Instruction Mnemonic Output Pointers	INST	Set to top of list
Sample Handshake Modes	HSMD Value	Zero
Instrument Status Byte	ISTA	Zero
Command Number	CNUM	Zero
Assign Cmd Number	ASGN	Set to "- - - -"

Axis Initialize

The "axis initialize" cycle may be initiated by performing one of the following:

- Issuing a "soft reset" while the axis has a measurement error,
- Asserting the rear panel START– control line,
- Asserting the rear panel AXIS_INITIALIZE– control line,
- Issuing the *SGO command (mnemonic or command number) to any A-Quad-B Axis Board, or
- Issuing the *AGO command (mnemonic or command number) to this A-Quad-B Axis Board.

The "axis initialize" performs the following:

- Clears programming errors,
- Clears measurement errors,
- Checks setup consistency (only if data changed),
- Updates compensation number.
- Resets the axis hardware,
- Sets the preset value (only if PRESET_ENABLE– is asserted),
- Enables output pulses,
- Updates the status bytes (*STA and *MSW).

ERROR MESSAGES

The error messages, combined with the front panel annunciators (LEDs), provide assistance with both system programming and hardware problems. LED indications and sequences are covered briefly on pages 4B-25, 7-17, and 7-30 of the HP 5527A/B Designer's Guide.

Error Indications

The SYSTEM ERROR LED remains off if the system operates properly. If an error is detected in the system — be it a hardware, programming or data entry error — the HP 5507A/B's operation is not suspended. Internal software enables the user to interrogate the system via the controller as to the source of the error. The ERRM? data request returns an ASCII data item that contains the following information:

- Error source information
- Error number
- Short description of the error

After eliminating the cause of the error, the error message can be cleared by initiating a system "soft reset". Out of range values for the A-Quad-B Axis board can be read back through the last bad value mnemonic (*LBV).

A-Quad-B Axis Board Error Messages

-500: Unidentified System Error *STA value: 0 (error not on this board)

This error occurs when the backplane Error- line is pulled low and all function boards have a zero *STA status byte. This will occur when the HP-IB board or user-installed equipment pulls the error line low, or in the event of a mailbox error or line driver hardware failure on any board.

-102: Card Self-test Failure *STA value: 254

A-Quad-B Axis board hardware errors found during the power-up self test. The serial and parallel ports may not be operating if the board fails its self test.

-101: System Error *STA value: 0 (error not on this board)

The A-Quad-B Axis board sources this error upon finding a malfunction on another board's mailbox. This could happen when their is a hardware problem. Check all boards in the backplane.

520: Input Format Error *STA value: 20

An attempt was made to enter an input string that contained a syntax/format error.

521: Command Number Out of Range *STA value: 21

An attempt was made to execute a command outside the permissible command number range.

522: Extended Command Number Error *STA value: 22

The A-Quad-B Axis Board produces this error when an entered extended command number is out of range or fails to execute.

525: Numeric Input Format Error *STA value: 25

An attempt was made to enter a numeric data string that contained a syntax/format error.

526: Numeric Entry Out of Range *STA value: 26

An attempt was made to enter a numeric data string outside the permissible numeric data range.

527: Block Input Format/Range Error *STA value: 27

An attempt was made to load block input data that contained a syntax/format error, contained data input that was out of range, or both. Reading the *LBV value may help determine the cause of the error.

528: Missing Item Separator *STA value: 28

An attempt was made to enter consecutive program element items without an item separator.

530: Unrecognized Mnemonic *STA value: 30

An attempt was made to enter a mnemonic that could not be recognized by the command interpreter.

531: Data Mnemonic Used as Command *STA value: 31

An attempt was made to load a data mnemonic without its corresponding data parameters.

532: Command Mnemonic Used with Data *STA value: 32

An attempt was made to load a command mnemonic followed by data parameters.

533: Write to Read-Only Mnemonic *STA value: 33

An attempt was made to write data to a read-only mnemonic.

534: Unknown Data Type *STA value: 34

The data type for a mnemonic on another HP 5507 function board is not consistent with the data types recognized by the A-Quad-B Axis board.

This error generally indicates that a mailbox has been corrupted. If re-booting the system does not eliminate the problem, replace the board that was being accessed by the A-Quad-B Axis board when the error occurred.

535: CNUM Entry Out of Range *STA value: 35

The entered CNUM parameter was outside the permissible CNUM entry range.

536: ASGN Error - Mnemonic Not Found *STA value: 36

The entered ASGN mnemonic is not in the internal list of mnemonics.

537: ERML Entry Out of Range *STA value: 37

An attempt was made to load an ERML mnemonic parameter outside the range of 0-9.

538: DFMT Entry Out of Range *STA value: 38

An attempt was made to load a DFMT mnemonic parameter outside the range of 0 – 5.

539: HSMD Entry Out of Range *STA value: 39

An attempt was made to load an HSMD mnemonic parameter value other than 0 or 1.

540: Measurement Signal Absent *STA value: 40

This error indicates that there was no measurement signal present when the last "soft reset" or "axis initialize" was received.

541: Measurement Loss of Lock *STA value: 41

This error is sourced when a glitch or dropout affects the axis measurement input. The measurement data is invalid when this happens.

542: Maximum Slew Rate Exceeded *STA value: 42

This error indicates that the axis velocity has exceeded the slew rate limit, resulting in invalid measurement data.

543: Position Counter Overflow *STA value: 43

The optics have traveled beyond the range of the electronics.

544: Pulse Circuits Failure - Re-BOOT *STA value: 44

The pulse generation circuits are no longer operational. The A-Quad-B Axis board must be re-booted to re-configure and test these circuits.

545: Pulse Counter Underflow *STA value: 45

The optics have moved too far in the negative direction, causing a negative number to be sent to the pulse generation circuits.

This can happen when *DPD = 0, *POF <0, and the optics move to the reset position.

546: EEPROM Addressing Error *STA value: 46

The board could not successfully read the programmed startup parameters from the onboard EEPROM. The factory settings will be used for all setup information.

547: System Mnemonic Write Data Error *STA value: 47

This error indicates that the command interpreter has failed, or that an internal data structure has been corrupted.

548: Output Data Formatting Error *STA value: 48

This error occurs when the board cannot properly format the output data.

550: Pulse Counter Preset Error *STA value: 50

This error indicates a hardware failure in the pulse generation circuits.

551: Setup Parameters Locked *STA value: 51

An attempt to modify the setup parameters or issue a *SAV command was made without first properly setting the setup enable switch. (*TST 254 can be used to temporarily enable the *SAV command.)

552: WCM Mnemonic Does NOT Exist *STA value: 52

The mnemonic specified by the two *WCM parameters cannot be found in the internal list of mnemonics. Refer to the INST mnemonic for a list of valid mnemonics.

553: Read Comp. Number Failure *STA value: 53

The programmed mnemonic for reading the initial compensation number failed to execute properly while updating the compensation number. The mnemonic associated with *CMP = 0 must be a floating-point mnemonic.

554: Write Expansion Coef. Failure *STA value: 54

The programmed mnemonic for writing the expansion coefficient failed to execute properly when the board tried to initialize the compensation system. The mnemonic associated with *CMP = 4 must be a read/write floating-point mnemonic.

555: Write Compensation Limit Failure *STA value: 55

The programmed mnemonic for writing the compensation limit failed to execute properly when the board tried to initialize the compensation system. The mnemonic associated with *CMP = 1 must be a read/write floating-point mnemonic.

556: Write Filter TC Failure *STA value: 56

The programmed mnemonic for writing the filter time constant failed to execute properly when the board tried to initialize the compensation system. The mnemonic associated with *CMP = 3 must be a read/write floating-point mnemonic.

557: Read Init. Comp. Number Failure *STA value: 57

The programmed mnemonic for reading the initial compensation number failed to execute properly when the board tried to initialize the compensation system. The mnemonic associated with *CMP = 2 must be a floating point mnemonic.

558: Write Init. Comp. Number Failure *STA value: 58

The programmed mnemonic for setting the initial compensation number failed to execute properly when the board tried to initialize the compensation system. The mnemonic associated with *CMP = 0 must be a read/write floating-point mnemonic when a mnemonic is defined for *CMP = 2.

559: SEND *BCN Value Failure *STA value: 59

One of the programmed mnemonics for sending the compensation value to other boards failed to execute properly when the board updated the *BCN value. The mnemonics associated with *CMP = 5 through *CMP = 9 must be read/write floating point mnemonics.

560: CMP Entry Out of Range *STA value: 60

An attempt was made to set the compensation mnemonic pointer to a value outside the range of 0 - 9.

561: DIR Entry Out of Range *STA value: 61

An attempt was made to load a *DIR mnemonic parameter value other than 0 or 1.

562: EMD Entry Out of Range *STA value: 62

An attempt was made to load an *EMD parameter value other than 0, 1, 2, or 3.

563: MEN Entry Out of Range *STA value: 63

An attempt was made to load a *MEN mnemonic parameter value other than 0 or 1.

564: OPT Entry Out of Range *STA value: 64

An attempt was made to load an *OPT mnemonic parameter value other than 0, 1, or 2.

565: PEN Entry Out of Range *STA value: 65

An attempt was made to load a *PEN mnemonic parameter value other than 0 or 1.

567: PUN Entry Out of Range *STA value: 67

An attempt was made to load a *PUN mnemonic parameter value other than 0 or 1.

568: QAD Entry Out of Range *STA value: 68

An attempt was made to load a *QAD mnemonic parameter value other than 0 or 1.

571: TST Entry Out of Range *STA value: 71

An attempt was made to load a *TST mnemonic parameter value other than 0, 1, 2, 3, 4, 5, 6, 7, 254, or 255.

572: CUR Entry Out of Range *STA value: 72

An attempt was made to load a *CUR mnemonic parameter value outside the specified range of –327 to 100.

573: MUR Entry Out of Range *STA value: 73

An attempt was made to load an *MUR mnemonic parameter value outside the specified range of –327 to 1000.

574: PUR Entry Out of Range *STA value: 74

An attempt was made to load a *PUR mnemonic parameter value outside the specified range of –327 to 1000.

575: WCM Entry Out of Range *STA value: 75

An attempt was made to load a *WCM mnemonic parameter that did not consist of two packed ASCII characters.

576: AVS Entry Out of Range *STA value: 76

An attempt was made to load an AVS mnemonic parameter outside the specified range.

577: BCN Entry Out of Range *STA value: 77

An attempt was made to load a *BCN mnemonic parameter value outside the specified range of 0.99 to 1.01.

578: CMD Entry Out of Range *STA value: 78

An attempt was made to load a *CMD mnemonic parameter value outside the specified range of the associated mnemonic.

579: COF Entry Out of Range *STA value: 79

An attempt was made to load a *COF mnemonic parameter value outside the specified range of –100 to 100.

580: DPD Entry Out of Range *STA value: 80

An attempt was made to load a *DPD mnemonic parameter value outside the specified range of 0 to 10500 mm or 0 to 410 inches (plan mirror optics), or the programmed deadpath value is incompatible with the *OPT value.

581: MPO Entry Out of Range *STA value: 81

An attempt was made to load an *MPO mnemonic parameter value outside the specified range of 0-100 mm or 0-4 inches.

582: POF Entry Out of Range *STA value: 82

An attempt was made to load a *POF mnemonic parameter value outside the specified range of ±*MPO.

583: PRD Entry Out of Range *STA value: 83

An attempt was made to load an *PRD mnemonic parameter value outside the specified range of ±10500 mm or ±410 inches (plane mirror optics) or the programmed preset distance is incompatible with programmed *DPD and/or *OPT values.

584: RES Entry Out of Range *STA value: 84

An attempt was made to load an *RES mnemonic parameter value outside the specified range of 0.0000001 to 0.1 or –0.0000001 to –0.1.

585: RES Incompatible with PUN & OPT *STA value: 85

The programmed *RES mnemonic parameter value conflicts with the selected measurement units (*PUN) and optics type (*OPT).

586: WCM Mnemonic not Floating Point *STA value: 86

An attempt was made to set the *WCM mnemonic to amnemonic that is not a floating point data type.

587: Bad Address for WCM Mnemonic *STA Value: 87

An attempt was made to set the *WCM mnemonic to a mnemonic for this board.

590: Serial Port A Buffer Overflow *STA value: 90

An attempt was made to send a data block to serial port A that exceeded the buffer boundries.

591: Serial Port A OverRun Error *STA value: 91

An attempt was made to send a data block to serial port A that exceeded the board's data processing rate.

592: Serial Port A Parity Error *STA value: 92

An attempt was made to send a data block to serial port A that conflicts with its programmed parity, or a data transmission error has occured.

593: Serial Port A Framing Error *STA value: 93

An attempt was made to send a data block to serial port A that conflicts with its programmed data format, or a data transmission error has occured.

594: Serial Port B Buffer Overflow *STA value: 94

An attempt was made to send a data block to serial port B that exceeded the buffer boundries.

595: Serial Port B OverRun Error *STA value: 95

An attempt was made to send a data block to serial port B that exceeded the board's data processing rate.

596: Serial Port B Parity Error *STA value: 96

An attempt was made to send a data block to serial port B that conflicts with its programmed parity, or a data transmission error has occured.

597: Serial Port B Framing Error *STA value: 97

An attempt was made to send a data block to serial port B that conflicts with its programmed data format, or a data transmission error has occured.

598: Parallel Port Buffer Overflow *STA value: 98

An attempt was made to send a data block to the parallel port exceeding the buffer boundries.

599: Fatal Error: SSSS PPPPPPPP TT NN *STA value: 99

This error indicates a firmware execution error caused by a hardware failure.

S's = Status Register, P's = Prgm Counter, T's = Type, N's = Number

Record the number (S, P, T, N) and a list of the last 10 commands sent to the board, then contact HP for assistance.

To clear the error, a "BOOT" command must be issued over HP-IB or the BOOT command number (0B4 Hex) sent over the serial or parllel ports.

MNEMONICS SUMMARY

Table 4E-4. A-Quad-B Axis Board Mnemonics Summary, lists all the mnemonics that the HP 10934A A-Quad-B Axis Board recognizes. The first group of mnemonics (no *) are only available over the serial and parallel ports. Mnemonics followed by an "i" are read/write integer (two byte) data mnemonics. Those followed by an "x" are read/write floating point mnemonics. Those followed by an "a" are read/write ASCII. Those followed by a "b" are read/write byte values. Those followed by "roa", "rox", "roi" or "rob" are read-only ASCII, floating point, integer and byte mnemonics respectively. Those without any suffix are commands.

Note that the data following a command number must be in internal format and not in ASCII format. Also, the command numbers may only be used over the serial and parallel ports.

