Transition DDMAQI2:2: When all data requested for the command has been transmitted or an error has been encountered that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQI3: Send_status state.

DDMAQI3: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA/ATAPI-6 standard and the I bit set to one.

Transition DDMAQI3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

DDMAQI4: Release: This state is entered when the device does not have the requested data available for immediate transfer.

When in this state, the device shall request that the Transport layer transmit a Register FIS with the REL bit set to one, with register content as described in the command description in the ATA/ATAPI-6 standard, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DDMAQI4:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

11.13 WRITE DMA QUEUED command protocol

This class includes:

- WRITE DMA QUEUED
- WRITE DMA QUEUED EXT

Execution of this class of command includes the transfer of one or more blocks of data from the device to the host using DMA transfer. All data for the command may be transferred without a bus release between the command receipt and the data transfer. This command may bus release before transferring data. The host shall initialize the DMA controller prior to transferring data. When data transfer is begun, all data for the request shall be transferred without a bus release.

DDMAQO0: DMA-queued_out	Determine whether to transfer or release.	
1. Ready to accept Data FIS	→	DDMAQO1: Send_DMA_activate
2. Command due to error.	→	DDMAQO4: Send_status
3. Bus release	→	DDMAQO5: Release

DDMAQO1: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.	
1. DMA Activate FIS transmitted.	→	DDMAQO2: Receive_data

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

DDMAQO2: Receive_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DDMAQO3: Prepare_data_buffer

DDMAQO3: Prepare_data_buffer	Prepare to receive the next Data FIS.	
1. Ready to receive.	→	DDMAQO1: Send DMA_activate
2. Command complete or aborted due to error.	→	DDMAQO4: Send_status

DDMAQO4: Send_status	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQO5: Release	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQO0: DMA_queued_out State: This state is entered when the device receives a WRITE DMA QUEUED command.

When in this state, device shall determine if it is ready to accept the requested data from the host.

Transition DDMAQO0: 1: When the device is ready to receive a Data FIS immediately, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO0:2: When the device has encountered an error that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQO4: Send_status state.

Transition DDMAQO0:3: When the device does not have the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQO5: Release state.

DDMAQO1:Send_DMA_activate: This state is entered when the device is ready to receive a Data FIS from the host.

When in this state, the device shall request that the Transport layer transmit a DMA Activate FIS.

Transition DDMAQO1:1: When the DMA Activate FIS has been transferred, the device shall transition to the DDMAQO2: Receive_data state.

DDMAQO2:Receive_data: This state is entered when the device transmitted a DMA Activate FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DDMAQO2:1: When the Data FIS has been received, the device shall transition to the DDMAQO3: Prepare_data_buffer state.

DDMAQO3: Prepare_data_buffer: This state is entered when the device has completed receiving a Data FIS from the host.

When in this state, the device shall prepare for receipt of the next Data FIS.

Transition DDMAQO3:1: When ready to receive the Data FIS, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO3:2: When all data requested for the command has been transmitted or an error has been encountered that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQO4: Send_status state.

DDMAQO4: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA/ATAPI-6 standard and the I bit set to one.

Transition DDMAQO4:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

DDMAQO5: Release: This state is entered when the device cannot receive the requested data immediately.

When in this state, the device shall request that the Transport layer transmit a Register FIS with REL set to one, with register content as described in the command description in the ATA/ATAPI-6 standard, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DDMAQO5:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

11.14 FPDMA QUEUED command protocol

This class includes:

- READ FPDMA QUEUED
- WRITE FPDMA QUEUED

DFPDMAQ1: AddCommandToQueue	Append command to internal device command queue and store TAG value.	
1. Device successfully en-queued the command	→	DFPDMAQ2: ClearInterfaceBsy
2. Command malformed	→	DFPDMAQ12: BrokenHost_ ClearBusy

DFPDMAQ2: ClearInterfaceBsy	Transmit Register FIS with BSY=0 and DRQ=0 and interrupt bit I=0 to mark interface ready for the next command.	
1. Register FIS transmission complete	→	DI0: Device_idle

DFPDMAQ3: DataPhaseReadSetup	Transmit a DMA Setup FIS to the host with the DMA Buffer Identifier = TAG and D=1 (direction is device to host) and interrupt bit I=0	
1. First Party DMA Setup FIS transmission complete	→	DFPDMAQ8: DataXmitRead

DFPDMAQ4: DataPhasePreWriteSetup		
1. DMA Setup FIS Auto-Activate option supported and enabled	→	DFPDMAQ5: DataPhase_ WriteSetup
2. DMA Setup FIS Auto-Activate option not supported or not enabled	→	DFPDMAQ6: DataPhase_ OldWriteSetup

