# PP Software

## 1. PP Command/Data Interfaces

The PP software exchanges commands and/or data over the following interfaces:

- The host SCSI interface (via the 53C720 coprocessor)
- The Data Acquisition Processor (DAP) via the PP status register
- The four possible Controller Modules and 16 possible Output Modules
- Miscellaneous devices via the fault register

The nature of the commands and/or data that can be exchanged over each of these interfaces is detailed in the following sections.

### 1.1 Host SCSI Interface

The SCSI interface is used to transmit commands from the host computer to the PP. Some of these commands may also request that data be transmitted to or from the host computer by the PP. The PP implements two kinds of SCSI commands: mandatory commands which all SCSI-2 compliant devices must support, and vendor specific Group 7 commands which provide the special functionality required by the PP. In addition to SCSI *commands*, the PP also implements certain SCSI *messages* which are required by the protocol.

The PP implements eight logical units, each of which is functionally identical. There is no functional distinction between the logical units. Any command recognized by the PP can be executed on any of the eight logical units. The existence of multiple logical units allows the host computer to have multiple commands pending.

Each of the implemented SCSI commands and messages are described in the following sections.

#### 1.1.1 SCSI Commands

The PP implements the following SCSI commands:

##### 1.1.1.1 INQUIRY

This is a mandatory Group 0 command. In response to this command the PP always returns the following data packet:

<table>
<tr><td>Bit<br>Byte</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="3">Peripheral qualifier<br>0h</td><td colspan="5">Peripheral device type<br>1Fh</td></tr>
<tr><td>1</td><td>RMB<br>0h</td><td colspan="7">Device-type modifier<br>0h</td></tr>
<tr><td>2</td><td colspan="2">ISO version<br>0h</td><td colspan="3">ECMA version<br>0h</td><td colspan="3">ANSI version<br>2h</td></tr>
</table>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | AENC 0h | TrmIO 0h | Reserved 0h | | Response data format 2h | | | |
| 4 | Additional Length 11h | | | | | | | |
| 5 | Reserved 0h | | | | | | | |
| 6 | Reserved 0h | | | | | | | |
| 7 | RelA 0h | WB32 0h | WB16 0h | Synch 1h | Link 0h | Res 0 | Que 0h | SftRe 0h |
| 8 | Vendor Indentification 55h - “U” | | | | | | | |
| 9 | 56h - “W” | | | | | | | |
| 10 | 20h - “ “ | | | | | | | |
| 11 | 43h - “C” | | | | | | | |
| 12 | 48h - “H” | | | | | | | |
| 13 | 45h - “E” | | | | | | | |
| 14 | 4Dh - “M” | | | | | | | |
| 15 | 20h - “ “ | | | | | | | |
| 16 | Product Identification 4Eh - N | | | | | | | |
| 17 | 4Dh - M | | | | | | | |
| 18 | 52h - R | | | | | | | |
| 19 | 20h “ “ | | | | | | | |
| 20 | 50h - P | | | | | | | |
| 21 | 50h - P | | | | | | | |

#### 1.1.1.2 REQUEST SENSE

This is a mandatory Group 0 command. It returns the current status of the selected logical unit. The PP returns the sense information in the following vendor specific data packet format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Error code 7Fh | | | | | | | |
| 1-6 | Reserved 0h | | | | | | | |

| | |
|---|---|
| 7 | Sense Key<br>See table below |

In this implementation the *Sense Key* returns vendor specific information. The possible values of the sense key are defined in the table that immediately follows.

| | |
|---|---|
| The last command executed successfully. | NO_SENSE<br>00h |
| The allocation field in a command packet was too small to receive the data requested. | ALLOC TOO SMALL<br>02h |
| The data length field in a command packet is to large to be accepted by the PP. | BUF TOO BIG<br>03h |
| The associated controller is not halted as is required for successful execution of the command. | NOT HALTED<br>04h |
| The associated controller is not running as is required for successful execution of the command. | NOT RUNNING<br>05h |
| The specified controller already had a command of this type pending. | COMMAND ALREADY PENDING<br>06h |
| The command contained an invalid field. | BAD FIELD<br>07h |
| The subprogram RAM became full before all the states in a LOAD RAM command could be loaded. | RAM FULL<br>08h |
| A controller deadlock condition was resolved automatically incrementing the FIFO sync counter for that controller. | FSYNC INC<br>0Ah |
| The command contained an invalid controller number. | BAD CTRL NUM<br>0Bh |
| The command included a descriptor that contained an invalid output card number. | BAD OCARD NUM<br>0Ch |
| The command included a descriptor that contained an invalid state match register number. | BAD SM REG NUM<br>0Dh |
| The DAP failed to respond to a initialization attempt during execution of an INITIALIZE PP command. | NO DAP RESPONSE<br>0Eh |
| The external faults cannot be reset because of an existing unmasked fault. | EXT FAULT<br>0Fh |
| A field descriptor contained an invalid offset field. | BAD OFFSET FIELD<br>10h |
| A field descriptor contained an invalid category field. | BAD CATEGORY FIELD<br>11h |
| A field descriptor contained an invalid card field. | BAD CARD FIELD<br>12h |
| The command was not recognized by the command dispatcher in the main loop (internal error). | BAD COMMAND<br>13h |
| The command was not recongnized. | ILLEGAL_REQUEST |

| | |
|---|---|
| | 14h |
| A hardware diagnostic failed. | HARDWARE_ERROR<br>15h |
| Previous command was aborted because of a SCSI bus reset. | ABORTED_COMMAND<br>16h |

#### 1.1.1.3 TEST UNIT READY

This is a mandatory Group 0 command. In this implementation, logical units are always considered ready. This command will always return a status of GOOD. No data packet is returned.