Table 4E-4. A-Quad-B Axis Board Mnemonics Summary

		COMMAND NUMBERS (HEX)			
MNEMONIC	(NOTES)	COMMAND	READ	WRITE	DESCRIPTION
ASGN a	(3)	—	179 (B3)	243 (F3)	Assign Command Number for Mnemonic
BOOT		180 (B4)	—	—	Hard Reset Command
CNFG roa		—	181 (B5)	—	System Configuration (read only ASCII)
CNUM i	(3)	—	182 (B6)	246 (F6)	Command Number for Mnemonic from ASGN
DFMT b	(3)	—	183 (B7)	247 (F7)	Output Data Format
ECHO b	(3, 4)	—	184 (B8)	248 (F8)	Echo Status/Control
ERML b	(3)	—	185 (B9)	249 (F9)	Reserved
ERRM roa		—	186 (BA)	—	Error Message
ERST		187 (BB)	—	—	Error Reset Command (Soft Reset)
HSMD b	(3)	—	188 (BC)	252 (FC)	Handshaking Mode
INST roa		—	189 (BD)	—	List of System Mnemonics
ISTA rob		—	190 (BE)	—	Instrument Status Byte
*AER		129 (81)	—	—	Axis Error Reset
*AGO		130 (82)	—	—	Initialize Axis
*AVS x	(1, 2)	—	131 (83)	195 (C3)	Reserved
*BCN x	(2)	—	132 (84)	196 (C4)	Base Compensation Number
*CMD x	(1, 2)	—	133 (85)	197 (C5)	Compensation Mnemonic Data Value
*CMP b		—	134 (86)	198 (C6)	Compensation Mnemonic Pointer
*COF x	(2)	—	135 (87)	199 (C7)	Compensation Offset
*CUR i	(2)	—	136 (88)	200 (C8)	Compensation Update Rate/Mode/Command
*DIR b	(1, 2)	—	137 (89)	201 (C9)	Laser Direction Sense
*DPD x	(1, 2)	—	138 (8A)	202 (CA)	Deadpath Distance (Must be ≥0)
*EMD b	(2)	—	139 (8B)	203 (CB)	Error Mode
*KHZ i	(1, 2)	—	140 (8C)	204 (CC)	Output Pulse Rate
*LBV rox		—	141 (8D)	—	Last Bad Value
*LHF roi		—	142 (8E)	—	Reserved
*MEN b	(2)	—	143 (8F)	207 (CF)	Multiplier External Sample Enable
*MEX	(1, 2)	144 (90)	—	—	HP 10780 Measurement Source
*MIN	(1, 2)	145 (91)	—	—	HP 5518 Measurement Source
*MLE rox		—	146 (92)	—	Multiplier Output External Sample

Table 4E-4. A-Quad-B Axis Board Mnemonics Summary (Continued)

MNEMONIC	(NOTES)	COMMAND NUMBERS (HEX) COMMAND	READ	WRITE	DESCRIPTION
*MPO x	(1, 2)	—	147 (93)	211 (D3)	Maximum Position Offset
*MSW roi		—	148 (94)	—	Miscellaneous Status Word
*MUL rox		—	149 (95)	—	Multiplier Output Internal Sample
*MUR i	(2)	—	150 (96)	214 (D6)	*MUL Update Rate/Mode/Command
*NAM roa		—	151 (97)	—	Board Name
*OPT b	(1, 2)	—	152 (98)	216 (D8)	Optics Type
*PEN b	(2)	—	153 (99)	217 (D9)	Position External Sample Enable
*POF x		—	154 (9A)	218 (DA)	Position Offset
*POS rox		—	155 (9B)	—	Position - Internal Sample
*PRD x	(2)	—	156 (9C)	220 (DC)	Preset Distance
*PSE rox		—	157 (9D)	—	Position - External Sample
*PUN b	(1, 2)	—	158 (9E)	222 (DE)	Pulse and I/O Units Select
*PUR i	(2)	—	159 (9F)	223 (DF)	*POS Update Rate/Mode/Command
*QAD b	(1, 2)	—	160 (A0)	224 (E0)	A-quad-B Output Format Select
*RCM roa		—	161 (A1)	—	Read Compensation Mnemonic
*RES x	(1, 2)	—	162 (A2)	226 (E2)	Pulse Resolution and Direction
*REV roi		—	163 (A3)	—	Firmware Revision Date Code
*RLP rox		—	164 (A4)	—	Raw Laser Position
*RSV rox		—	165 (A5)	—	Raw Subtraction Value
*SAM b	(2)	—	166 (A6)	230 (E6)	Serial Port A Mode
*SAR i	(2)	—	167 (A7)	231 (E7)	Serial Port A Baud Rate
*SAV	(1)	168 (A8)	—	—	Save Setup Parameters
*SBM b	(2)	—	169 (A9)	233 (E9)	Serial Port B Mode
*SBR i	(2)	—	170 (AA)	234 (EA)	Serial Port B Baud Rate
*SGO		171 (AB)	—	—	System Go (Start)
*STA rob		—	172 (AC)	236 (EC)	Board Status Byte
*STP		173 (AD)	—	—	Stop Axis Pulses
*TST b		—	174 (AE)	238 (EE)	Test Mode
*WCM i;i	(1, 2)	—	175 (AF)	239 (EF)	Write Compensation Mnemonic

NOTES
1) The setup enable switch must be in the enable position to issue the command or change this value.
2) Value/state stored in non-volatile memory by *SAV command.
3) Serial and parallel ports only.
4) Effective on serial ports only.

ASCII Table

Table 4E-5 provides a partial listing of the ASCII code for use when programming the compensation options with the *WCM mnemonic.

Table 4E-5. Partial Table of ASCII Characters and Their Values

char	dec	hex	char	dec	hex
0 -	48	30	5 -	53	35
1 -	49	31	6 -	54	36
2 -	50	32	7 -	55	37
3 -	51	33	8 -	56	38
4 -	52	34	9 -	57	39
A -	65	41	a -	97	61
B -	66	42	b -	98	62
C -	67	43	c -	99	63
D -	68	44	d -	100	64
E -	69	45	e -	101	65
F -	70	46	f -	102	66
G -	71	47	g -	103	67
H -	72	48	h -	104	68
I -	73	49	i -	105	69
J -	74	4A	j -	106	6A
K -	75	4B	k -	107	6B
L -	76	4C	l -	108	6C
M -	77	4D	m -	109	6D
N -	78	4E	n -	110	6E
O -	79	4F	o -	111	6F
P -	80	50	p -	112	70
Q -	81	51	q -	113	71
R -	82	52	r -	114	72
S -	83	53	s -	115	73
T -	84	54	t -	116	74
U -	85	55	u -	117	75
V -	86	56	v -	118	76
W -	87	57	w -	119	77
X -	88	58	x -	120	78
Y -	89	59	y -	121	79
Z -	90	5A	z -	122	7A

SECTION IV

SUBSECTION F
HP 10936A SERVO-AXIS BOARD

CONTENTS

LIST OF TABLES

LIST OF FIGURES

SECTION IV

SUBSECTION F
HP 10936A SERVO-AXIS BOARD

HP 10936A SERVO-AXIS BOARD DESCRIPTION

The HP 10936A Servo-Axis Board has three main functions. First it is a 16-bit binary interface to the HP 5507A. Second, it monitors a measurement and reference signal to keep track of position. Third, it uses this position information and software-programmable servo constants to generate motor drive signals that close the feedback loop through an external power amplifier.

The 16-bit binary interface connects directly to an HP 9000 Series 200/300 Computer's GPIO interface. It has access to all boards in an HP 5507A. Thus, communication to and from any board is through either a servo-axis board's 16-bit interface, or the HP 5507A's HP-IB interface.

SERVO-AXIS BOARD SETUP

One option from each of the following categories must be selected to properly setup the Servo-Axis board. *Table 4F-1* presents a brief overview, and the following pages describe each category in detail. Throughout the table, the numbers in parenthesis underneath the mnemonics are the corresponding binary interface command numbers. When two numbers are listed, the first is the command, and the second is the data value. Information on specific mnemonics can be found in the Reference Section (pages 4F-29 through 4F-54).

After setting up the Servo-Axis board, the difference equation coefficients and profile parameters must be initialized. This process is described in "Initializing Motion Parameters".

The general operating commands are covered in "Servo System Operation", while more involved capabilities of the Servo-Axis board are discussed in subsequent paragraphs.

Address

The servo-axis board may be set to backplane address S, T, U, V, W, X, Y, or Z by setting a DIP switch at the top edge of the board (*Figure 4F-1*). For proper operation, each servo-axis board must be set to a unique address (see *Table 4A-1*). Throughout this section the selected address is represented by the "*" preceding the HP-IB mnemonic (see page 4F-67 for binary interface commands that access a particular backplane address).

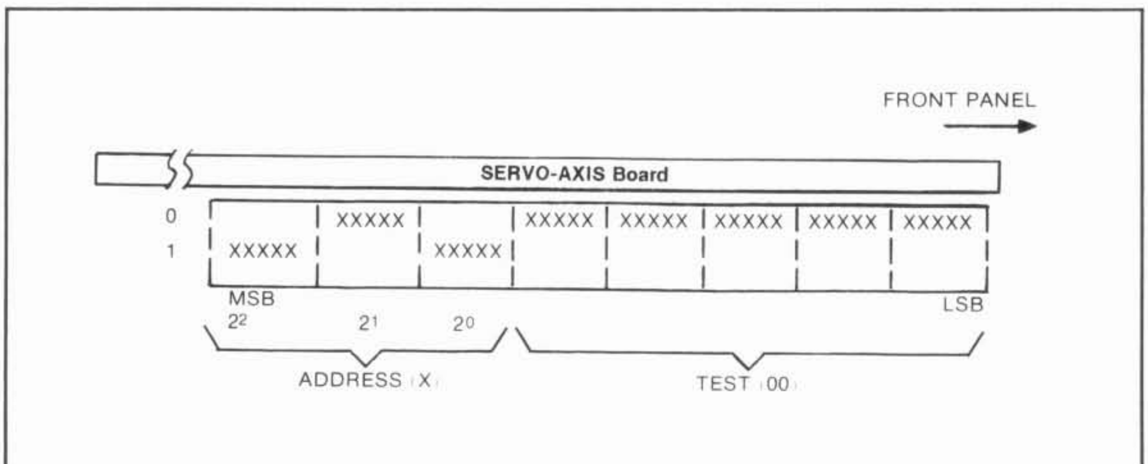

Figure 4F-1. HP 10936A Address Switches

Table 4F-1. Servo-Axis Board Setup Summary

Default	Options	Description
See *Figure 3-1*	S,T,U,V,W,Y,Z	Backplane address: use this letter in place of * in all mnemonics.
*OPT1 (48, 1)	*OPT0, *OPT2 (48, 0) (48, 2)	Selects Optics type: 0 - Linear or Single Beam 1 - Plane Mirror 2 - Special λ/256
*DIR0 (40, 0)	*DIR1 (40, 1)	Selects Direction sense: Set so *POS value increases as reflector moves away from the interfermoeter.
*PTR112;*DTA0 (72, 0)	*PTR112;*DTA1...*PTR112;*DTA7 (72, 1) (72, 7)	Selects Limit Switch Polarity: Bit 0 (low limit) and Bit 1 (high limit): 0 - negative true 1 - positive true Bit 2: 0 - Enable Limits 1 - Disable Limits
*MEX (207)	*MIN (208)	Selects measurement input: *MEX - HP 10780A/B *MIN - HP 5518A
*MET (202)	*ENG, *LAM, *RAW (201) (203) (204)	Selects I/O units: *ENG - Compensated Inches *LAM - Compensated Lambda *MET - Compensated mm *RAW - Uncompensated Lambda
*DPD0 (24,0)	*DPD distance (24, distance)	Specifies the deadpath distance for proper compensation of *POS and *DES values.
*PTR111;*DTA −5nm (70, −5nm)	*PTR111;*DTA distance (70, distance)	Specifies the maximum following error. If | Reference Position − Actual Position | exceeds this value, then the servo will be turned off. Negative values turn off this function.
*INT (206)	*EXT (205)	Selects *POS Sample Trigger: INT - Internal clock EXT - External Sample- line
*EMD0 (44, 0)	*EMD1,*EMD3 (44, 1)...(44, 3)	Selects Error Mode: Drive turned off by: 0,2 - This axis measurement errors 1,3 - Any system error Values 2,3 change Error and Path Error hardware output lines from axis signals to system signals.
Undefined	*PTR106;*DTA value;value (66, value, value)	Specifies mnemonic for *UCN command: see Define Mnemonic for *UCN Command.
*MCM0 (46, 0)	*MCM1 ... *MCM255 (46, 1) ... (46, 255)	Specifies Master/Slave relationship for coordinated multiaxis moves.
*SVC0 (54, 0)	*SVC1 ... *SVC12 (54, 1) ... (54,12)	Selects Position Profiling Method: see Reference Source section.
*CLK0 (36, 0)	*CLK1 ... *CLK4 (36, 1) ... (36, 4)	Selects Servo Clock source: see Servo Clock Source.
*SPD8000 (52, 8000)	*SPD1000 ... *SPD32000 (52, 1000) ... (52, 32000)	Sets Servo Sample Period: period (seconds) = 125 E-9 * value
*DEQ0 (38, 0)	*DEQ1,*DEQ2 (38, 1) (38, 2)	Selects Difference Equation: 0 - Proportional-Integral-Differential 1 - Nth order IIR Filter 2 - Downloaded custom equation
*OUT1 (50, 1)	*OUT0 ... *OUT7 (50, 0) ... (50, 7)	Selects Motor Drive Outputs: Setting a bit enables that output. Bit 0 - ±10 V Analog Bit 1 - Pulse-width Modulated Bit 2 - 16-bit Digital

Optics Type

Three optics types are available. One is selected with the *OPTn mnemonic where n corresponds to the following options:

n	Optics selected (optical resolution)
0:	Linear and Single Beam optics ($\lambda/64$).
1:	Plane mirror optics ($\lambda/128$)
2:	Quad pass optics ($\lambda/256$).

The Servo-Axis board selects n=1 at power up.

Direction Sense

A number of installation factors affect optical direction sense. The *DIR mnemonic matches the electronics to the specified optical configuration so the position value increases as the reflector moves away from the optics. Set *DIR to 0 when F1 is in the measurement path, and set it to 1 when F2 is in the measurement path (see pages 2-8, 5-5, and 5-6). At power up, *DIR is set to 0.

Limit Switch Polarity

Two hardware lines can be used to tell the servo system to stop motion in either the plus or minus direction. The 2 least-significant-bits in *DTA when *PTR equals 112 specify the polarity (positive or negative true) of these input lines (Bit 0 - Low Limit, Bit 1 - High Limit). Setting a bit to 1 specifies positive true. These lines can alternately be used for general purpose input. Setting Bit 2 to 1 disables the limit function (See also *PTR113;*DTA).