DFPDMAQ5: DataPhase_WriteSetup	Transmit a DMA Setup FIS to the host with the DMA Buffer Identifier = TAG and D=0 (direction is host to device) and Auto-Activate = 1 and interrupt bit I=0	
1. DMA Setup FIS transmission complete	→	DFPDMAQ9: DataXmitWrite

DFPDMAQ6: DataPhase_OldWriteSetup	Transmit a DMA Setup FIS to the host with the DMA Buffer Identifier = TAG and D=0 (direction is host to device) and Auto-Activate=0 and interrupt bit I=0	
1. DMA Setup FIS transmission complete	→	DFPDMAQ7: DataPhase_ XmitActivate

DFPDMAQ7: DataPhase_XmitActivate	Transmit a DMA Activate FIS to the host	
1. DMA Activate FIS transmission complete	→	DFPDMAQ9: DataXmitWrite

DFPDMAQ8: DataXmitRead	Transmit Data FIS to the host	
1. Transfer count for previous DMA Setup FIS not exhausted and no error encountered	→	DFPDMAQ8: DataXmitRead
2. Transfer count for previous DMA Setup FIS exhausted and data transfer for this command not complete and no error encountered[1]	→	DI0: Device_idle
3. Finished with data transfer for this command and no error encountered	→	DI0: Device_idle
4. Unrecoverable error has occurred	→	DI0: Device_idle
NOTE: 1. This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands.		

DFPDMAQ9: DataXmitWrite	Receive Data FIS from host	
1. Transfer count for previous DMA Setup FIS not exhausted and no error encountered	→	DFPDMAQ7: DataPhase_ XmitActivate
2. Transfer count for previous DMA Setup FIS exhausted and data transfer for this command not complete and no error encountered[1]	→	DI0: Device_idle
3. Finished with data transfer for this command and no error encountered	→	DI0: Device_idle
4. Unrecoverable error has occurred	→	DI0: Device_idle
NOTE: 1. This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands.		

DFPDMAQ10: SendStatus	Transmit Set Device Bits FIS with ERR = 0, interrupt bit I = 1, and bit ***n*** in SActive field set to one where ***n*** = TAG for each command TAG value that has completed since the last status return	
1. Set Device Bits FIS transmission complete	→	DI0: Device_idle

DFPDMAQ11: ERROR	Halt command processing and transmit Set Device Bits FIS to host with ERR bit in Status field set, interrupt bit I = 1, ATA error code set in Error field, and bits in SActive field not set for any outstanding queued commands and bits set for any successfully completed queued commands for which completion notification not yet delivered.	
1. Set Device Bits FIS transmission complete	→	DFPDMAQ13: WaitforClear

DFPDMAQ12: BrokenHost_ClearBusy	Halt command processing and transmit Register FIS to host with ERR bit in Status field set, interrupt bit I = 1, BSY = 0, DRQ=0, and Error field = 04h	
1. Register FIS transmission complete	→	DFPDMAQ13: WaitforClear

DFPDMAQ13: WaitforClear	Wait for host to either issue READ LOG EXT command with log page=10h or issue SRST	
1. READ LOG EXT command with log page=10h received	→	DFPDMAQ14: SendQueue_ CleanACK
2. SRST received	→	DSR0: Software_reset_ asserted
3. Any command other than READ LOG EXT with log page=10h received	→	DFPDMAQ12: BrokenHost_ ClearBusy

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

DFPDMAQ14: SendQueue_CleanACK	Discard all commands in the pending device queue. Transmit Set Device Bits FIS with ERR in Status field = 0, ERROR=0, SActive field = FFFFFFFFh, and interrupt bit I=0.		
1. Set Device Bits FIS transmission complete		→	DPIOI0: PIO_in

DFPDMAQ1: AddCommandToQueue: This state is entered when the device has checked the command and determined it to be a native queued type command, and Native Command Queuing is supported and enabled.

When in this state, the device shall check the TAG validity and verify that it is not already assigned to an outstanding command. If valid, the device shall append the command to its internal command queue and store the new TAG value.

Transition DFPDMAQ1:1: When the device determines the TAG is valid, and has added the command to its internal command queue, the device shall transition to the DFPDMAQ2: ClearInterfaceBusy state.

Transition DFPDMAQ1:2: When the device determines that the received command is malformed, an error has occurred and the device shall transition to the DFPDMAQ12: BrokenHost_ClearBusy state. A command may be considered malformed as a result of any of its parameters being invalid, including the use of a TAG value that corresponds to an existing TAG value for a pending command, or as a result of the received command being a non-queued command while queued commands are outstanding.