#### 1.1.1.4 INITIALIZE PP

This is a vendor specific Group 7 command. The host computer uses this command to initialize the PP to a known condition at the beginning of an experiment. The following initialization actions are performed:

- An INIT command is send to each of the control modules (see PP Controller Card, Section 3.3.1).
- All assembly register fields in all control modules and all output modules are cleared to zero.
- The controller select registers in all output modules are cleared to zero.
- An *Initialize DAP* status message is sent to the DAP (see Section 1.2).
- External faults are reset.
- The *status reference number* is incremented.
- Clears the Next Ram Address register for each controller to zero.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations while the above initialization steps are completed. After the initialization action is completed, the PP will immediately reconnect and complete the operation with a status of GOOD or CHECK CONDITION. If the DAP fails to respond to the *Initialize DAP* status message the sense key will be NO DAP RESPONSE. If the attempt to reset the external faults is not successful, the sense key is EXT FAULT. The format of the command packet is as follows:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code<br>E0h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-11 | Reserved<br>0h | | | | | | | |
| 12 | Control | | | | | | | |

#### 1.1.1.5 START

This is a vendor specific Group 7 command. The host computer uses this command to start the specified controller.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. If an invalid controller number was specified, the PP will immediately reconnect and complete the operation with a status of BAD CTRL NUM. Otherwise, the PP issues a WRT FBSA to write the FIFO buffer starting address of the specified controller with the Next RAM Address for that controller. The PP then issues a RUN command to the specified controller and immediately reconnects and completes the operation with a status of GOOD. The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code E1h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved 0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8-11 | Reserved 0h | | | | | | | |
| 12 | Control | | | | | | | |

#### 1.1.1.6 STOP

This is a vendor specific Group 7 command. The host computer uses this command to halt all the controllers. This is distinguished from an ABORT command (see 1.1.1.7) by the fact that the resulting status passed to the DAP will be STOPPED rather than ABORTED.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. The PP then issues a HALT command to each of the four controllers, starting with controller #1. After issuing a HALT command to each controller the PP will immediately reconnect and complete the operation with a status of GOOD. The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code E3h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-11 | Reserved 0h | | | | | | | |

| | |
|---|---|
| 12 | Control |

#### 1.1.1.7 ABORT

This is a vendor specific Group 7 command. The host computer uses this command to halt all the controllers. This is distinguished from a HALT command (see 1.1.1.6) by the fact that the resulting status passed to the DAP will be ABORTED rather than STOPPED.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. The PP then issues a HALT command to each of the four controllers, starting with controller #1. After issuing a HALT command to each controller the PP will immediately reconnect and complete the operation with a status of GOOD. The format of the command packet is as follows:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code<br>E4h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-11 | Reserved<br>0h | | | | | | | |
| 12 | Control | | | | | | | |

#### 1.1.1.8 GET NEXT STATUS

This is a vendor specific Group 7 command. The host computer uses this command to fetch the PP status information. The PP will always respond to this command by immediately performing a SCSI disconnect so as to free the bus for other operations. At some later time, the PP will reconnect and will respond to this command as appropriate.

If the Global Status is RUNNING, the PP will not respond until such time as the PP's internally maintained *Status Reference Number* becomes greater than the *Request Number* in the command packet. At such time as the PP's internal *Status Reference Number* becomes greater than the *Request Number* in the command packet, the PP will reconnect to the SCSI bus and transmit the status data. The *Status Reference Number* is incremented whenever the status of any of the four control modules changes or by action of the INITIALIZE PP command.

If the Global Status is *not* RUNNING, the PP will respond immediately.

The GET NEXT STATUS command packet has the following format:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code<br>E5h | | | | | | | |

| | | |
|---|---|---|
| 1 | Logical Unit # | Reserved |
| 2 | Reserved<br>0h | |
| 3 | Reserved<br>0h | |
| 4 | Request Number - MSB | |
| 5 | ... | |
| 6 | ... | |
| 7 | Request Number - LSB | |
| 8 | Data Length - MSB | |
| 9 | ... | |
| 10 | ... | |
| 11 | Data Length - LSB | |
| 12 | Control | |

The Data Length field must contain the length of the buffer (in bytes) that has been allocated to receive the data. If the command is successful the status data will be returned in a data packet with the following format:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Status Response Number - MSB | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | 00h | | | | | | | |
| 3 | Status Response Number - LSB | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | 00h | | | | | | | |
| 6 | 00h | | | | | | | |
| 7 | Global Status | | | | | | | |
| 8-27 | Controller 1 Status Descriptor | | | | | | | |
| 28-47 | Controller 2 Status Descriptor | | | | | | | |
| 48-67 | Controller 3 Status Descriptor | | | | | | | |
| 68-87 | Controller 4 Status Descriptor | | | | | | | |

The *Status Response Number* field returned in the data packet contains the PP *Status Reference Number* that was current at the time of the response. The host should use this returned value as the *Request Number* in the next GET NEXT STATUS command it sends to the host. This will request that the PP respond with

more status data as soon as newer data is available. The host can request an immediate response by using a *Request Number* of zero.