Measurement Receiver

Measurement inputs can come from one of two measurement receivers: internal (HP 5518A Laser Head) or external (HP 10780B Receiver). *MIN selects the HP 5518A's receiver; *MEX selects the HP 10780B Receiver.

The Servo-Axis board powers up with the external measurement input selected.

Units

Four distance units are used by the Servo-Axis board when communicating with the control computer. The power-up default is millimeters (*MET). Inches, Lambda, and Uncompensated Lambda units are selected with *ENG, *LAM, and *RAW mnemonics respectively. The Lambda units correspond to the resolution of the optics selected by the *OPTn mnemonic. All units, other than Uncompensated Lambda, are compensated by the *TCN value used for atmospheric and material temperature compensation.

Deadpath Distance

Deadpath can introduce error into a measurement. The Servo-Axis board will minimize this effect by compensating for unequal path length deadpath (See Section V) provided the *DPD mnemonic is set to the appropriate value. *DPD should be set to the distance between the interferometer and the reflector when the stage is at the Home or Zero position.

Following Error

Some situations require the servo system to stop should the following error exceed a preset threshold. Setting the *PTR111;*DTA value to this threshold causes the Servo-Axis board to turn off the servo if |Reference Position − Actual Position| exceeds the threshold. Setting *PTR111;*DTA to a negative value disables this function (power-up default).

Sample Trigger Source

The stage position is always available through the *POS mnemonic, however its value is only updated at specific times as set by the sample trigger source. Two sources are available, internal (*INT) and external (*EXT).

In internal sample mode (power-up default), *POS is continuously updated in the background when the drive is not enabled, and at the Servo Sample rate when it is enabled.

When in external sample mode, *POS is updated by the *PTR109 command or on the falling edge of the External Sample- input.

NOTE

The Motor Drive outputs are not affected by the sample trigger source.

Error Mode

The Servo-Axis board will turn the drive off (set the Motor Drive signals and Drive Enable Out- to 0) under a number of circumstances. The error mode mnemonic (*EMDn) selects one of two response modes.

In *EMD0 or EMD2, only measurement errors from this axis or a *DRE0 command will turn of the servo.

In *EMD1 or *EMD3, any HP 5507A error (other axis error, programming error, etc.) will turn off the servo.

The difference between error modes 0 and 2 (or modes 1 and 3) is modes 2 and 3 change the rear panel Error and Path Error output lines from axis signals to system signals.

Define Mnemonic for *UCN Command

The Servo-Axis board can update its compensation number (*TCN) directly from another function board when this mnemonic is defined. To define a mnemonic, the four ASCII characters corresponding to the desired mnemonic are packed into two one-word integers. These two integers are then sent sequentially to *DTA when *PTR equals 106.

For example, the X Servo-Axis board's *TCN value should come from the compensation board's *WTC mnemonic and the Y Servo-Axis board should use the same value as the X board is using. Sending "XPTR106;DTA22103; 21571" tells the X board to use VWTC (ASCII codes for V, W, T, and C are 86, 87, 84, and 67 respectively, 86*256+87 = 22103 and 84*256+67 = 21571).

Sending "YPTR106;DTA22612; 17230" tells the Y board to use XTCN (ASCII codes for X, T, C, and N are 88, 84, 67, and 78 respectively, 88*256+84 = 22612 and 67*256+78 = 17230).

Now sending XUCN causes the X Servo-Axis board to read the VWTC value and set XTCN equal to it. Similarly sending YUCN causes the Y Servo-Axis board to read the XTCN value and set YCTN equal to it.

Master/Slave Relationships

A Servo-Axis board may be a master over three (or fewer) other Servo-Axis boards. The bits in the value sent to *MCM specify the backplane address(es) of the board(s) this one will control. Both the *CLK and the *SVC mnemonics must be set consistent with *MCM for proper operation of coordinated multiaxis moves.

ADDRESS:	S	T	U	V	W	X	Y	Z
BIT:	7	6	5	4	3	2	1	0
VALUE:	128	64	32	16	8	4	2	1

For example, sending "XSVC1;YSVC2;ZSVC2;XCLK1;YCLK2;ZCLK2;XMCM3" puts the X Servo-Axis board as master of the Y and Z Servo-Axis boards.

Reference Source (Position Profiling Method)

Reference position refers to the profile, or contour, that the stage will follow during a move. The following five sources are available for generating the reference position:

a) Internal position profiling

b) External position profiling

c) Internal velocity profiling

d) External velocity profiling

e) Buffered position profiling

Within each of the Internal and Buffered options are three specific choices, depending on whether the Servo-Axis board is

1) In stand alone operation,

2) A master over other Servo-Axis boards (coordinated multiaxis moves), or

3) A slave to another Servo-Axis board.

Two additional operating modes (buffered post filter injection and buffered filter stimulus) are available to aid in selecting the difference equation coefficients (See Buffered Injection below).

The *SVCn mnemonic selects which option according to the following table:

n	Profile Source
0 -	Internal position profile - stand alone
1 -	Internal position profile - master
2 -	Internal position profile - slave
3 -	External position profile
4 -	Internal velocity profile - stand alone
5 -	Internal velocity profile - master
6 -	Internal velocity profile - slave
7 -	External velocity profile
8 -	Buffered position profile - stand alone
9 -	Buffered position profile - master
10 -	Buffered position profile - slave
11-	Buffered post-filter injection - stand alone
12-	Buffered Filter Stimulus - stand alone

At power up, n is set to 0.

Internal Position Profiling

In internal position profiling stand alone mode, when a *DES is sent, a series of reference positions are generated by the HP 10936A Servo-Axis board that will move the stage to the desired destination while keeping the delta acceleration, acceleration and velocity of the stage within the limits specified by *DAC, *ACC, and *VEL respectively. Internal position profiling master/slave mode operates similarly, the exception is that the master board's delta acceleration, acceleration and velocity limits operate on the vector sum of all the axes involved. Also, the slave boards wait for a signal from the master before starting to move to their requested destinations. Sending *DES to the master initiates the coordinated multiaxis move.

External Position Profiling

In this mode the *DES value becomes the reference position instantly.

Internal Velocity Profiling

Internal velocity profiling operates the same as internal position profiling, except that *DES is ignored and *VEL initiates profiles based on *DAC and *ACC.

External Velocity Profiling

In this mode the *VEL value becomes operational instantly.

Buffered Position Profiling

In this mode, a buffer is filled with incremental position values (see Programming Reference Buffer Operations). These values are linearly interpolated based on the specified interpolation count (*PTR 115), and the resulting values are added to the reference position on a sample by sample basis after a *BGO command is sent. This process continues until the end of the buffer is reached. In master mode, a sync pulse is sent over the inter-servo connector at the start of the move. In slave mode, the board will initiate the move when an inter-servo sync pulse is detected.

Buffered Injection and Filter Stimulus

These modes are useful when determining loop response and tuning the servo system. In either mode, the buffer is filled with the desired test signal (See Programming Buffer Operations Reference). The *BGO command causes these stored values to be injected at the specified location. The post filter (difference equation) injection allows testing closed loop system response or open loop motor-stage response (set coefficients to zero). The filter stimulus allows testing the difference equation response and the open loop system response (in this mode, the only input to the difference equation is the stored values). The Trace function should be used to collect position and difference equation output data with both of these modes. (See Programming Trace Functions).

Servo Clock Source

The Servo Clock source must correspond to the master/slave/stand alone reference source selected above. One of the five sources available is selected by sending the *CLKn mnemonic. The meaning of each n value is as follows:

n	Operation
0:	Internal Stand Alone. Makes this Servo-Axis board independent of any other board. This HP 10936A's Servo Clock is generated internally. Default value.
1:	Internal Master. This Servo-Axis board generates the Servo Clock for other Servo-Axis boards. The Servo Clock is sent over the top inter-servo connector.
2:	Slave to another HP 10936A. This Servo-Axis board receives its Servo Clock from another Servo-Axis board, through the top inter-servo connector.
3:	External Stand Alone. This Servo-Axis board is independent of other boards, and receives its Servo Clock through the Rear Panel connector.
4:	External Master. This Servo-Axis board receives its Servo Clock through the Rear Panel connector, and drives the inter-servo connector with the same signal for slave Servo-Axis boards to use.

Servo Sample Period

The internal Servo Clock's frequency can be programmed from 250 Hz to 8 kHz with the *SPDn mnemonic. The value of n specifies how many 125 ns clock periods are in each Servo Sample period. The following equation shows this relationship:

n = 8E+6 / desired frequency

For Master/Slave systems, set *SPD to the same value for all coordinated boards.

At power up, n is set to 8000, corresponding to a 1 kHz update rate.

Difference Equation

The difference equation is used when calculating the motor drive signals from the position error values (reference position minus current position – see *Figure 3-16a*). One of three equations may be selected with the *DEQn mnemonic.

n	Equation Selected
0	Proportional-Integral-Differential (PID)
1	Nth order infinite impulse response (IIR) filter (9th order maximum)
2	Customer downloaded custom equation

The most intuitive and the easiest to use equation (PID) is the power-up default. An update rate of about 8 kHz is possible with this equation. *Figures 4F-2* and *4F-3* show selections one and two as block diagrams.

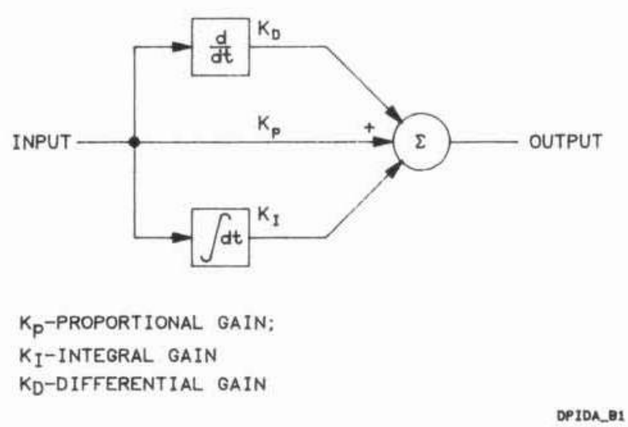

Figure 4F-2. PID Algorithm

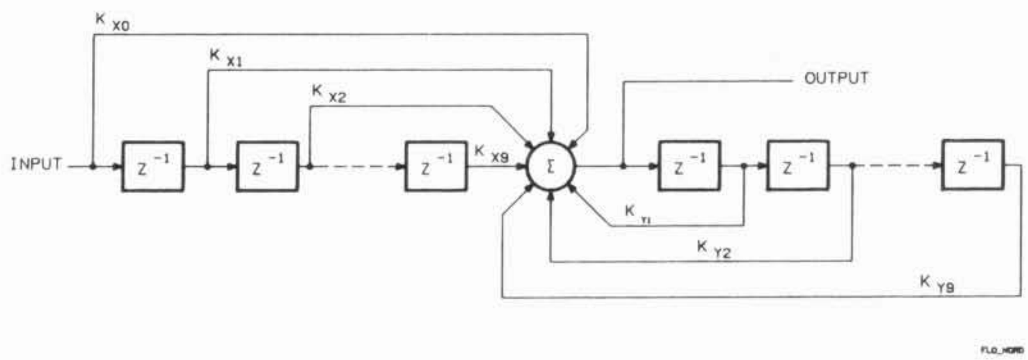

Figure 4F-3. 9th Order IIR Filter

Motor Drive Outputs

Three Motor Drive outputs are available; 16-bit digital, pulse-width modulated, and ±10 V analog. The three low order bits of the parameter for *OUTn enable these outputs. Setting any bit to 1 turns on the selected output option as follows:

Bit	n	Output
0	1	±10 V from a 16-bit DAC
1	2	Pulse-width modulated output
2	4	16-bit digital output

Note: The 16-bit binary interface output port is used if bit 2 is set, preventing any output over the binary interface.

The Servo-Axis board powers up with the ±10V output enabled.

INITIALIZING MOTION PARAMETERS

The following parameters control system response and must be set before the Servo-Axis board will generate any Motor Drive signals.

Difference Equation Coefficients

The selected difference equation needs coefficients for each of its terms. For the PID equation, three are needed. For the IIR equation, up to 19 can be used. These coefficients are programmed by setting a pointer (*PTR) to a value, sending the sequence of values to the *DTA mnemonic, and then setting the pointer to a base value to load all the coefficients at once. The order for the PID equation is K_P, K_I, K_D, and for the IIR filter is K_{X0}, K_{Y1}, K_{X1}, K_{Y2}, ...K_{Y9}, K_{X9} .

For example, the following sequence sets the terms for the PID equation to 0.753 for the proportional gain, 0.02 for the integral gain, and 1.233 for the differential gain, and then loads the terms for use by setting *PTR to 0.

"XPTR1;XDTA0.753;0.02;1.233;XPTR0"

Generally, the values for these terms will be determined empirically (see Tuning the Servo System). More detailed information on the pointer and additional banks of coefficients can be found under the *PTR mnemonic.

Profile Parameters

One to three parameters are used to control the internal generation of reference positions.

*VELn	Sets the maximum velocity (in current I/O units) for position profiling, or the desired velocity for velocity profiling.
*ACCn	Sets the maximum acceleration (in g's) for both position and velocity profiling.
*DACn	Sets the delta acceleration rate in g's per second.

These parameters should be set to correspond to (or be less than) the capabilities of the motor-stage system.

Feedforward Coefficients

Feedforward terms offer two benefits: First, proper feedforward helps minimize position tracking error during a move. Second, feedforward can help optimize settling time characteristics. Three terms are available. Their coefficients are set with the following mnemonics:

*FFVn	Sets the velocity feedforward coefficient.
*FFAn	Sets the acceleration feedforward coefficient.
*FFDn	Sets the delta acceleration feedforward coefficient.

See Tuning The Servo System for additional information on setting these parameters.

SERVO SYSTEM OPERATION

Drive Activation

The Servo-Axis board powers up with the Drive Enable Out- line off and the Motor Drive signals zeroed. Either of the following two mnemonics can turn on these outputs; however, the latter one is the recommended method. Turning off the drive is done with the first mnemonic.

*DREn	Turns on (n = 1), or off (n = 0) the Drive Enable Out- and Motor Drive signals.
*ACQ	Sets the destination to the current position and then turns on the Drive Enable Out- and Motor Drive signals.

Position Control

Sending a value to the *DES mnemonic causes the Servo-Axis board to generate the Motor Drive signals according to the above parameters, and move the stage to the desired position.

Alternately, if the buffered position profile has been selected, the *BGO mnemonic causes the reference position to be updated by the buffer values, and the Motor Drive signals cause the stage to follow the profile in the buffer (see Buffered Position Profiling).