DFPDMAQ2: ClearInterfaceBusy: This state is entered when the device has appended the command to its internal queue and is ready transmit a Register FIS with BSY=0 and DRQ=0 to indicate that the interface is ready to receive the next command

Transition DFPDMAQ2:1: When the Register FIS has been transmitted, the device shall transition to the DI: Device_idle state.

DFPDMAQ3: DataPhaseReadSetup: This state is entered when the device has determined that it is ready to transmit data for a previously queued READ FPDMA QUEUED command.

When in this state, the device shall transmit a DMA Setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the direction set to D=1 (host memory write).

Transition DFPDMAQ3:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ8: DataTransmitRead state.

DFPDMAQ4: DataPhasePreWriteSetup: This state is entered when the device has determined that it is ready to receive data for a previously queued WRITE FPDMA QUEUED command.

When in this state, the device shall determine if the DMA Setup Auto-Activate option is supported and enabled, and then make the appropriate state transition.

Transition DFPDMAQ4:1: If the DMA Setup FIS Auto-Activate option is enabled, the device shall transition to the DFPDMAQ5: DataPhase_WriteSetup state.

Transition DFPDMAQ4:2: If the DMA Setup FIS Auto-Activate option is not supported or not enabled, the device shall transition to the DFPDMAQ6: DataPhase_OldWriteSetup state.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 405 of 587

DFPDMAQ5: DataPhase_WriteSetup: This state is entered when the device is ready to Auto Activate and receive data for a previously queued WRITE FPDMA QUEUED command.

When in this state, the device transmits a DMA Setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the direction set to D=0 (host memory read), and Auto-Activate=1.

Transition DFPDMAQ5:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ9: DataXmitWrite state.

DFPDMAQ6: DataPhase_OldWriteSetup: This state is entered when the device is ready to receive data for a previously queued WRITE FPDMA QUEUED command, and the device does not support Auto-Activate, or it is not enabled.

When in this state, the device transmits a DMA setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the direction set to D=0 (host memory read), and Auto-Activate=0.

Transition DFPDMAQ6:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ7: DataPhase_XmitActivate state.

DFPDMAQ7: DataPhaseXmit_Activate: This state is entered after the device has completed transmission of a DMA Setup FIS for a WRITE FPDMA QUEUED command or the device has finished receiving a Data FIS for a WRITE FPDMA QUEUED command, and the transfer count is not exhausted.

When in this state, the device transmits a DMA Activate FIS to the host indicating readiness to receive Data FISes from the host.

Transition DFPDMAQ7:1: When the device completes the transmission of the DMA Activate FIS, the device shall transition to the DFPDMAQ9: DataXmitWrite state.

DFPDMAQ8: DataXmitRead: This state is entered after the device has completed transmission of a DMA Setup FIS for a READ FPDMA QUEUED command.

When in this state, the device transmits a Data FIS to the host.

Transition DFPDMAQ8:1: If the transfer count for the previous DMA Setup FIS is not exhausted and no error is encountered, the device remains in the DFPDMAQ8: DataXmitRead state.

Transition DFPDMAQ8:2: If the transfer count for the previous DMA Setup FIS is exhausted, and the data transfer for this command is not complete, and no error is encountered, the device shall transition to the DI0: Device_idle state.

This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands

Transition DFPDMAQ8:3: If the device has completed the data transfer for this command, and no error is encountered, the device shall transition to the DI0: Device_idle state.

Transition DFPDMAQ8:4: If the device determines that an unrecoverable error has occurred, the device shall transition to the DI0: Device_idle state.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 406 of 587

DFPDMAQ9: DataXmitWrite: This state is entered after the device has completed transmission of a DMA Setup FIS for a WRITE FPDMA QUEUED command.

When in this state, the device receives a Data FIS from the host.

Transition DFPDMAQ9:1: After the data FIS reception is complete, and if the transfer count for the previous DMA Setup FIS is not exhausted and no error is encountered, the device shall transition to the DFPDMAQ7: DataPhase_XmitActivate state.

Transition DFPDMAQ9:2: If the transfer count for the previous DMA Setup FIS is exhausted, and the data transfer for this command is not complete, and no error is encountered, the device shall transition to the DI0: Device_idle state.

This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands

Transition DFPDMAQ9:3: If the device has completed the data transfer for this command, and no error is encountered, the device shall transition to the DI0: Device_idle state.

Transition DFPDMAQ9:4: If the device determines that an unrecoverable error has occurred, the device shall transition to the DI0: Device_idle state.