The *Global Status* field contains the same composite status byte that the PP sends to the DAP (see Section 1.2). The remainder of the packet consists of four controller status descriptors, one four each controller. The format of the controller status descriptor is shown in the table found immediately below. Each descriptor contains a 16 bit *Controller Status Register* field and four 24 bit *ESR* (Experiment State Register) fields which return the controller register contents.

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | Controller Status Register - MSB | | | | | | | |
| 3 | Controller Status Register - LSB | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | ESR1 - MSB | | | | | | | |
| 6 | ... | | | | | | | |
| 7 | ESR1 - LSB | | | | | | | |
| 8 | 00h | | | | | | | |
| 9 | ESR2 - MSB | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | ESR2 - LSB | | | | | | | |
| 12 | 00h | | | | | | | |
| 13 | ESR3 - MSB | | | | | | | |
| 14 | ... | | | | | | | |
| 15 | ESR3 - LSB | | | | | | | |
| 16 | 00h | | | | | | | |
| 17 | ESR4 - MSB | | | | | | | |
| 18 | ... | | | | | | | |
| 19 | ESR4 - LSB | | | | | | | |

#### 1.1.1.9 GET CONFIGURATION

This is a vendor specific Group 7 command. The host computer uses this command to determine the hardware configuration of the PP. The command returns the type number of each controller card and each output card that is plugged into the PP.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations while the configuration is determined. If all controllers are halted, the PP will poll the daughter cards to read

the configuration and then reconnect and return the configuration data. If any of the controllers are running, the PP will immediately reconnect and complete the operation with a status of CHECK CONDITION and a sense key of NOT HALTED without returning any data.

The GET CONFIGURATION command packet has the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code E6h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-7 | Reserved 0h | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Data Length - LSB | | | | | | | |
| 12 | Control | | | | | | | |

The *Data Length* field must contain the length (in bytes) of the buffer that has been allocated to receive the data. The configuration data is returned in a packet with the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | Controller 1 Type - MSB | | | | | | | |
| 3 | Controller 1 Type - LSB | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | 00h | | | | | | | |
| 6 | Controller 2 Type MSB | | | | | | | |
| 7 | Controller 2 Type LSB | | | | | | | |
| 8 | 00h | | | | | | | |
| 9 | 00h | | | | | | | |
| 10 | Controller 3 Type - MSB | | | | | | | |
| 11 | Controller 3 Type - LSB | | | | | | | |
| 12 | 00h | | | | | | | |

| | |
|---|---|
| 13 | 00h |
| 14 | Controller 4 Type - MSB |
| 15 | Controller 4 Type - LSB |
| 16 | 00h |
| 17 | 00h |
| 18 | Output Card 1 Type - MSB |
| 19 | Output Card 1 Type - LSB |
| 20-79 | Type data for Output Cards 2 - 16 |

#### 1.1.1.10 LOAD RAM

This command is used to load fields in the state assembly register and write the assembled states into the subprogram RAM of the specified controller. The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. If an invalid controller number was specified, the PP will immediately reconnect and complete the operation with a status of BAD CTRL NUM. This command can only be executed successfully if the specified controller is halted. If the controller specified in the command packet *is* running, the PP will immediately reconnect and terminate the command with a status of CHECK CONDITION and a sense key of NOT HALTED. Likewise, if the controller specified in the command packet already has a LOAD RAM command pending the PP will immediately reconnect and terminate the command with a status of CHECK CONDITION and a sense key of COMMAND ALREADY PENDING. The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code E7h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved 0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Data Length - LSB | | | | | | | |
| 12 | Control | | | | | | | |

After validating the command, the PP will reconnect and transfer the number of field descriptors specified by the DATA LENGTH field of the command packet. The PP

will then again disconnect and remain disconnected until all the specified fields have been loaded.

The data packet transferred consists of one or more field descriptors. Each field descriptor consists of four bytes of information organized as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Load State | Category | | Card Number | | | | |
| 1 | Field Offset - LSB | | | | | | | |
| 2 | Field Data - MSB | | | | | | | |
| 3 | Field Data - LSB | | | | | | | |

The two FIELD DATA bytes comprise a 16 bit word which specifies the actual data to be loaded into the specified assembly register field. The two bit CATEGORY specifies what kind of field is being loaded, i.e., a mother board field, a controller card field, or an output card field. This is encoded as follows:

| **Type of field selected** | **CATEGORY value** |
|---|---|
| None | 0h |
| Mother Board field | 1h |
| Controller Card field | 2h |
| Output Card field | 3h |

The five bit CARD NUMBER specifies, for an output card, the number of the card slot in which the card is located. Output card slots are numbered starting with one and ending with 16. The CARD NUMBER is ignored in the case of a controller field, since the controller number is specified in the command packet, and in the case of a mother board field.

The FIELD OFFSET specifies the relative address of the assembly register field on the specified card. Assembly register fields are addressed as 16 bit entities with the first field on a card being numbered as zero, the second field being numbered as one, etc.

If the LOAD STATE bit is set it indicates that after the field is loaded into the assembly register the entire assembled state should be written into the subprogram RAM associated with the specified controller. The state is written at the address specified by the PP's Next RAM Address register for the specified controller. Each time a state is loaded the Next RAM Address register is incremented to point to the next available location.

As was stated above, the PP disconnects after receiving the data packet of field descriptors and remains disconnected until such time as all the specified states have been loaded into subprogram RAM. When this process completes successfully the PP will then reconnect and complete the command with a status of GOOD.

There are several possible error conditions that will cause the command to complete prematurely with a status of CHECK CONDITION. If the field descriptor contains an invalid CATEGORY, CARD NUMBER, or FIELD OFFSET the sense key will be BAD

FIELD. If the subprogram RAM becomes full before all the states are loaded the sense key will be RAM FULL.