For velocity profiling, sending *VEL causes the Servo-Axis board to generate Motor Drive signals that move the stage at the requested velocity.

General programming

The following mnemonics are also useful:

*MKR	Sets the marker value used for *PRE and the Threshold and Window outputs.
*MKW	Sets the window width used for the Window output.
*NAM	Returns the Servo-Axis board's name "SRVO".
*NUL	Sets the window width used for the Null Status Bit.
*POS	Returns the last position sampled (see Sample Trigger Source).
*PRE	Presets the position counter to the *MKR value.
*PTR102;*DTA	Returns the current trip state.
*PTR103;*DTA	Returns the current Motor Drive value.
*PTR104;*DTA	Returns the current Reference Position.
*PTR107;*DTA	Returns the last bad value written to the Servo-Axis board.
*PTR109	Acts like a pulse on the external sample input.
*PTR113;*DTA	Returns the Servo-Axis board's Miscellaneous Status Word.
*REV	Returns the Servo-Axis board's firmware revision date.
*STA	Returns the Servo-Axis board's status byte.
*TCN	Sets the total compensation number.
*TRC	Sets up the TRACE operation (See Programming Trace Functions).
*TST	Sets the measurement frequency to a fixed value for testing.
*UCN	Causes the Servo-Axis board to update its *TCN value using the mnemonics specified through *PTR106;*DTA.
*ZRO	Zeros the position counter and the Destination Register and sets TCN_0 = *TCN (used for deadpath correction).

Detailed information about each of the above mnemonics is in the Servo-Axis Board Mnemonics Reference (page 4F-29), which lists all the mnemonics alphabetically.

TUNING THE SERVO SYSTEM

Many methods exist for optimizing servo operation. The procedure outlined below produces sufficient performance for many systems, however, some systems will require more thorough and/or analytical techniques. Application note 325-10, "Sub-micron Positioning with the HP5527A Laser Position Transducer System", discusses some examples, and many text and reference books discuss closed loop systems in great detail.

A basic, empirical tuning procedure is:

a) Set *SVC to 3, External position profile.

b) Have stage take small steps and collect Actual Position trace data. Plot the trace data to see stage's step response.

c) While repeating step b, increase K_D until stage begins to be unstable, set K_D to 1/3 this value.

d) While repeating step b, increase K_P until the desired performance is reached.

e) Set *SVC to 0, Internal position profile. Set *DAC, *ACC and *VEL to 100, 0.2, and 50 respectively. (These values work well with low friction stages, other values may be more appropriate for your actual application.)

f) Have the stage take 3 mm steps and collect Actual Position and Reference trace data. Compute following error by subtracting Reference from Actual Position. Plot the result.

g) While repeating step f, adjust *FFV until the center of the plot is on zero (see *Figure 4F-4a* and *b*).

h) While repeating step f, adjust *FFA until the ends of the plot are on zero (see *Figure 4F-4b* and *c*).

i) While repeating step f, adjust *FFD until the transition areas are as close to zero as possible (see *Figure 4F-4c* and *d*).

j) Adjust K_I to reduce any positioning error.

This completes the basic tuning procedure. Additional adjustments may be necessary to further improve performance.

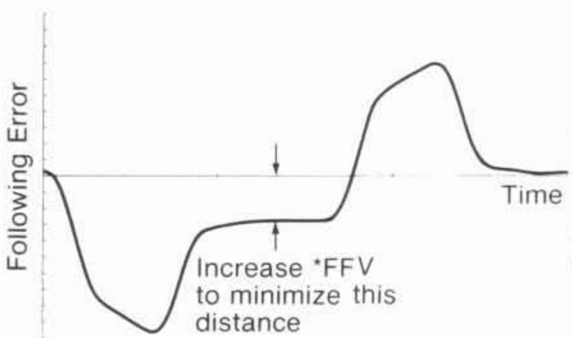

Figure 4F-4a. Adjusting Velocity Feedforward

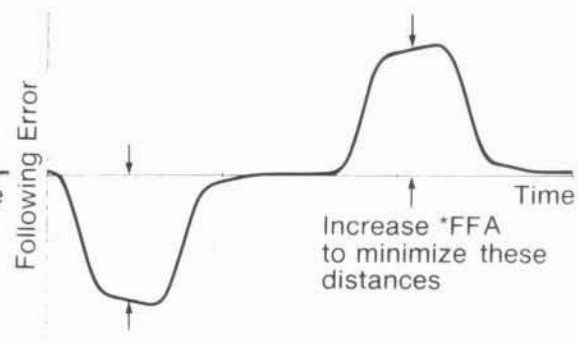

Figure 4F-4b. Adjusting Acceleration Feedforward

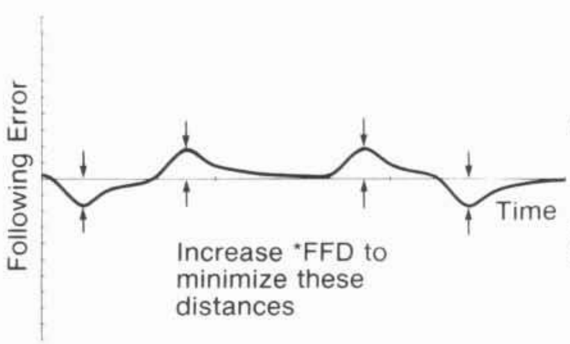

Figure 4F-4c. Adjusting Delta Acceleration Feedforward

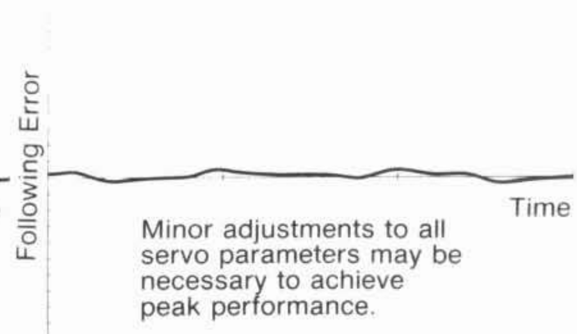

Figure 4F-4d. Adjusting Final Tracking Accuracy

Figure 4F-4. Adjusting Feedforward Terms

BINARY INTERFACE OPERATION

The 16-bit binary interface allows high-speed communication between the HP 5507A and a host computer. Commands are sent to the Servo-Axis board, which then takes some action based on the command number. The command number may

- direct this board to execute some command,
- direct another Servo-Axis board to execute some command,
- request data from this board,
- request data from another Servo-Axis board,
- send data to this board,
- send data to another Servo-Axis board, or
- direct this board to compile, execute, read, or write a compiled mnemonic.

The following paragraphs detail these operations. Section III provides the handshaking and timing information required to transfer each 16-bit value to or from the binary interface. Command numbers 0 through 511 correspond to functions on the servo-axis board that receives them, regardless of its backplane address. Command numbers 512 through 4607 correspond to the same functions on servo-axis boards at specific backplane addresses (see page 4F-67).

Command Execution

When the command number is for command execution, no further transfers are required and the Servo-Axis board is ready for the next command number. For example, sending 201, 205, and 210 in sequence to the binary interface will cause the Servo-Axis board to set the units to English, set external sample mode, and zero the position counter.

Data Requests

When the command number indicates a data request, the appropriate data is fetched and put in an output buffer. The data is then read by the host computer by reading the number of data words corresponding to the data type (additional reads will re-execute the same data request[‡]). Any further writes to the binary interface will be interpreted as commands, which may overwrite any unread data in the data buffer. For example, sending 3 to the binary interface loads the output buffer with the software revision date code which can be read by reading one word from the binary interface. If a 14 is written to the interface before the read takes place, then the date code in the output buffer is overwritten with position data in floating point format (see Data Formats — Four-Word Floating Point). The position data can be read by reading four words from the binary interface.

Data Writes

When the command number is for a compiled mnemonic or sending data, the Servo-Axis board will interpret the next few writes (number varies with data type) as the compiled mnemonic number and/or data to be written to the specified variable. The correct number of writes must take place before the Servo-Axis board is ready to receive the next command number. The sequence 48, 1, 22, 16367, -569, 11866, -6949, 4803, 3 contains only three command numbers, the rest are data values. The 48 tells the Servo-Axis board to use the next number (1) as the optics type. The 22 tells the board that the next four words are the TCN value in floating point format (0.999728766). Finally, the 4803 causes the board to execute the compiled mnemonic specified in the next data word, in this case 3 (see Compiled Mnemonics on page 4F-16).

‡ Firmware Revision after 2820, earlier revisions return 0's.

Data Formats

Each binary interface command has specific data formats associated with it. A total of four different formats are used. The following paragraphs describe each in detail.

One-Word Integers

One-word integers use a 16-bit two's-complement data format to represent the integer numbers from -32768 to 32767. This format is used for the commands themselves (command numbers), and for most data items.

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	-32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

For HP 9000 Series 200/300 computers using an HP 98622 GPIO Interface, the following statements illustrate how to read and write data in this format.

```
INTEGER Xacq,Xdir_wr,Xdre_rd,Xdre               ! These MUST be integers
READ Xacq,Xdir_wr,Xdre_rd                       ! Assign command numbers
DATA 3272,  3112 ,  3115
ASSIGN @Gpio to 12;WORD,FORMAT OFF,EOL OFF      ! Configure I/O channel
OUTPUT @Gpio;Xacq,Xdir_wr,1,Xdre_rd             ! Send command and data
ENTER @Gpio;Xdre                                ! Read One-Word Integer
```

Two-Word Integers

The two-word integer format uses two consecutive 16-bit values to transfer a 32-bit two's-complement integer. The most significant 16 bits are sent first. This format is used for miscellaneous 32-bit variables and for commands that deal only with RAW count data.

FIRST WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	-2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}

SECOND WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

For HP 9000 Series 200/300 computers using an HP 98622 GPIO Interface, the following statements illustrate how to read data in this format. (Data can be written in this format too, however doing so in BASIC is very cumbersome. HP recommends using Pascal, C, or assembly language when writing data in the two-word integer format).

```
REAL Xpos_raw
INTEGER Xpos_rd_raw,Data_0,Data_1               ! These MUST be integers
READ Xpos_rd_raw                                ! Assign command number
DATA  3087
ASSIGN @Gpio to 12;WORD,FORMAT OFF,EOL OFF      ! Configure I/Ochannel
OUTPUT @Gpio;Xpos_rd_raw                        ! Send Request for Data
ENTER @Gpio;Data_0,Data_1                       ! Read Two Integers
X_pos_raw=Data_0*2^16+Data_1+2^16*(Data_1<0)    ! Combine values
```

Note: 3087 = 3072 + 15, 3072 is base number for Servo-Axis board at address X, 15 is read raw position command (see page 4F-67).

Four-Word Floating Point

Four-word floating point data uses four consecutive 16-bit values to transfer a 64-bit IEEE-P754 Double Precision floating point value. The first word contains the most significant bits. This format is used for all variables requiring the precision or range of floating point numbers.

FIRST WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
meaning:	Sign	e10	e9	e8	e7	e6	e5	e4	e3	e2	e1	e0	f51	f50	f49	f48

SECOND WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
meaning:	f47	f46	f45	f44	f43	f42	f41	f40	f39	f38	f37	f36	f35	f34	f33	f32

THIRD WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
meaning:	f31	f30	f29	f28	f27	f26	f25	f24	f23	f22	f21	f20	f19	f18	f17	f16

FOURTH WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
meaning:	f15	f14	f13	f12	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1	f0

The value of the number represented (v) is given by the following formulas:

if 0<e<2047, then v=(-1)^s * 2^(e-1023) * 1.f

if e=0 then, v=0

if e=2047, then v is undefined

where:

e is the number represented by bits e10 through e0 (e10 is MSB - i.e., 2^10)

f is the fraction represented by bits f51 through f0 (f51 is MSB - i.e., 2^-1)

For HP 9000 Series 200/300 computers using an HP 98622 GPIO Interface, the following statements illustrate how to read and write data in this format.

```
REAL Xdes,Xpos
INTEGER Xdes_wr,Xpos_rd                        ! These MUST be Integer
READ Xdes_wr,Xpos_rd,Xdes                      ! Assign command numbers
DATA 3082    ,3086    ,75.236                  !   & destination
ASSIGN @Gpio to 12;WORD,FORMAT OFF,EOL OFF     ! Configure I/O channel
OUTPUT @Gpio;Xdes_wr,Xdes,Xpos_rd              ! Output commands & data
ENTER @Gpio;Xpos                               ! Read real number
```

ASCII

The ASCII data format consists of a sequence of 16-bit transfers with the code for the ASCII character in the least significant seven bits. The last two characters will always be a carriage return and a line feed. This format is only used for reading error messages or compiled mnemonic status from the HP 5507A.

FIRST WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	0	0	0	0	0	0	0	0	0	a6	a5	a4	a3	a2	a1	a0

NEXT WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	0	0	0	0	0	0	0	0	0	a6	a5	a4	a3	a2	a1	a0

.

.

.

.

.

SECOND LAST WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

LAST WORD

bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

where a6 through a0 are the ASCII code for the particular character.

For HP 9000 Series 200/300 computers using an HP 98622 GPIO Interface, the following statements illustrate how to read data in ASCII format (all ASCII data is read only).

```
DIM Xerr$[80]                                   ! Dimension string variable
INTEGER Xerr                                    ! These MUST be integer
READ Xerr                                       ! Assign command number
DATA 4824
ASSIGN @Gpio to 12;WORD,FORMAT OFF,EOL OFF      ! Configure I/O channel
ASSIGN @Asc to 12                               ! Configure ASCII channel
OUTPUT @Gpio;Xerr                               ! Output request for data
ENTER @Asc;Xerr$                                ! Read in string
```

Compiled Mnemonics

Most of the command numbers for the binary interface are dedicated to Servo-Axis boards. Access to other function boards in the HP 5507A is through the five command numbers (4800 through 4804) associated with compiled mnemonics. One of these numbers is used to tell the interface what mnemonic to compile. Three are used to read, write or execute previously compiled mnemonics. The remaining command number is used to read the status of a specific compiled mnemonic number. The following paragraphs describe each operation in detail.

Defining a Compiled Mnemonic

Command number 4800 is used to tell the interface what mnemonic to compile. The mnemonic is specified by two one-word integers which are formed by taking the ASCII characters for the mnemonic and packing two characters into each word. For example, sending 4800, 22081, 21590 causes the interface to compile the mnemonic "VATV" (ASCII codes for V, A, T, and V are 86, 65, 84 and 86 respectively, 86*256 + 65 = 22081 and 84*256 + 86 = 21590).