#### 1.1.1.11 LOAD FIFO

This command is used to load fields in the state assembly register and write the assembled states into the state FIFO of the specified controller. The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. If an invalid controller number was specified, the PP will immediately reconnect and complete the operation with a status of BAD CTRL NUM. If the controller specified in the command packet already has a LOAD FIFO command pending the PP will immediately reconnect and terminate the command with a status of CHECK CONDITION and a sense key of COMMAND ALREADY PENDING. Likewise, if the DATA LENGTH field specifies a length that is too large for the PP's buffer (1000 field descriptors) the PP will immediately reconnect and terminate the command with a status of CHECK CONDITION and a sense key of BUF TOO BIG. If none of these errors occur, the PP will reconnect, transfer the data, and then complete the command with a normal status.

Completion of the command *does not* indicate that the data has been successfully loaded into the controller FIFO. The PP will load the data into the FIFO as space becomes available. The host computer must use the FIFO WAIT command to determine when loading has been completed. If an error occurs while loading the FIFO the error will be reported via the sense key of the FIFO WAIT command. Errors in the data being loaded to the FIFO can result in sense keys of BAD OFFSET FIELD, BAD CARD FIELD, or BAD CATEGORY FIELD.

A controller FIFO load can only be executed successfully if the specified controller is running. If the controller specified in the command packet *is* halted, or halts before all the data is successfully loaded into the FIFO, the FIFO WAIT command returns a sense key of NOT RUNNING.

If, while data is being loaded into the controller FIFO, the controller FIFO becomes full while the controller remains in a FIFO sync wait, the PP will automatically increment the controllers FIFO sync counter to resolve this deadlocked condition. In this case the LOAD FIFO operation will successfully complete, but will the status returned by FIFO WAIT will be CHECK CONDITION with a sense key of FSYNC INC. This is intended as a warning, not as a fatal error condition.

The format of the LOAD_FIFO command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code E8h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved 0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |

| | |
|---|---|
| 10 | ... |
| 11 | Data Length - LSB |
| 12 | Control |

The format of the data packet is the same as was described for the LOAD RAM command in Section 1.1.1.10

#### 1.1.1.12 READ NEXT RAM ADDRESS

This command is used to read the contents of the Next RAM Address register from the each of the controllers. The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. The PP will then reconnect and return the contents of the four registers as soon as the registers have been read. The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code E9h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-7 | Reserved 0h | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Data Length - LSB | | | | | | | |
| 12 | Control | | | | | | | |

The *Data Length* field must contain the length (in bytes) of the buffer that has been allocated to receive the data. The data is returned in a packet with the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | Controller 1 Next RAM Address - MSB | | | | | | | |
| 3 | Controller 1 Next RAM Address - LSB | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | 00h | | | | | | | |
| 6 | Controller 2 Next RAM Address MSB | | | | | | | |

| | |
|---|---|
| 7 | Controller 2 Next RAM Address LSB |
| 8 | 00h |
| 9 | 00h |
| 10 | Controller 3 Next RAM Address - MSB |
| 11 | Controller 3 Next RAM Address - LSB |
| 12 | 00h |
| 13 | 00h |
| 14 | Controller 4 Next RAM Address - MSB |
| 15 | Controller 4 Next RAM Address - LSB |

#### 1.1.1.13 ALLOCATE OUTPUT CARDS

This is a vendor specific Group 7 command. The host computer uses this command to allocate output cards to the specified controller.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. If any of the controllers are running, the PP will immediately reconnect and terminate the command with a status of CHECK CONDITION and a sense key of NOT HALTED.

The ALLOCATE OUTPUT CARDS command packet has the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code EAh | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved 0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Data Length - LSB | | | | | | | |
| 12 | Control | | | | | | | |

After validating the command, the PP will reconnect and transfer the number of output card descriptors specified by the DATA LENGTH field of the command packet. The PP will then again disconnect and remain disconnected until all the specified output cards have been allocated.

The data packet transferred consists of one or more output card descriptors. Each output card descriptor consists of four bytes of information organized as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | 00h | | | | | | | |
| 3 | Output Card Slot Number | | | | | | | |

After transferring the data packet the PP remains disconnected until such time as all the specified output cards have been allocated. When and if this process completes successfully the PP will then reconnect and complete the command with a status of GOOD.

It is possible for this command to complete prematurely with a status of CHECK CONDITION. If the field descriptor contains an invalid OUTPUT CARD SLOT NUMBER the sense key will be BAD FIELD.

#### 1.1.1.14 LOAD SM REGISTERS

This is a vendor specific Group 7 command. The host computer uses this command to load one or more state match registers in the specified controller.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations.

The LOAD SM REGISTERS command packet has the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code EBh | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved 0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Data Length - LSB | | | | | | | |
| 12 | Control | | | | | | | |

After validating the command, the PP will reconnect and transfer the number of SM register descriptors specified by the DATA LENGTH field of the command packet. The PP will then again disconnect and remain disconnected until all the specified state match registers have been loaded.

The data packet transferred consists of one or more SM register descriptors. Each SM register descriptor consists of eight bytes of information organized as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | 00h | | | | | | | |
| 3 | SM Register Number | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | Register Value - MSB | | | | | | | |
| 6 | ... | | | | | | | |
| 7 | Register Value - LSB | | | | | | | |

The SM REGISTER NUMBER specifies one of the four state match registers in a controller and has legal values in the range of 1-4. The REGISTER VALUE is the 24 bit value that is to be loaded into the specified register. After transferring the data packet the PP remains disconnected until such time as all the specified registers have been loaded. When and if this process completes successfully the PP will then reconnect and complete the command with a status of GOOD.