The interface responds by verifying the existence of the mnemonic, assigning the next compiled mnemonic number to it, and putting this number and the mnemonic type in the output buffer. These two one-word integers should be read by the host computer to get the assigned compiled mnemonic number and to verify proper compilation (0 will be returned for both when compilation fails). The following chart indicates how to interpret the mnemonic type returned above.

Bit:	15 through 7	6	5 & 4	3	2	1	0
Value:	--	64	-	8	4	2	1
Meaning:	Not Used	Read Only	Not Used	ASCII (read only)	Four Word F.P.	One Word Int.	Command

In the above example the mnemonic type would be a 4.

The compiled mnemonic number depends on how many mnemonics have already been compiled. The interface assigns numbers sequentially starting with 1. Any attempt to re-compile the same mnemonic will return the same compiled mnemonic number (a hard reset will clear all compiled mnemonics).

Any four character mnemonic that would be recognized by a function board may be compiled. Mnemonics that are specific to the HP-IB board can not be compiled.

Executing a Compiled Mnemonic

Command numbers 4801, 4802 and 4803 are used to read from, write to, and execute a compiled mnemonic. The compiled mnemonic number must follow the command number, and any data must follow that (the mnemonic type returned above specifies the data format). Reads from compiled mnemonics put the requested data in the output buffer. If the compiled mnemonic number in the above example was 5, then sending 4802, 5, 16437, 15400, -2622, -28836 sends the value 21.235 to the VATV mnemonic. Sending 4801, 5 will put the current value of the VATV mnemonic in the output buffer. This value is then read by reading four words (see Data Formats — Four-Word Floating Point).

Verifying a Compiled Mnemonic

Command number 4804 is used to verify what mnemonic and data type are associated with a particular compiled mnemonic number. Again the compiled mnemonic number must follow the command number. The interface responds by putting the mnemonic type and the mnemonic in the output buffer. The mnemonic type is a one-word integer (see chart under Defining Compiled Mnemonics) and the mnemonic is a four character ASCII string followed by Carriage Return and Line Feed (see Data Formats—ASCII). If the compiled mnemonic number has not been assigned, then the mnemonic type will be zero and the mnemonic will be four dashes.

Error Messages

The HP-IB interface has one mnemonic, ERRM, for requesting the most recent system error message. The binary interface allows reading error messages from all function boards, should errors occur on multiple boards. The command numbers 4811 through 4826 read error messages from function boards at address K through Z respectively. Command number 4810 reads the error message of this Servo-Axis board. All error messages are output in ASCII data format.

NOTE

The status of each function board (*STA) should be read first to verify that an error exists, as some firmware revisions do not clear the error message data after the error has been cleared.

BUFFER UTILIZATION

The Servo-Axis board has 24000 long (32-bit wide) words of buffer memory that are shared between trace and buffered reference. At power up, and after a *PTR105 command, the board assigns all of this memory as trace data. Each write to the reference buffer causes the first long word of trace memory to be reassigned to store reference buffer data. Each time a trace is performed, any memory not used by the reference buffer is filled with trace data. This operation maximizes the memory available for trace, but overwrites some of trace memory with each write to buffered reference (ie. Read any trace data before changing the reference buffer data).

Each reference buffer value uses one long word of memory, allowing 24000 positions to be stored. If the interpolation count is set to zero, and the sample period set to 1000 (8 kHz), then a three second journey can be stored. Increasing either the interpolation count or the sample period lengthens this journey time. For example, using a 2 kHz sample rate and an interpolation count of 4, a 60 second journey can be stored.

Each trace data item uses one or less long words of memory (motor drive and difference equation output each use only 16 bits, or 1/2 long words). Tracing all parameters (reference position, actual position, equation output, and motor drive) requires three long words per sample period, allowing 8000 sample periods, or 1 second of data at the fastest sample rate. Reducing the sample rate or number of parameters traced will increase the maximum trace time. For example, tracing only Actual Position (one long word) with a 2 kHz sample rate results in 12 seconds of trace data.

When both buffered reference and trace functions are utilized, the times for each should be adjusted accordingly. For example, filling the buffer with 2000 data points, setting an interpolation value of 9, and using a 2 kHz sample rate produces a 10 second stored trip, and allows Actual Position to be traced for 11 seconds.

PROGRAMMING TRACE FUNCTIONS

Up to four different system signals can be recorded in real-time by the Servo-Axis board. There are 24,000 long (32-bit) words of buffer space available for data storage. Part, or all, can be reserved for trace data (the rest is available for reference generation – see Buffer Utilization).

The *TRCn mnemonic selects which system signals to store in the buffer, and when to start storing them. Then, once per Servo-Clock period, the selected signals are recorded for later retrieval. Recording stops when the end of buffer memory is reached. The following table shows the correspondence between n (in *TRCn) and the system signals that can be traced.

Bit	n	Purpose	Data Length
0	1	Reference Data	(32-bit long-word)
1	2	Actual Position Data	(32-bit long-word)
2	4	Difference Equation Output	(16-bit word)
3	8	Motor Drive	(16-bit word)
4	16	Not currently used	
5	32	Not currently used	
6	64	Start trace on *DES/*VEL/*BGO command	
7	128	Start trace on this *TRC command	

Trace data is stored in the same order as above. For example, specifying a trace of all four items with XTRC 79 causes the data to be stored in the following order: Reference Data, Actual Position Data, Difference Equation Output, Motor Drive, Reference Data, Actual Position Data, etc. Furthermore, the trace will not start until a *DESn command is sent.

Additional signals can be calculated by combining two or more of the traced parameters. For example, subtracting Reference from Actual Position yields following error, and subtracting Difference Equation Output from Motor Drive yields Feedforward. Graphing these parameters versus time greatly reduces the time spent optimizing servo-system performance.

Reading Trace Data

The data recorded during a trace can be read over the 16-bit binary interface, or through HP-IB. HP-IB will be very slow, however, with a full trace taking about 40 seconds to transfer, compared with about 0.5 seconds over the binary interface (HP 9000 Series 300 computer). The following two paragraphs illustrate using each interface.

HP-IB

Reading trace data over HP-IB is fairly simple, but it takes time. First, the trace buffer pointer is reset to the beginning of trace memory with a *PTR100 command. This puts the first data value in the *DTA mnemonic, which is read with a *DTA? query. Next, the pointer is repeatedly incremented with *PTR101 commands, and the data read with *DTA? queries. Each data item read corresponds to a 16-bit word in trace memory, most significant 16 bits first for 32-bit data items (Reference Position and Actual Position). Attempts to read beyond the end of trace memory return zero. The following BASIC code segment illustrates this sequence:

```
100  OPTION BASE 1                              ! Set 1 as first element for arrays
110  DIM Buffer(1000),Position(500)             ! Dimension Data Arrays
120  Max_index=300                              ! Specify # of data points to read
130      REM
131      REM  Additional code goes here
132      REM
200  OUTPUT 703;"XTRC66;XDES10"                 ! Trace Actual Position
210      REM
211      REM  Additional code can go here
212      REM
300  OUTPUT 703;"XPTR100;XDTA?"                 ! Point to first Trace Data Item
310  FOR I=1 TO 2*Max_index                     ! Loop_1: Read portion of Trace Data
320    ENTER 703;Buffer(I)                      !   Read Data Item
330    OUTPUT 703;"XPTR101;XDTA?"               !   Point to next item
340  NEXT I                                     ! End of Loop_1
350  FOR I=2 TO 2*Max_index STEP 2              ! Loop_2: Combine 2 Data Items
360    Position(I/2)=Buffer(I-1)*2^16+Buffer(I)+2^16*(Buffer(I)<0)
370  NEXT I                                     ! End of Loop_2
400      REM
410      REM  Plotting or data analysis routine goes here
420      REM
500  END
```

Note: The "703" in the above code segment points to the HP 5507A at HP-IB address 03. The "X" in the mnemonics point to a Servo-Axis board at backplane address "X"

Binary Interface

Reading trace data over the binary interface is simpler than reading it over HP-IB. Again, the trace buffer pointer must be reset to the beginning of trace memory, but this time a binary 1 is sent to the Servo-Axis board over the interface (see Binary Interface Operation). Now, all the trace data can be read by successive reads of the Servo-Axis board's 16 data output lines. Attempts to read beyond the end of trace memory return zero. The following BASIC code segment illustrates this (for HP 9000 series 200/300 computers, the first FOR-NEXT loop can be replaced with a TRANSFER statement):

```
100  OPTION BASE 1                              ! Set 1 as first element for arrays
110  INTEGER Buffer(1000)                       ! Dimension INTEGER Data Array
111  REAL Position(500),Dest                    ! Dimension REAL Data Array
120  Max_index=300                              ! Specify # of data points to read
130      REM
131      REM  Additional code goes here
132      REM
200  OUTPUT 12 USING "W,#";56,66,10,Dest        ! Trace Actual Position
210      REM
211      REM  Additional code can go here
212      REM
300  OUTPUT 12 USING "W,#";1                    ! Point to first Trace Data Item
310  FOR I=1 TO 2*Max_index                     ! Loop_1: Read portion of Trace Data
320    ENTER 12 USING "W,#";Buffer(I)           !   Read Data Item
340  NEXT I                                     ! End of Loop_1
350  FOR I=2 TO 2*Max_index STEP 2              ! Loop_2: Combine 2 Trace Data Items
360    Position(I/2)=Buffer(I-1)*2^16+Buffer(I)+2^16*(Buffer(I)<0)
370  NEXT I                                     ! End of Loop_2
400      REM
410      REM  Plotting or data analysis routine goes here
420      REM
500  END
```

Note: The "12" in the above code segment points to the computer's HP 98622 GPIO interface. The "W,#" indicates that a 16-bit binary Word is the data format, and that no termination sequence should be sent or received.

PROGRAMMING REFERENCE BUFFER OPERATIONS

Three operating modes use the reference buffer as a data source. One of them (*SVC 8 through *SVC 10) treats the data as offset destination values. This mode allows custom motion profiles which can not be generated from the velocity, acceleration and delta acceleration terms (see Reference Source - Buffered Position Profiling). The other two modes (*SVC 11 and *SVC 12) treat the data as numeric input to be injected at a specific point in the system (see *Figure 4F-5*, Servo System Injection Points). These latter modes are useful when tuning the servo system and determining the open and closed-loop transfer functions.

To use these three modes, *SVC is set as desired, and *CLK and *MCM are set accordingly (same master/slave/stand alone state). The I/O units are specified and then the buffer is filled with the desired journey/test signal (see Writing Buffer Data below), and a *BGO command is given.

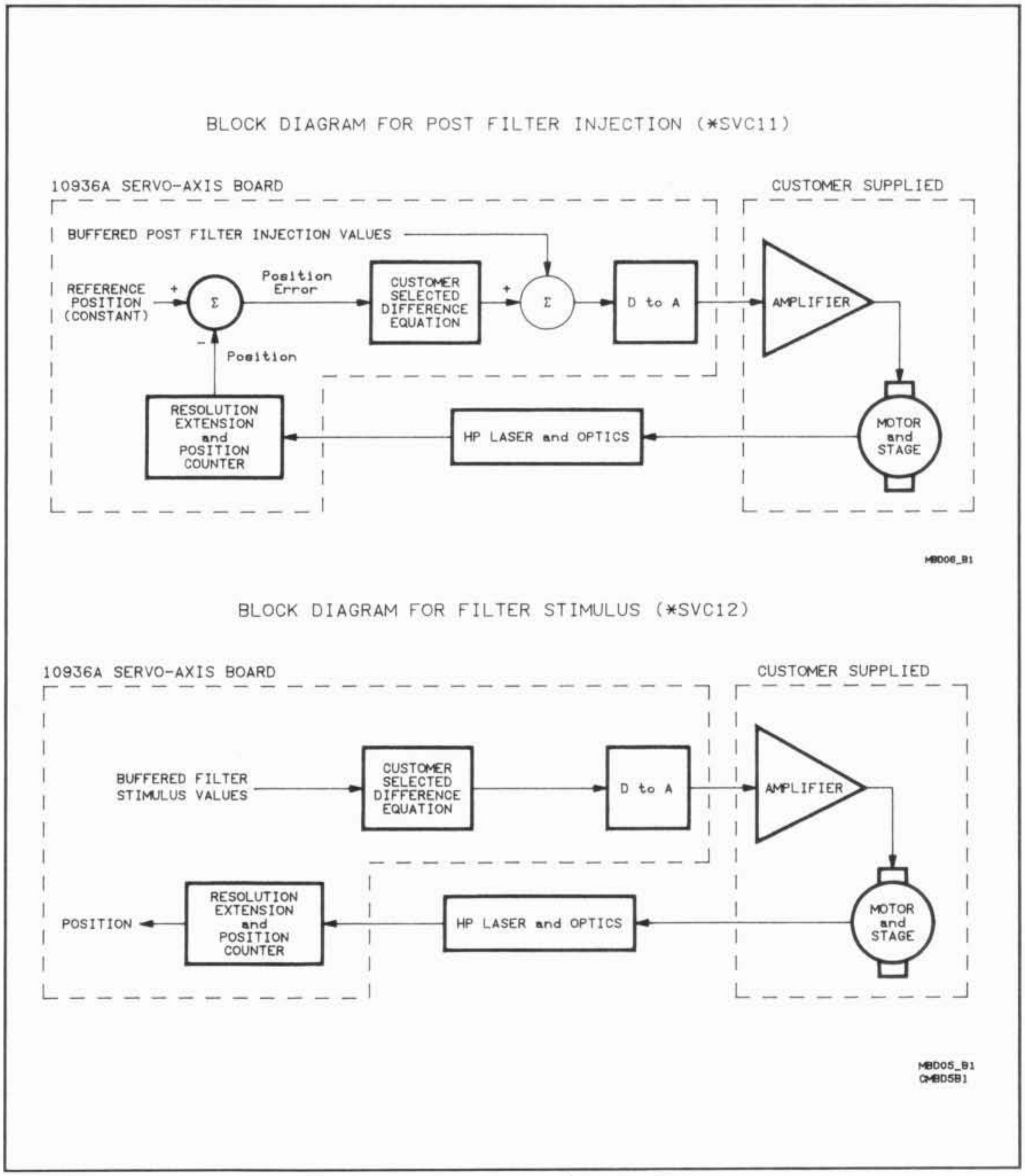

Figure 4F-5. Servo System Injection Points

The *BGO command allows several options for the buffer's use. The "Forward" and "Reverse" options described on page 4F-30 refer to the order in which the buffer values will be used (first buffer value used first for Forward, or used last for Reverse). The "Positive" and "Negative" options refer to the sign conventions for the buffer (positive buffer values correspond to Positive or Negative stage motion). The "Follow Master" option causes the board to use the same *BGO option as its master.

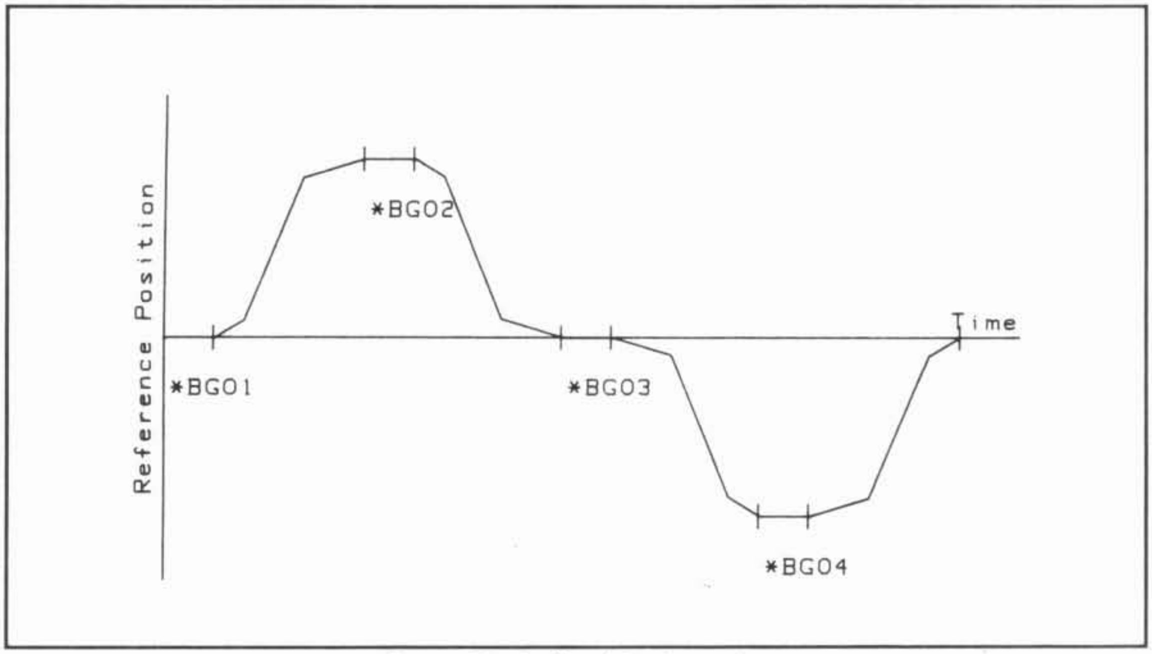

Figure 4F-6. Buffered Go Operation

Writing Buffer Data

When loading the buffer, the Servo-Axis board converts each value sent (unless *RAW or *LAM is programmed) from current units (*ENG or *MET) to compensated lambda units. Then, when using the buffer for Buffered Position Profiling (*SVC 8 -*SVC 10), the board uses the current *TCN and *DPD values to convert to raw counts before adjusting the reference position. Note that the buffer values are offset destinations, meaning all are relative to the stage position when a *BGO command is received (for Reverse operation, the last value in the buffer is subtracted from all values during the move).

To increase system flexibility and performance, the Buffered Position Profiling algorithms (*SVC 8-10) interpolate between buffer values when *PTR115;*DTA is non-zero. Setting the Interpolation Count to N causes the board to insert N servo sample periods between each buffer value, and update the reference position proportionatly on each sample period. For *SVC 11 and *SVC 12‡ the *PTR115;*DTA value is a loop count instead of an interpolation count. Setting the loop count to N causes the board to automatically repeat the *BGO command N additional times (see page 4F-48).

For Buffered Post Filter Injection (*SVC 11), the board clips the 32-bit compensated lambda value to 16 bits (maximum or minimum integer used if value outside range of 16-bit integers) and injects it after the filter. For Buffered Filter Stimulus (*SVC 12), the board sends the full 32-bit compensated lambda value to the difference equation in place of position error. To accurately know the injected signal, HP recommends programming *RAW or *LAM units when using Buffered Injection/Stimulus.

‡ Revisions after 2820 only; earlier revisions ignore this value in these modes.

The following paragraphs illustrate loading the buffer using either HP-IB or the binary interface.

HP-IB

Writing data to the reference buffer over HP-IB consists of

- setting the desired units (*ENG, *MET, *LAM, *RAW),
- clearing and pointing to the buffer with a *PTR105 command,
- sending data values with repeated *DTA Data_value commands.

The following program segment for HP 9000 Series 200/300 computers shows this sequence. Note that *DTA does not have to be repeated each time, speeding data transfer.

```
100  OPTION BASE 1                          ! Set 1 as first element for arrays
110  DIM Position(1000)                     ! Dimension Data Array
120  Max_index=1000                         ! Specify number of data points
130      REM
140      REM  Additional code goes here to put desired profile in Position Array
150      REM  and setup Servo-Axis board as desired.
160      REM
200  OUTPUT 703;"XSVC8;XCLK0;XMCM0"         ! Select Buffered Position Profiling
210  OUTPUT 703;"XPTR115;XDTA19"            ! Set 20 Sample Periods/Buffer Value
220      REM
230      REM  Additional code can go here
240      REM
300  OUTPUT 703;"XENG;XPTR105"              ! Set Units & Clear Reference Buffer
310  OUTPUT 703;"XDTA";                     ! Setup to write repeatedly to XDTA
320  FOR I=1 TO Max_index                   ! Loop: Write profile to buffer
330    OUTPUT 7 USING "K";DROUND(Position(I),9)!   Limit to 9 digits & no spaces
340  NEXT I                                 ! End of Loop
400      REM
410      REM  Additional code can go here
420      REM
500  OUTPUT 703;"XBGO1"                     ! Initiate Buffered Move
600      REM
610      REM  Additional code can go here
620      REM
700  END
```

In this example, the "X" indicates that a Servo-Axis board is set to this backplane address and "703" indicates that the HP 5507A is set to HP-IB bus address 3.

Binary Interface

Writing data to the reference buffer over the binary interface is almost identical to using HP-IB. The following steps are required:

- set desired units (201 thru 204 - ENG, MET, LAM, RAW),
- clear the buffer with a 216 command,
- send data values with repeated 65, Data_value commands.

The following program segment for HP 9000 Series 200/300 computers shows this sequence. For the binary interface, the 65 command must be repeated each time to tell the board what the next four words mean (four-word floating point value).

```
100  OPTION BASE 1                          ! Set 1 as first element for arrays
110  REAL Position(1000)                    ! Dimension Data Array
120  Max_index=1000                         ! Specify number of data points
130      REM
140      REM  Additional code goes here to put desired profile in Position Array
150      REM  and setup Servo-Axis board as desired.
160      REM
200  OUTPUT 12 USING "W,#";54,8,36,0,46,0   ! Select Buffered Position Profiling
210  OUTPUT 12 USING "W,#";76,19            ! Set 20 Sample Periods/Buffer Value
220      REM
230      REM  Additional code can go here
240      REM
300  OUTPUT 12 USING "W,#";201,216          ! Set Units & Clear Reference Buffer
320  FOR I=1 TO Max_index                   ! Loop: Write profile to buffer
330    OUTPUT 12 USING "W,#";65,Position(I) !   Output Position in 4-Word F.P.
340  NEXT I                                 ! End of Loop
400      REM
410      REM  Additional code can go here
420      REM
500  OUTPUT 12 USING "W,#";212              ! Initiate Buffered Move
600      REM
610      REM  Additional code can go here
620      REM
700  END
```

Note: The "12" in the above code segment points to the computer's HP 98622 GPIO interface. The "W,#" indicates that a 16-bit binary Word is the data format, and that no termination sequence should be sent or received.

CUSTOM DIFFERENCE EQUATIONS

Although two difference equation algorithms are built into the HP 10936A's firmware, some applications will require more sophisticated/specialized servo routines. The downloaded difference equation allows this flexibility.

The algorithm must take a 32-bit position error input and produce a 16-bit output value based on the error and programmed coefficients (some additional parameters are also available - see Parameter Locations below). The routine must also be fast. Slow routines will reduce the maximum servo clock rate, and will cause the board to lock-up should the sample period (*SPD) be set too short (monitoring test point 5, located near board center by U24, indicates the relative time spent on servo sample period tasks verses the background routines). As a bench-mark value, the PID algorithm executes in less than 40 μs. Due to the timing requirements, HP recommends writing the routine in assembly code. The following paragraphs outline the conventions used and the parameter locations.

Subroutine Conventions

The following are some Dos and Don'ts to follow when writing a custom Difference Equation routine for the HP 10936A Servo-Axis board.

a) The routine must be implemented as a machine language (68020) subroutine, i.e.,. its last instruction is an RTS.

b) The routine may use A0 and A1 without saving their values. If additional address registers are needed, then they must be saved before use, and restored before returning.

c) The routine may use any of the data registers (D0 through D7). Note that on entry, D0 contains the current position and D1 contains the reference position minus the current position.

d) The routine should only use the coefficients in block A or B as indicated by the coeff_in_use variable. This will guarantee that none of the coefficients are being changed.

e) The routine must generate a 16-bit signed output value. This word is sign extended to 32-bits and left in D0 before returning. It must also be stored in Eq_out for proper operation of trace functions.

f) The routine must fit in 4096 bytes. Attempting to load a longer routine will generate an error.

g) The code must be relocateable. Different firmware revisions may place the code in different locations.

h) The routine should be well behaved (i.e., not jump beyond itself, access other variables, etc.). Should it inadvertently do so, the system will probably lock-up and/or generate a firmware execution error #799, Fatal Exception: SSSS PPPPPPPP FVVV(see page 4F-59).

```
Line  Address
                                           chip          68020
                                           opt           case
      33800000                           k_block       equ $33800000
      33800190                           coeff_in_use  equ $33800190
      33800194                           coeff_a       equ $33800194
      338001E0                           coeff_b       equ $338001e0
      3380022C                           history       equ $3380022c
      338002CC                           hist_ptr      equ $338002cc
      338002D0                           misc16        equ $338002d0
      338002F0                           misc32        equ $338002f0
      33800330                           drive         equ $33800330
      33800332                           eq_out        equ $33800332
      33800334                           order         equ $33800334
      33800336                           spd           equ $33800336
      33800338                           trip_state    equ $33800338
      33800339                           nulled        equ $33800339
                                         *
                                         *
                                         *  begin PID equation
                                         *
                                         equa_pid
      00000000 207C 3380 022C              move.l      #history,a0
      00000006 2279 3380 0190              move.l      coeff_in_use,a1
                                         *
      0000000C 2601                        move.l      d1,d3
      0000000E 4C19 3C02                   muls.l      (a1)+,d2:d3
                                         *
      00000012 2819                        move.l      (a1)+,d4
      00000014 6608                        bne.s       pid_do_int
      00000016 20FC 0000 0000              move.l      #0,(a0)+
      0000001C 6022                        bra.s       pid_no_int
                                         *
                                         pid_do_int
      0000001E 2001                        move.l      d1,d0
      00000020 D090                        add.l       (a0),d0
      00000022 6812                        bvc.s       int_acc_ok
                                         *
      00000024 4A81                        tst.l       d1
      00000026 6B08                        bmi.s       int_lim_neg
      00000028 203C 7FFF FFFF              move.l      #$7fffffff,d0
      0000002E 6006                        bra.s       int_acc_ok
                                         *
                                         int_lim_neg
      00000030 203C 8000 0000              move.l      #$80000000,d0
                                         *
                                         int_acc_ok
      00000036 20C0                        move.l      d0,(a0)+
      00000038 4C04 0C04                   muls.l      d4,d4:d0
      0000003C D680                        add.l       d0,d3
      0000003E D584                        addx.l      d4,d2
Line  Address
                                         *
                                         pid_no_int
      00000040 2001                        move.l      d1,d0
      00000042 9090                        sub.l       (a0),d0
      00000044 2081                        move.l      d1,(a0)
```

```
00000046 4C11 0C04              muls.l    (a1),d4:d0
0000004A D680                   add.l     d0,d3
0000004C D584                   addx.l    d4,d2
                        *
0000004E 4A82                   tst.l     d2
00000050 6B10                   bmi.s     check_output_neg
00000052 6600 0006              bne       output_lim_pos
                        *
00000056 4A83                   tst.l     d3
00000058 6A16                   bpl.s     output_ok
                        *
                        output_lim_pos
0000005A 263C 7FFF FFFF         move.l    #$7fffffff,d3
00000060 600E                   bra.s     output_ok
                        *
                        check_output_neg
00000062 5282                   addq.l    #1,d2
00000064 6604                   bne.s     output_lim_neg
00000066 4A83                   tst.l     d3
00000068 6B06                   bmi.s     output_ok
                        *
                        output_lim_neg
0000006A 263C 8000 0000         move.l    #$80000000,d3
                        *
                        output_ok
00000070 2003                   move.l    d3,d0
00000072 4840                   swap      d0
00000074 33C0 3380 0332         move.w    d0,eq_out
0000007A 48C0                   ext.l     d0
                        *
0000007C 4E75                   rts
                        *
                                end
```

Symbol Table

Label	Value
check_output_neg	9:00000062
coeff_a	33800194
coeff_b	338001E0
coeff_in_use	33800190
drive	33800330
eq_out	33800332
equa_pid	9:00000000
hist_ptr	338002CC
history	3380022C
int_acc_ok	9:00000036
int_lim_neg	9:00000030
k_block	33800000
misc16	338002D0
misc32	338002F0
nulled	33800339
order	33800334
output_lim_neg	9:0000006A
output_lim_pos	9:0000005A
output_ok	9:00000070
pid_do_int	9:0000001E
pid_no_int	9:00000040
spd	33800336
trip_state	33800338

Parameter Locations

The following table lists all the parameters, their location, data length and meaning.

Table 4F-2. Parameter Locations for Custom Difference Equations

Label	Address (HEX)	Data Length	Meaning
k_block	3380 0000	2 Words	Coefficient banks 1 - 5, 100 values
coeff_in_use	3380 0190	2 Words	Pointer to coeff_a or coeff_b
coeff_a	3380 0194	2 Words	Coefficient block A, 19 values
coeff_b	3380 01E0	2 Words	Coefficient block B, 19 values
history	3380 022C	2 Words	Data Area for maintaining values between sample periods, 40 values. Zeroed when drive is enabled
temp	3380 02CC	1 Word	Same as history but only 1 value (used by IIR as index to history)
misc16	3380 02D0	1 Word	16 miscellaneous 16-bit parameters
misc32	3380 02F0	2 Words	16 miscellaneous 32-bit parameters
drive	3380 0330	1 Word	Current Motor Drive Value
eq_out	3380 0332	1 Word	Put Output Word Here and in D0
order	3380 0334	1 Word	Indicates where first non-zero coefficient is. =INT(coef#/2)
spd	3380 0336	1 Word	Value set with *SPD mnemonic
trip_state	3380 0338	1 Byte	Value returned with *PTR102;*DTA?
nulled	3380 0339	1 byte	Null status 0 or 1 - see *PTR113;*DTA
current_pos	Reg D0	2 Words	Current position in Raw Counts
pos_err	Reg D1	2 Words	Position Error in Raw Counts (Reference Position - Actual Position)

Downloading 68020 Code

Downloading the actual machine code is straightforward once it has been assembled. First the equation space is cleared with a *PTR114 command. Then each word is sent to the *DTA mnemonic, making sure to treat each code word as a signed 16-bit integer. The following program segments show how this can be done using an HP 9000 series 200/300 computer.