It is possible for this command to complete prematurely with a status of CHECK CONDITION. If the register descriptor contains an invalid SM REGISTER NUMBER the sense key will be BAD SM REG NUM.

#### 1.1.1.15 LOAD CONDITIONAL ACTION REGISTER

This is a vendor specific Group 7 command. The host computer uses this command to load the conditional action register in the specified controller.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations.

The LOAD CONDITIONAL ACTION REGISTER command packet has the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code<br>ECh | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved<br>0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8-10 | Reserved<br>0h | | | | | | | |

| | |
|---|---|
| 11 | Register Value |
| 12 | Control |

Since the register value is contained in the command packet, this command does not transmit a separate data packet. When the register has been successfully loaded the PP will then reconnect and complete the command with a status of GOOD.

It is possible for this command to complete prematurely with a status of CHECK CONDITION. If the CONTROLLER NUMBER is invalid the sense key will be BAD CTRL NUM.

#### 1.1.1.16 LOAD FAULT MASK REGISTER

This is a vendor specific Group 7 command. The host computer uses this command to load the fault mask register.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations.

The LOAD FAULT MASK REGISTER command packet has the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code EDh | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-9 | Reserved 0h | | | | | | | |
| 10 | Register Value - MSB | | | | | | | |
| 11 | Register Value - LSB | | | | | | | |
| 12 | Control | | | | | | | |

Since the register value is contained in the command packet, this command does not transmit a separate data packet. When the register has been successfully loaded the PP will then reconnect and complete the command with a status of GOOD.

#### 1.1.1.17 READ EXTERNAL FAULT REGISTER

This is a vendor specific Group 7 command. The host computer uses this command to read the contents of the external fault register.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. The PP will then reconnect and return the contents of the register as soon as the register has been read. The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| 0 | Operation Code<br>EEh | |
| 1 | Logical Unit # | Reserved |
| 2-7 | Reserved<br>0h | |
| 8 | Data Length - MSB | |
| 9 | ... | |
| 10 | ... | |
| 11 | Data Length - LSB | |
| 12 | Control | |

The *Data Length* field must contain the length (in bytes) of the buffer that has been allocated to receive the data. The data is returned in a packet with the following format:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | External Fault Register- MSB | | | | | | | |
| 3 | External Fault Register - LSB | | | | | | | |

#### 1.1.1.18 READ STATE NUMBER

This command is used to read the contents of the State Number register from each of the controllers. The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. The PP will then reconnect and return the contents of the four registers as soon as the registers have been read. The format of the command packet is as follows:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code<br>EFh | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-7 | Reserved<br>0h | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |

| | |
|---|---|
| 11 | Data Length - LSB |
| 12 | Control |

The *Data Length* field must contain the length (in bytes) of the buffer that has been allocated to receive the data. The data is returned in a packet with the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | Controller 1 State Number - MSBh | | | | | | | |
| 2 | ... | | | | | | | |
| 3 | Controller 1 State Number - LSB | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | Controller 2 State Number - MSB | | | | | | | |
| 6 | ... | | | | | | | |
| 7 | Controller 2 State Number - LSB | | | | | | | |
| 8 | 00h | | | | | | | |
| 9 | Controller 3 State Number - MSB | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Controller 3 State Number - LSB | | | | | | | |
| 12 | 00h | | | | | | | |
| 13 | Controller 4 State Number - MSB | | | | | | | |
| 14 | ... | | | | | | | |
| 15 | Controller 4 State Number - LSB | | | | | | | |

#### 1.1.1.19 READ STATE MEMORY

This is a vendor specific Group 7 command. The host computer uses this command to read the contents of the specified state memory location.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations while the configuration is determined. If all controllers are halted, the PP will read the state memory and then reconnect and return the memory data. If any of the controllers are running, the PP will immediately reconnect and complete the operation with a status of CHECK CONDITION and a sense key of NOT HALTED without returning any data.

The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

| | | |
|---|---|---|
| 0 | Operation Code<br>F0h | |
| 1 | Logical Unit # | Reserved |
| 2-5 | Reserved<br>0h | |
| 6 | State Memory Address - LSB | |
| 7 | State Memory Address - MSB | |
| 8 | Data Length - MSB | |
| 9 | ... | |
| 10 | ... | |
| 11 | Data Length - LSB | |
| 12 | Control | |

The *Data Length* field must contain the length (in bytes) of the buffer that has been allocated to receive the data. The data is returned in a packet with the following format:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Mother Board Field Block | | | | | | | |
| 64 | Controller 1 Field Block | | | | | | | |
| 128 | ... | | | | | | | |
| 256 | Controller 4 Field Block | | | | | | | |
| 320 | Output Card 1 Field Block | | | | | | | |
| 384 | ... | | | | | | | |
| 1280 | Output Card 16 Field Block | | | | | | | |

Each field block returns the contents of the 16 possible 16 bit fields associated with that device. Each block has the following format:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | Field Offset 0 - MSB | | | | | | | |
| 3 | Field Offset 0 - LSB | | | | | | | |
| 4 | ... | | | | | | | |
| 60 | 00h | | | | | | | |

| | |
|---|---|
| 61 | 00h |
| 62 | Field Offset 15 - MSB |
| 63 | Field Offset 15 - LSB |

#### 1.1.1.20 FIFO WAIT

This is a vendor specific Group 7 command. The host computer uses this command to query the status of the last LOAD FIFO command issued to the specified controller.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations. If the PP determines that the last LOAD FIFO command on the specified controller has not yet completed (i.e., all the data has not yet been loaded into the controller FIFO) the command will not reconnect until such time as the LOAD FIFO has been completed (either successfully or due to an error).