HP-IB

The following program list is an example of downloading 68020 code via HP-IB.

```
100  OPTION BASE 1                            ! Set 1 as first element for arrays
110  INTEGER Code(2048)                       ! INTEGER array to hold assembly code
130      REM
140      REM  Additional code goes here to put desired assembly code into code
150      REM  array and setup Servo-Axis board as desired. Also Max_index
160      REM  MUST be specified to indicate how many code words to send.
170      REM
300  OUTPUT 703;"XPTR114"                     ! Clear Equation Space
310  OUTPUT 703;"XDTA";                       ! Setup for repeated writes
320  FOR I=1 TO Max_index                     ! Loop: Write code to servo-axis bd.
330    OUTPUT 7 USING "K";Code(I)             !   send one 16-bit word, no spaces
340  NEXT I                                   ! End of Loop
400      REM
410      REM  Additional code can go here
420      REM
700  END
```

Note: The "703" in the above code segment points to the HP 5507A at HP-IB address 03. The "X" in the mnemonics point to a Servo-Axis board at backplane address "X".

Binary Interface

The following program list is an example of downloading 68020 code via the Binary Interface.

```
100  OPTION BASE 1                            ! Set 1 as first element for arrays
110  INTEGER Code(2048)                       ! INTEGER array to hold assembly code
130      REM
140      REM  Additional code goes here to put desired assembly code into code
150      REM  array and setup Servo-Axis board as desired. Also Max_index
160      REM  MUST be specified to indicate how many code words to send.
170      REM
300  OUTPUT 12 USING "W,#";221                ! Clear Equation Space
320  FOR I=1 TO Max_index                     ! Loop: Write code to servo-axis bd.
330    OUTPUT 12 USING "W,#";75,Code(I)       !   send one 16-bit word
340  NEXT I                                   ! End of Loop
400      REM
410      REM  Additional code can go here
420      REM
700  END
```

Note: The "12" in the above code segment points to the computer's HP 98622 GPIO interface. The "W,#" indicates that a 16-bit binary Word is the data format, and that no termination sequence should be sent or received.

PUTTING EVERYTHING TOGETHER

The HP 10936A Servo-Axis Board's numerous options may make setup and operation appear difficult. The best way to move up the learning curve rapidly is to initially work with only one axis.

- Program the optics type, measurement input source, and limit switch polarity. If the optics are set wrong, then all measurements will be off by a factor of two or four. If the source is set wrong, there will always be a measurement error. If the polarity is wrong, there won't be any output when verifying feedback polarity (step after next).
- Determine the direction sense by moving the stage manually and reading *POS values. Set *DIR so the values returned are increasing as the optics move away from each other (if this is set incorrectly, then errors will be introduced by the deadpath correction equations and the limit switches will be at the wrong ends).
- Verify proper feedback polarity by monitoring the DAC OUT line while manually moving the stage (connect the amplifier input to a variable voltage source, set K_P to a small value, 0.1 is good, and send a *ACQ command to turn on the DAC OUT signal). If a positive signal into the amplifier moves the stage and causes a more positive DAC OUT output signal, then the feedback is backwards. Swapping the motor leads should fix it.
- Select the proper motor output (*OUT) for the application and connect this output signal to the power amplifier, closing the feedback loop.
- Leave the remaining parameters at default values.
- Roughly tune this axis by following the empirical tuning procedure outlined in Tuning The Servo System. See Programming Trace Functions for details about getting the data used to plot step response.
- Repeat the above for all axes.

Now, some of the more involved options can be explored.

- Experiment with other *SPD, *VEL, *ACC, and *DAC values and see how they affect performance (changing *SPD affects gain of K_I and K_D). Again use trace to plot system response.
- Program a buffered filter stimulus signal and trace actual position. The open-loop transfer function can be calculated from this data.
- Program a buffered post-filter injection signal and trace difference equation output. The closed-loop transfer function can be calculated from this data.
- Define a mnemonic for *UCN to use.
- Experiment with some compiled mnemonics and read information from a compensation or axis board through the binary interface.

Finally, with the information and experience gathered above, go back and set all parameters to the "optimal" values. Fine tune the servo system, using the difference equation that provides the required system performance. Setup a master/slave system, always sending the master's destinations last. Use the mnemonics reference section for detailed information about specific commands, and the table on page 4F-2 and the list on page 4F-10 for quick reference (there also is a summary list of all mnemonics and binary interface commands at the end of this section). Use the next page for recording optimal parameters and any additional notes about the system.

OPTIMAL PARAMETERS:

K_P	K_I	K_D	SPD	VEL	ACC	DAC	FFV	FFA	FFD
____	____	____	____	____	____	____	____	____	____
K_{X0}	K_{X1}	K_{X2}	K_{X3}	K_{X4}	K_{X5}	K_{X6}	K_{X7}	K_{X8}	K_{X9}
____	____	____	____	____	____	____	____	____	____
	K_{Y1}	K_{Y2}	K_{Y3}	K_{Y4}	K_{Y5}	K_{Y6}	K_{Y7}	K_{Y8}	K_{Y9}
	____	____	____	____	____	____	____	____	____

NOTES:

SERVO-AXIS BOARD MNEMONICS REFERENCE

The following mnemonics are recognized by the Servo-Axis board. The "*" preceding them denotes the address of the selected board, which can be "S", "T", "U", "V", "W", "X", "Y", or "Z". It will be set per table 4D-1 at the factory.

Mnemonics noted as "Read-only" or "Read/Write" can be read over HP-IB by issuing the mnemonic followed by a "?" (e.g., XREV?). These mnemonics can be read over the binary interface by sending the read command number then reading the specified number of data words.

Values can be written to "Read/Write" mnemonics by adding a number suffix to the HP-IB mnemonic (e.g., XDES 75.0), or by sending the write command number followed by the specified number of data words over the binary interface.

Mnemonics noted as "Commands" are not associated with a value that can be read or written. Issuing the command mnemonic over HP-IB or the command number over the binary interface causes some action to be taken, or causes the servo-axis board to change states.

The "GPIO Command Number" referred to throughout this section is the integer value that must be sent over the 16-bit binary interface to execute the given function. Command numbers 0 through 511 correspond to functions on the servo-axis board that receives them, regardless of its backplane address. Command numbers 512 through 4607 correspond to the same functions on servo-axis boards at specific backplane addresses (see page 4F-67).

The "GPIO Data Length" indicates how many 16-bit words are associated with the data item, and how the words will be interpreted (see also Binary Interface Operation, Data formats).

*ACC

Description: Profiling Acceleration Limit (g's)

Type: Read/Write Floating Point Data

Range: Greater than 0 Power-up Default: 0.1g

GPIO Command Number: 6 (write), 7 (read)

GPIO Data Length: Four-Word Floating Point

*ACC sets the maximum acceleration/deceleration for an internal position/velocity profiled move. (See *SVC).

*ACQ

Description: Acquires ("locks on" to) the Current Stage Position

Type: Command

GPIO Command Number: 200

*ACQ is a friendly way of safely turning on the servo system. It samples the position counter to get its current value, sets the Destination (*DES) equal to that value, turns on Drive Enable out- (*DRE1), and starts executing the selected difference equation. This prevents the system from receiving a large error signal when the drive is turned on.

In most instances, *ACQ is the recommended way to turn on the servo. See also *DRE.

Note: Errors may prevent this mnemonic from turning on the servo.

*BGO

Description: Buffered Position Go

Type: Read/Write Integer Data

Range: 0 to 4 Power-up Default: 0

GPIO Command Number: 211 through 215 (command), 60 (read)

GPIO Data Length: One-Word Integer (for read command only)

This command causes the servo-axis board to start adding the buffer values to the Reference Position on a sample by sample basis. The servo-axis board performs a linear interpolation between Reference Positions when *PTR 115;*DTA is set to a value greater than 0. (The value of *PTR 115;*DTA determines how many servo sample periods between buffered reference values).

This operation continues until the end of the buffer is reached. *BGO is only acted on by a master or independent servo-axis board. Slave boards are triggered by an inter-servo sync pulse. See also *PTR 115;*DTA.

*BGO values and corresponding GPIO command numbers and meanings are:

*BGO Value	GPIO Command	Meaning
0	211	Follow master
1	212	Forward POSITIVE
2	213	Forward NEGATIVE
3	214	Reverse POSITIVE
4	215	Reverse NEGATIVE

*CLK

Description: Selects Servo Clock Source

Type: Read/Write Integer Data

Range: 0 to 4 Power-up Default: 0

GPIO Command Number: 36 (write), 37 (read)

GPIO Data Length: One-Word Integer

*CLK selects the Servo Clock source as follows:

Value	Operation
0:	Internal Stand Alone. Makes this Servo-Axis board independent of any other board. This HP 10936A's Servo Clock is generated internally. Default value.
1:	Internal Master. This Servo-Axis board generates the Servo Clock for other Servo-Axis boards. The Servo Clock is sent over the top inter-servo connector.
2:	Slave to another HP 10936A. This Servo-Axis board receives its Servo Clock from another Servo-Axis board, through the top inter-servo connector.
3:	External Stand Alone. This Servo-Axis board is independent of other boards, and receives its Servo Clock through the rear panel connector.
4:	External Master. This Servo-Axis board generates the Servo Clock for other Servo-Axis boards, and receives its Servo Clock through the Rear Panel connector.

The Servo Clock source must correspond to the Master/Slave service mode selected (*SVC). See also the *MCM mnemonic.

The Servo-Axis board powers up in Internal Stand Alone mode.

***DAC**

Description: Profiling Delta Acceleration Limit (g's per second)

Type: Read/Write Floating Point Data

Range: Greater than zero   Power-up Default: 1g/sec

GPIO Command Number: 8 (write), 9 (read)

GPIO Data Length: Four-Word Floating Point

*DAC sets the rate of change of acceleration during a move. (only used for internal position/velocity profiling - see *SVC)

***DEQ**

Description: Difference Equation

Type: Read/Write Integer Data

Range: 0,1,2   Power-up Default: 0

GPIO Command Number: 38 (write), 39 (read)

GPIO Data Length: One-Word Integer

*DEQ selects the difference equation as follows:

0: PID Algorithm

1: Nth order IIR Filter

2: Customer downloaded equation

See Servo-Axis Board Setup — Difference Equations, for descriptions of each equation.

***DES**

Description: Destination Variable

Type: Read/Write Floating Point Data

Range:

English I/O units† : ±209 inches

Metric I/O units† : ±5310 millimeters

Raw†† or Lambda I/O : ±1,073,741,823 counts

† Values are for plane mirror interferometer. Range doubles if linear or single beam optics are used.

†† GPIO command #'s 12 and 13 may be used to write and read the destination in RAW counts as a two-word integer, regardless of the current units specified. (Values for other Servo-Axis Boards will be in their current units. See page 4F-67.)

Power-up Default: 0

GPIO Command Number: 10 (write), 11 (read)

GPIO Data Length: Four-Word Floating Point

Writing a value to *DES causes the stage to move to this position. The current units (*ENG, *LAM, *MET, or *RAW), compensation value (*TCN), deadpath (*DPD), and service mode (*SVC) are used when making the move (destinations are ignored when *SVC is set for velocity or Buffered reference profiling). The *ZRO mnemonic zeroes the destination as well as the position counter.

*DIR

Description: Measurement Direction Sense

Type: Read/Write Integer Data

Range: 0 or 1 Power-up Default: 0

GPIO Command Number: 40 (write), 41 (read)

GPIO Data Length: One-Word Integer

This mnemonic specifies the direction sense of the system. Set it so the *POS value is increasing as the reflector is moving away from the interferometer.

*DPD

Description: Deadpath Distance

Type: Read/Write Floating Point Data.

Range:

English I/O units†: ±209 inches

Metric I/O units†: ±5310 millimeters

Raw or Lambda I/O: ±1,073,741,823 counts

† Values are for plane mirror interferometer. Range doubles if linear or single beam optics are used.

Power-up Default: 0

GPIO Command Number: 24 (write), 25 (read)

GPIO Data Length: Four-Word Floating Point

*DPD sets the deadpath distance in current units (*ENG, *LAM, *MET, or *RAW).

The servo-axis board uses this distance to automatically compensate for air deadpath by applying the following equations when *ENG, *LAM or *MET units are selected (no deadpath correction is made when *RAW units are being used).

$$*\mathrm{POS} = \begin{pmatrix}\text{Accumulated}\\ \text{Counts}\end{pmatrix} \times (*\mathrm{TCN}) \times \begin{pmatrix}\text{unit}\\ \text{conversion}\\ \text{constant}\end{pmatrix} + (*\mathrm{TCN} - \mathrm{TCN}_0) \times \frac{*\mathrm{DPD}}{\mathrm{TCN}_0}$$

$$\text{Destination in Raw Counts} = \frac{*\mathrm{DES}}{(*\mathrm{TCN}) \times (\text{unit conversion constant})} - \frac{(*\mathrm{TCN} - \mathrm{TCN}_0)}{*\mathrm{TCN}} \times \frac{*\mathrm{DPD}}{(\mathrm{TCN}_0) \times \begin{pmatrix}\text{unit}\\ \text{conversion}\\ \text{constant}\end{pmatrix}}$$

where TCN_0 equals *TCN's value when the most recent *DPD or *ZRO mnemonic was received.

*DRE

Description: Drive Enable

Type: Read/Write Integer Data

Range: 0 or 1 Power-up Default: 0

GPIO Command Number: 42 (write), 43 (read)

GPIO Data Length: One-Word Integer

*DRE0 turns off the Drive Enable Out- line, forces the Motor Drive outputs to zero, and stops the difference equation from executing.

*DRE1 sets the Drive Enable Out- line to the active state, and starts the difference equation. Note that this command will have no effect if other conditions, such as errors, prevent the servo from being turned on.

The servo board powers up with the Drive Enable Out- line disabled. See also *ACQ.