If the PP determines that no data from a pending load FIFO command is waiting to be loaded into the FIFO of the specified controller (i.e., the last LOAD FIFO command has completed or no load FIFO commands have been issued for this controller) the PP will immediately reconnect and return a status. If the last LOAD FIFO command completed successfully, or no such command has been issued to this controller, the FIFO WAIT command will return with a status of good. If the last LOAD FIFO command terminated prematurely due to an error, the FIFO WAIT command will return a status of CHECK CONDITION and a sense key that reflects the error that terminated the LOAD FIFO command. See LOAD FIFO, 1.1.1.11 for a description of the possible errors.

Additionally, if the FIFO WAIT command remains inactive longer than the *command time-out* period, the command will reconnect and complete with a status of CHECK CONDITION and a sense key of TIMEOUT. In this case the host must re-issue the command to continue waiting for the FIFO load operation to complete.

The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code F1h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-6 | Reserved 0h | | | | | | | |
| 7 | Controller Number | | | | | | | |
| 8-11 | Reserved 0h | | | | | | | |
| 12 | Control | | | | | | | |

#### 1.1.1.21 TEST STATE MEMORY

This is a vendor specific Group 7 command. The host computer uses this command to test the integrity of state memory.

The PP always responds to this command by immediately performing a SCSI disconnect operation so as to free the bus for other operations while the test is performed. If all controllers are halted, the PP will perform a diagnostic on the state memory at the specified field address and then reconnect and report the results. If any of the controllers are running, the PP will immediately reconnect and complete the operation with a status of CHECK CONDITION and a sense key of NOT HALTED without performing the diagnostic.

The command performs diagnostic tests on all 65K memory locations associated with the field address specified by the command packet. Two tests are performed: a basic checkerboard (alternating ones and zeros) to confirm data integrity and an addressing test. When the tests are completed (or terminated because there are two many errors to report) the PP reconnects and returns a status of GOOD.

The format of the command packet is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Operation Code F2h | | | | | | | |
| 1 | Logical Unit # | | | Reserved | | | | |
| 2-3 | Reserved 0h | | | | | | | |
| 4 | Byte Flag | Category | | Card Number | | | | |
| 5 | Field Offset | | | | | | | |
| 6 | Controller # - MSB | | | | | | | |
| 7 | Controller # - LSB | | | | | | | |
| 8 | Data Length - MSB | | | | | | | |
| 9 | ... | | | | | | | |
| 10 | ... | | | | | | | |
| 11 | Data Length - LSB | | | | | | | |
| 12 | Control | | | | | | | |

The BYTE FLAG field is set to specify that the field under test is one byte, rather than two bytes, in length.

The two bit CATEGORY specifies what kind of state memory field is being tested, i.e., a mother board field, a controller card field, or an output card field. This is encoded as follows:

| **Type of field selected** | **CATEGORY value** |
|---|---|
| None | 0h |

| | |
|---|---|
| Mother Board field | 1h |
| Controller Card field | 2h |
| Output Card field | 3h |

The five bit CARD NUMBER specifies, for an output card or a controller card, the number of the card slot in which the card is located. Output card slots are numbered starting with one and ending with 16. Controller cards are numbered starting with one and ending with four.

The FIELD OFFSET specifies the relative address of the memory field on the specified card. Memory fields are addressed as 16 bit entities with the first field on a card being numbered as zero, the second field being numbered as one, etc.

The CONTROLLER # is only used when testing an output card field. If the value is zero then controller one will be used by default. If the value is non-zero, it must specify a valid controller number (1, 2, 3, or 4) and the specified controller must exist.

The *Data Length* field must contain the length (in bytes) of the buffer that has been allocated to receive the data. If state memory errors are detected, data is returned in the buffer with the following format:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Error Count - MSB | | | | | | | |
| 1 | ... | | | | | | | |
| 2 | ... | | | | | | | |
| 3 | Error Count - LSB | | | | | | | |
| 4 | 1st Error Descriptor | | | | | | | |
| 20 | 2nd Error Descriptor | | | | | | | |
| ... | ... | | | | | | | |
| ... | Nth Error Descriptor | | | | | | | |

The ERROR COUNT field specifies the number of memory errors, if any, that were detected. For each detected error, a memory descriptor with the following format describes the error:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | 00h | | | | | | | |
| 1 | 00h | | | | | | | |
| 2 | 00h | | | | | | | |
| 3 | Diagnostic Type | | | | | | | |
| 4 | 00h | | | | | | | |
| 5 | 00h | | | | | | | |

| | |
|---|---|
| 6 | Memory Address- MSB |
| 7 | Memory Address- LSB |
| 8 | 00h |
| 9 | 00h |
| 10 | Data Written - MSB |
| 11 | Data Written - LSB |
| 12 | 00h |
| 13 | 00h |
| 14 | Data Read - MSB |
| 15 | Data Read - LSB |

A *Diagnostic Type* of zero indicated a memory data test failure, one indicates a memory address test failure. If the number of errors detected exceeds the buffer capacity of the PP, or the size of data buffer as specified in the command packet, the test is terminated and those errors detected prior to termination are reported.

### 1.1.2 SCSI Messages

The PP implements the following SCSI messages:

#### 1.1.2.1 COMMAND COMPLETE

This is a mandatory SCSI-2 message. The PP sends this message to indicate normal completion of a command.

#### 1.1.2.2 DISCONNECT

This is an optional SCSI-2 message. The PP sends this message to the initiator (the host computer) to indicate that is about to break the connect by going to the bus free phase.

#### 1.1.2.3 IDENTIFY

This is a mandatory SCSI-2 message. The PP expects an IDENTIFY message as the first message received after being selected by the initiator (the host computer). If no such message is forthcoming, or some other message is sent first, the PP will terminate the command with an unexpected disconnect by going immediately to the bus free phase. The format of the IDENTIFY message received by the PP must be as follows:

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Identify<br>1h | DiscP<br>1h | Luntar<br>0h | Resrvd<br>0h | Resrvd<br>0h | Logical Unit # | | |

If the format of a received message is not as shown, the command will be terminated with a status of CHECK CONDITION and a sense key of ILLEGAL REQUEST.

When performing a reconnect, the PP will always send an IDENTIFY message immediately after successfully reselecting the initiator. The format of the message sent is the same as above.

#### 1.1.2.4 MESSAGE REJECT

This is a mandatory SCSI-2 message. The PP will send this message to the initiator (the host computer) if it receives a message it does not recognize; i.e., a message other than IDENTIFY, NO OPERATION, or SYNCHRONOUS DATA TRANSFER REQUEST.

The PP may also receive this message from the host. However, if the PP receives this message in response to any message *other than* SYNCHRONOUS DATA TRANSFER REQUEST, it constitutes a gross protocol error and the PP should respond with an unexpected disconnect by going immediately to the bus free phase.

#### 1.1.2.5 NO OPERATION

This is a mandatory SCSI-2 message. It is sent by the initiator (the host computer) if the PP requests a message when no valid message is available. The PP ignores this message.

#### 1.1.2.6 RESTORE POINTERS

This is an optional SCSI-2 message. The PP sends this message to the initiator (the host computer) to request the I/O process context be restored from its previously saved state. This can be used after a reconnect or for error recovery.

#### 1.1.2.7 SAVE DATA POINTERS

This is an optional SCSI-2 message. The PP sends this message to the initiator (the host computer) to request the I/O process context be saved. This is done prior to a disconnect operation.

#### 1.1.2.8 SYNCHRONOUS DATA TRANSFER REQUEST (SDTR)

This is an optional SCSI-2 message. This message can be both sent and received by the PP and it used to negotiate parameters for synchronous data transfer. If the PP receives an SDTR message from the host it will respond by sending an SDTR message to the host to indicate the maximum synchronous data transfer parameters it can accept. The format of the SDTR message the PP sends to the HOST is as follows:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Extended message 01h | | | | | | | |
| 1 | Extended message length 03h | | | | | | | |
| 2 | SDTR code 01h | | | | | | | |
| 3 | Transfer period factor (100 nS) 19H | | | | | | | |
| 4 | REQ/ACK offset 08H | | | | | | | |

If the *Transfer period factor* in the received packet is greater than 19h, the PP will utilize the greater value during synchronous transfers. Likewise, if the *REQ/ACK offset* in the received packet is less than 08h, the PP will utilize the lesser value during synchronous transfers.

After a hardware reset the PP will revert to asynchronous transfers until such time as a another SDTR negotiation is initiated by the host.

### 1.2 The PP Status Register (PP_STATUS)

The PP control software can pass PP status information to the DAP via the PP_STATUS register. Writing the PP_STATUS register also sets the PP_STS_BSY bit in the STFLAGS2 register. This bit remains set until the DAP acknowledges receipt of the status information. The eight bit status value is broken into the following fields:

| Bit / Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | INIT | SOURCE | | | TYPE | | | |

The three bit SOURCE fields specifies the source of the status as follows:

| Value | Source |
|---|---|
| 0 | Non - error statuses |
| 1 | Controller #1 error |
| 2 | Controller #2 error |
| 3 | Controller #3 error |
| 4 | Controller #4 error |
| 5 | DAP Fault |
| 6 | External Fault |
| 7 | Miscellaneous Fault |

The possible TYPE field values for non-error statuses are as follows:

| Status | Value | Meaning |
|---|---|---|
| RUNNING | 0h | The PP is running. |
| HALTED | 1h | The PP has halted normally. |
| STOPPED | 2h | The PP has halted due to a user request. |
| ABORTED | 3h | The PP has halted due to a user requested experiment abort. |
| CONDITIONAL STOP | 4h | The PP has halted because the condition specified by the conditional action mask was met. |

The possible TYPE field vales for controller errors are as follows:

| Status | Value | Meaning |
|---|---|---|
| LOCAL_HALT_ERROR | 0h | Controller completed a state with a set *halt* bit in the *control* field. This is only an error on controllers *other than* controller #1. |

| | | |
|---|---|---|
| STOP_ERROR | 1h | Controller received a STOP command from the CPU interface. This is only an error on controllers *other than* controller #1. |
| FIFO_EMPTY_ERROR | 2h | Controller FIFO was empty at the end of a state. |
| DAP_ERROR | 3h | Controller completed a state had a set *DAP Notify* bit and a set *DAP Nowait* bit in the *control* field without a response from the DAP. This should only occur on controller #1. |
| RAM_ERROR | 4h | Controller attempted to fetch a subprogram state from a location in state memory that was not allocated as subprogram RAM. |
| CONDITIONAL STOP ERROR | 5h | Controller halt because the condition specified by the conditional action mask was met. This is only an error on controllers *other than* controller #1. |

In the case of a DAP fault the TYPE field value is the number of the DAP fault (1-7). In the case of an external fault the TYPE field value is the number of the lowest order unmasked external fault bit that was found set (1-13).