*DTA

Description: Data Entry Mnemonic Used With *PTR

Type: Read/Write Floating Point Data

Range: Depends on *PTR value Power-up Default: 0

GPIO Command Number: Depends on *PTR value

*DTA is used in conjunction with *PTR for servo coefficient entry, and for setting and viewing miscellaneous parameters. When a value is sent to *PTR, the HP 10936A Servo-Axis board loads the data corresponding to that pointer value into *DTA. The data may be read by reading *DTA, or altered by writing to *DTA. To speed coefficient entry, *PTR is automatically incremented by 1 each time a coefficient value is written to *DTA. (*PTR values 1-18, 21-38, 41-58, 61-78, 81-98). See *PTR for *DTA's various meanings.

*EMD

Description: Error Mode

Type: Read/Write Integer Data

Range: 0 to 3 Power-up Default: 0

GPIO Command Number: 44 (write), 45 (read)

GPIO Data Length: One-Word Integer

Bit 0 value

- 0 - places the servo-axis board in Error Mode 0. Only measurement errors from this axis or a *DRE0 command will shut off the servo while in Error Mode 0. The servo board powers up in this Error Mode.
- 1 - places the servo-axis board in Error Mode 1. Any 5507A error (other axis error, programming error, etc.) will turn off the servo until it is again started with *DRE1 or *ACQ.

Bit 1 value

- 0 - sets the rear panel Error and Path Error Signals to represent this axis only
- 1 - sets the rear panel Error and Path Error Signals to represent the entire system (Equivalent to front panel LED's and HP-IB status byte).

*ENG

Description: Select English I/O Units

Type: Command

GPIO Command Number: 201

This command sets the I/O units to compensated (*TCN) inches. The mnemonics affected are *DES, *DPD, *MKR, *NUL, *POS, *VEL, and *PTR 104, 105 and 111. Note that *UCN may change *TCN's value and that slave boards use their master's *TCN. Bits 2 and 3 of *PTR113;*DTA indicate which units have been selected. Also see *LAM, *MET, and *RAW.

The Servo-Axis board powers up in Metric units.

*EXT

Description: External *POS Sample Trigger Source

Type: Command

GPIO Command Number: 205

When a servo-axis board is in external sample mode, the value returned with the *POS mnemonic is updated on the falling edge of the External Sample- input, by sending *PTR 109, or by sending GPIO command 218. Transitions on the External Sample- input which occur before the board is placed in external sample mode, and transitions which occur more rapidly than the servo board can update *POS are ignored. Bit 4 of *PTR113;*DTA indicates the current sample mode; 0 for internal or 1 for external.

The Motor Drive outputs are not affected by the *POS sample trigger source.

The power-up sample mode is Internal (see *INT).

*FFA

Description: Acceleration Feedforward Coefficient

Type: Read/Write Integer Data

Range: 0 to 32767 Power-up Default: 0

GPIO Command Number: 18 (write), 19 (read)

GPIO Data Length: One-Word Integer

*FFA sets the amount of acceleration feedforward to be used in calculating the Motor Drive outputs. See Tuning the Servo System .

*FFD

Description: Delta Acceleration Feedforward Coefficient

Type: Read/Write Integer Data

Range: 0 to 32767 Power-up Default: 0

GPIO Command Number: 20 (write), 21 (read)

GPIO Data Length: One-Word Integer

*FFD sets the amount of delta acceleration feedforward to be used in calculating the Motor Drive outputs. See Tuning the Servo System.

*FFV

Description: Velocity Feedforward Coefficient

Type: Read/Write Integer Data

Range: 0 to 32767 Power-up Default 0

GPIO Command Number: 16 (write), 17 (read)

GPIO Data Length: One-Word Integer

*FFV sets the amount of velocity feedforward to be used in calculating the Motor Drive outputs. See Tuning the Servo System .

***INT**

Description: Internal *POS Sample Trigger Source

Type: Command.

GPIO Command Number: 206

When a servo board is in Internal sample mode, the value returned with the *POS mnemonic is updated at the servo sample rate (set by *SPD) when drive is enabled, or at approximately 4 kHz when the drive is not enabled. Bit 4 of *PTR113;*DTA indicates the current sample mode.

The Motor Drive outputs are not affected by the *POS sample trigger source.

The power-up sample mode is Internal (see also *EXT).

***LAM**

Description: Select Lambda I/O Units

Type: Command

GPIO Command Number: 203

This command sets the I/O units to compensated (*TCN) λ units (λ/64, λ/128, or λ/256 depending on optics used -see *OPT). Bits 2 and 3 of *PTR113;*DTA indicate which units have been selected. The mnemonics affected are *DES, *DPD, *MKR, *NUL, *POS, *VEL, and *PTR 104, 105, and 111. Also see *ENG, *MET, and *RAW.

Note that *UCN may change *TCN's value and that slave boards use their master's *TCN.

The Servo-Axis board powers up in Metric units.

***MCM**

Description: Master Control Mask

Type: Read/Write Integer Data

Range: 0 to 255 Power-up Default: 0

GPIO Command Number: 46 (write) 47 (read)

GPIO Data Length: One-Word Integer

This mnemonic specifies which slave servo-axis boards this one is master over. The *SVC and *CLK values must be set appropriately on all servo-axis boards for proper operation of a master-slave system. The following table correlates *MCM's value to the possible board addresses.

ADDRESS	S	T	U	V	W	X	Y	Z
BIT	7	6	5	4	3	2	1	0
VALUE	128	64	32	16	8	4	2	1

For example, sending "XSVC1;YSVC2;ZSVC2;XCLK1;YCLK2;ZCLK2;XMCM3" puts the X Servo-Axis board as master of the Y and Z Servo-Axis boards.

*MET

Description: Select Metric I/O Units

Type: Command

GPIO Command Number: 202

This command sets the I/O units to compensated (*TCN) millimeters. Bits 2 and 3 of *PTR113;*DTA indicate which units have been selected. The mnemonics affected are *DES, *DPD, *MKR, *NUL, *POS, *VEL, and *PTR 104, 105, and 111. Also see *ENG, *LAM, and *RAW.

Note that *UCN may change *TCN's value and that slave boards use their master's *TCN.

The Servo-Axis board powers up in Metric units.

*MEX

Description: External Measurement Input Channel

Type: Command

GPIO Command Number: 207

Measurement inputs can come from one of two sources: Internal (HP 5518A Laser Head) or External (HP 10780B Receiver). *MEX selects the External source (HP 10780B).

The servo-axis board powers up in External measurement input mode.

*MIN

Description: Internal Measurement Input Channel

Type: Command

GPIO Command Number: 208

Measurement inputs can come from one of two sources: Internal (HP 5518A Laser Head) or external (HP 10780B Receiver). *MIN selects the internal one (HP 5518A).

The servo-axis board powers up in External measurement input mode.

*MKR

Description: Marker Position for Threshold and Window Outputs

Type: Read/Write Floating Point Data

Range:

English I/O units† : ±209 inches

Metric I/O units† : ±5310 millimeters

Raw†† or Lambda I/O : ±1,073,741,823 counts

† Values are for plane mirror interferometer. Range doubles if linear or single beam optics are used.

†† GPIO command #'s 28 and 29 may be used to write and read the destination in RAW counts as a two-word integer, regardless of the current units specified. (Values for other Servo-Axis Boards will be in their current units. See page 4F-67.)

Power-up Default: 0

GPIO Command Number: 26 (write) 27 (read)

GPIO Data Length: Four-Word Floating Point

*MKR sets, in current units:

1. The position where the Threshold output changes state (it is low when the stage position is less than *MKR's value).
2. The center location for the Window Outputs (this output is high when the stage is within *MKW's specified range of *MKR's value).

*MKW

Description: Marker Width

Type: Read/Write Integer Data

Range: 0 to 12 Power-up Default: 0

GPIO Command Number: 30 (Write) 31 (Read)

GPIO Data Length: One-Word integer

*MKW sets the range which will activate the rear panel Window output. This output is true (high) whenever the stage position is within *MKW's specified value of *MKR's location. The following table shows the ± window width for the different values (distances are for Plane Mirror optics, double the values for Linear and Single Beam optics, and halve them for $\lambda/256$ optics).

Value	Microinches	Microns
0	0.2	0.005
1	0.4	0.010
2	0.8	0.020
3	1.6	0.040
4	3.1	0.079
5	6.2	0.158
6	12.5	0.316
7	24.9	0.633
8	49.8	1.266
9	99.7	2.532
10	199.4	5.064
11	398.7	10.128
12	797.5	20.256

*NAM

Description: Board Name

Type: Read-only ASCII Data

GPIO Command Number: None (See GPIO command 2)

Requesting *NAM from the servo board returns the string "SRVO <CR><LF>& EOI".

***NUL**

Description: Null Distance (in current units)

Type: Read/Write Floating Point Data

Range:

English I/O units†: ±0 to 418 inches

Metric I/O units†: ±0 to 10620 millimeters

Raw†† or Lambda I/O: ±0 to 2,147,483,647 counts

† Values are for plane mirror interferometer. Range doubles if linear or single beam optics are used.

†† GPIO command #'s 34 and 35 may be used to write and read the destination in RAW counts as a two-word integer, regardless of the current units specified. (Values for other Servo-Axis Boards will be in their current units. See page 4F-67.)

Power-up Default: 0

GPIO Command Number: 32 (write), 33 (read)

GPIO Data Length: Four-Word Floating Point

*NUL sets the ± epsilon value used to determine the state of the NULL bit in the Servo-Axis board's miscellaneous status word (*PTR 113;*DTA). The current units (*ENG, *LAM, *MET, or *RAW) and compensation (*TCN) are used to convert the value sent to raw-count units. The Null bit is set when |current position − destination| < *NUL value. The system null status bit (in HP-IB Status Byte or *PTR 113;*DTA) is the AND of all NULL bits in an HP 5507A.

***OPT**

Description: Optics Type

Type: Read/Write Integer Data

Range: 0,1,2 Power-up Default: 1

GPIO Command Number: 48 (write), 49 (read)

GPIO Data Length: One-Word Integer

This mnemonic informs the servo-axis board which optics are installed, so that it may select the correct I/O unit conversion constants.

*OPT0: Linear and Single Beam optics (λ/64).

*OPT1: Plane mirror optics (λ/128) - Power-up Default.

*OPT2: Special λ/256 optics.

***OUT**

Description: Motor Drive Output

Type: Read/Write Integer Data

Range: 0 to 7 Power-up Default: 1

GPIO Command Number: 50 (write), 51 (read)

GPIO Data Length: One-Word Integer

The three low order bits of *OUT's value enable the various Motor Drive outputs. Setting any bit to 1 turns on the selected output option as follows:

Bit	value	Output
0	1	$\pm$10 V from a 16-bit DAC (Power-up Default)
1	2	Pulse-width modulated output
2	4	16-bit digital output

Note: The 16-bit binary output port is used if bit 2 is set, preventing any outputs through the binary interface.

*POS

Description: Position Output Variable

Type: Read-only Floating Point Data

GPIO Command Number: 14 (read)

GPIO Data Length: Four-Word Floating Point

*POS returns the stage position in current units (*ENG, *LAM, *MET, or *RAW††). All but *RAW are compensated by the *TCN and *DPD values.

The value returned by *POS is updated according to the *POS Sample Trigger source (*EXT or *INT). See also *PTR 109.

†† GPIO command 15 will return the position in RAW counts as a two word integer regardless of the current I/O units. (Values for other Servo-Axis Boards will be in their current units. See page 4F-67.)

*PRE

Description: Position Counter Preset

Type: Command

GPIO Command Number: 209

*PRE presets the position counter to the value currently in the Marker register (*MKR). The lowest five bits of the position counter are zeroed by the *PRE command, thereby placing a limit on the degree of accuracy to which the counter may be preset.

To preset the counter to a specific position, load that position into the marker register and execute the *PRE command. The marker register contents will be copied into the position counter.

*PTR

Description: Data Entry Pointer

Type: Read/Write Integer data

Range: 0 to 117 Power-up Default: 0

GPIO Command Number: Depends on *PTR value.

*PTR is used in conjunction with *DTA for servo coefficient entry, and for setting and viewing miscellaneous parameters. When a value is sent to *PTR, the HP 10936A Servo-Axis board loads the data corresponding to that pointer value into *DTA. The data may be read by reading *DTA, or altered by writing to *DTA. To speed coefficient data entry, *PTR is automatically incremented by 1 each time a coefficient value is written to *DTA. (*PTR values 1-18, 21-38 ..., 81-98).

Some *PTR values have no corresponding data and are used only as commands. The functions of the various *PTR values are listed as separate mnemonics on the following pages and summarized in *Table 4F-5*.

***PTR0**

Description: Start using Coefficient Bank 1 (*PTR1 thru *PTR19)

Type: Command

GPIO Command Number: 100

The Servo-Axis board double buffers all coefficients to prevent the selected difference equation from using a partially updated set of values. The *PTR0 command tells the board that this set of coefficients is complete, and to start using them. Typically it will be sent after the coefficients are loaded or modified (*PTR1;*DTA through *PTR19;*DTA). It can also be used to switch from a different bank of coefficients (banks 2 through 5) to Bank 1 coefficients.

***PTR1;*DTA through *PTR19;*DTA**

Description: Difference Equation Coefficients - Bank 1

Type: Floating Point Data

Range: -524,288 to 524,287 Default: 0

Resolution: 1/4096

GPIO Command Number: 101 to 119 (Write)

GPIO Data Length: Four-Word Floating Point

These *PTRx;*DTA values are used by the selected difference equation to process the position error value. Their significance for each equation is listed below (a downloaded equation has access to all of them and it will define their significance).

Writing values to these auto increment *PTR's value, speeding data entry.

Note that updates to any of these values will not be available to the difference equation until a *PTR0 command is sent.

*PTR Number	GPIO Command Number (Write)	Coefficients
1	101	Kp for PID, Kx0 for IIR
2	102	Ki for PID, Ky1 for IIR
3	103	Kd for PID, Kx1 for IIR
4	104	Not Used for PID, Ky2 for IIR
5	105	Not Used for PID, Kx2 for IIR
6	106	Not Used for PID, Ky3 for IIR
7	107	Not Used for PID, Kx3 for IIR
8	108	Not Used for PID, Ky4 for IIR
9	109	Not Used for PID, Kx4 for IIR
10	110	Not Used for PID, Ky5 for IIR
11	111	Not Used for PID, Kx5 for IIR
12	112	Not Used for PID, Ky6 for IIR
13	113	Not Used for PID, Kx6 for IIR
14	114	Not Used for PID, Ky7 for IIR
15	115	Not Used for PID, Kx7 for IIR
16	116	Not Used for PID, Ky8 for IIR
17	117	Not Used for PID, Kx8 for IIR
18	118	Not Used for PID, Ky9 for IIR
19	119	Not Used for PID, Kx9 for IIR