In the case of miscellaneous fault, the TYPE field value is one of the following:

| | | |
|---|---|---|
| UNKNOWN EXTERNAL FAULT | 0h | Although the controllers were halted by an external fault, there is presently no unmasked fault. This could occur if the host changed the fault mask before status processing was complete. |
| UNKNOWN EXTERNAL HALT | 1h | The cause of an external halt could not be determined. This should never happen. |

The single bit INIT field is used to request initialization of the DAP. If the DAP reads the PP_STATUS register and determines that the INIT flag is set, it clears the DAP_FLTn bits in the DAPOUT register. This will remove any DAP fault conditions that might otherwise prevent the PP controllers from running. The SCSI command INITIALIZE PP writes a status of HALTED with INIT set to the PP_STATUS register. It waits for the DAP to respond and then writes a new status of HALTED without the FLAG bit set.

### 1.3 Controller Interface

Each of the four controllers is accessible to the PP CPU via the assembly register, the CPU command interface, and the STFLAGS1 register.

The assembly register fields in a controller are write accessible as memory mapped addresses starting at the assembly register base address for that controller (see the PP Memory Map document, section II-L). Reading these same addresses will return the field contents of the currently addressed controller state memory word for diagnostic purposes.

The controller CPU interface is also a set of memory mapped addresses. Each controller card has a set of memory mapped addresses starting at the controller card interface base address for that card (see the PP Memory Map document, section II-L). Specific addresses are assigned to each CPU accessible register on a controller card and the

contents of these registers can be read/written via these addresses. There are also addresses which can be written to perform certain controller commands, such as starting or stopping the controller. In these cases the actual data written is irrelevant, the action of executing the write at that address executed the command. See the PP Controller Card document, section 3.3 for register assignment details.

Each controller also has three status output signals, BUSY, IRQ, and FFULL, which are visible from the STFLAGS1 register (see the PP Memory Map document, section II-B). The BUSY output indicates that a controller has not yet completed the last requested state memory write operation. The IRQ output indicates that the controller is requesting an interrupt because of a change of status. And the FFULL output indicates that the state memory FIFO associated with that controller is currently full.

### 1.4 Output Card Interface

Each of the 16 possible output cards has an assembly register and a controller select register which are accessible to the PP CPU.

The assembly register on an output card has 16 possible fields (each 16 bits wide) which are write accessible as memory mapped addresses starting at the assembly register base address for that output card slot (see the PP Memory Map document, section II-L). Reading these same addresses will return the field contents of the currently addressed controller state memory word for diagnostic purposes.

Each output card also has a controller select register the contents of which determine which controller card the output module is currently associated with. The controller select register for each output card slot is accessible by the PP CPU at a memory mapped address (see the PP Memory Map document, section II-M).

### 1.5 Fault Interface

The 16 bit FAULT register indicates the status of external devices whose operational readiness is required to perform an experiment. Three of these bits show the status of the DAP, the remaining 13 are connected to uncommitted external fault inputs. These bits are all latched, once a bit has been set by an external condition it will remain set until it is cleared by the action of writing a one to the FAULT_RESET bit of the FLGACK register. Each FAULT register bit has a corresponding mask bit in the FLT_MSK register. If this mask bit is clear, then the corresponding FAULT register bit is “unmasked” and will cause the EXT_FLT signal to be asserted any time the FAULT bit is set. The EXT_FLT signal will force all controllers to a halted condition. The state of the EXT_FLT signal is visible from the STFLAGS2 register.

## 2. Program Structure

The program will have two primary components: a main event loop and a SCSI co-processor interrupt routine.

The main loop will perform the following actions:

- Check the CTRn_IRQ bits in the STFLAGS1 register to determine if any of the four controllers have changed status. If a status change has occurred a new composite status should be written to the PP_STATUS register, the status buffer should be rebuilt (used as the response for GET NEXT STATUS commands) and the SCSI interrupt routine should be notified.
- Check the MAIN_LOOP_SEVICE_REQUIRED flag to determine if any of the SCSI unit control blocks (UCBs) are in a state that requires service from the main loop.

If it is set the UCB list should be searched and a service routine called for each UCB requiring service. Notify the SCSI interrupt routine when service is completed.

- Check the load FIFO buffer for each controller. If the buffer for a controller is not empty then CTRn_BUSY and CTRn_FFULL bits for that controller should be checked to ascertain that the controller has completed the last state write and that the controller state FIFO is not full. If this condition is true then load the assembly register and initiate the state write.
- Loop to beginning.

The SCSI interrupt routine will be very similar to that in the DAP software. The interrupt routine will be invoked by the co-processor at pre-determined points in the co-processor script. At each interrupt the interrupt routine will determine what action is required next and restart the SCSI script at the appropriate point. If no further action is required to complete the current command, the interrupt routine will search the UCBs to determine if any of the units are awaiting service by the co-processor. If no service is required the script will be restarted at the point where it waits to be selected by the initiator. If the main loop alters the state of any UCB it will notify the co-processor.