DFPDMAQ10: SendStatus: This state is entered when the data transfer for this command, or aggregated commands, is completed and the device is ready to send status.

When in this state, the device transmits a Set Device Bits FIS to the host with ERR = 0, interrupt bit I = 1, and bit ***n*** in SActive field set to one where ***n*** = TAG for each command TAG value that has completed since the last status return.

Transition DFPDMAQ10:1: When the device completes the transmission of the Set Device Bits FIS, the device shall transition to the DI0: Device_idle state.

DFPDMAQ11: ERROR: This state is entered when the device has encountered an unrecoverable error.

When in this state, the device halts command processing and transmits a Set Device Bits FIS to the host with ERR = 1, interrupt bit I = 1, ATA error code set in the Error field, and bits in SActive field not set for any outstanding queued commands (including the erring command) and bits set for any successfully completed queued command for which a completion notification has not yet been provided to the host.

Transition DFPDMAQ11:1: When the device completes the transmission of the Set Device Bits FIS, the device shall transition to the DFPDMAQ13: WaitforClear state.

DFPDMAQ12: BrokenHost_ClearBusy: This state is entered when the device has received a READ FPDMA QUEUED or WRITE FPDMA QUEUED command with a TAG that already exists in its command queue, or when the received command is a not a READ FPDMA QUEUED; and not a WRITE FPDMA QUEUED; and not a DEVICE RESET; and there are native queued command(s) outstanding.

When in this state, the device halts command processing and transmits a Register FIS to the host with ERR = 1 in the status field, interrupt bit I = 1, BSY = 0, DRQ = 0, and ATA error code set in the Error field.

Transition DFPDMAQ12:1: When the device completes the transmission of the Register FIS, the device shall transition to the DFPDMAQ13: WaitforClear state.

DFPDMAQ13: WaitforClear: This state is entered when the device has transmitted an error FIS to the host and is awaiting a READ LOG EXT command with log page = 10h or a soft reset. Any other commands return Register – Device to Host FIS with the ERR bit set to one in the Status field.
Transition DFPDMAQ13:1: If the device receives a READ LOG EXT command with log page = 10h, the device shall transition to the DFPDMAQ14: SendQueue_CleanACK state.

Transition DFPDMAQ13:2: If the device receives a SRST, the device shall transition to the DSR0: Software_reset_asserted state.

Transition DFPDMAQ13:3: If the device receives command other than READ LOG EXT command with log page = 10h, the device shall transition to the DFPDMAQ12: BrokenHost_ClearBusy state.

DFPDMAQ14: SendQueue_CleanACK: This state is entered when the host has responded to an error FIS with a READ LOG EXT command with log page = 10h.

The device shall discard all commands in the pending queue and transmit a Set Device Bits FIS with ERR in the status field = 0, ERROR = 0, SActive field = FFFFFFFFh, and Interrupt bit I = 0.

Transition DFPDMAQ14:1: when the Set Device Bits FIS transmission is complete, the device shall transition to the DPIOI0: PIO_in state.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 408 of 587

12 Host command layer protocol

12.1 FPDMA QUEUED command protocol

This high-level state machine describes the behavior of the host for the Native Command Queuing command protocol. The host behavior described by the state machine may be provided by host software and/or host hardware and the intent of the state machines is not to indicate any particular implementation.

This class includes:

- READ FPDMA QUEUED
- WRITE FPDMA QUEUED

HFPI0: Idle		
1. Free TAG location and command waiting to have TAG assigned	→	HFPDMAQ2: PresetACTBit
2. Command with assigned TAG awaiting issue and BSY=0 and not First-party DMA Data Phase	→	HFPDMAQ3: IssueCommand
3. Interrupt received from device.	→	HFPDMAQ4: DeviceINT
4. Default	→	HFPI0: Idle
NOTE: 1. If more than one condition is true, the host may apply a vendor specific priority		

HFPDMAQ1: AddCommandToQueue	Append command to internal host command queue	
1. Unconditional	→	HFPI0: Idle

HFPDMAQ2: PresetACTBit	Assign free TAG value to command. Write SActive register with value that has bit set in bit position corresponding to assigned TAG value	
1. TAG value assigned to command and SActive register written with new TAG bitmask	→	HFPI0: Idle

HFPDMAQ3: IssueCommand	If not First-party DMA Data Phase, transmit Register FIS to device with new command and assigned TAG value	
1. Register FIS transmission complete (command issued)	→	HFPI0: Idle
2. Register FIS transmission deferred (command not issued)	→	HFPI0: Idle

HFPDMAQ4: DeviceINT	Read Status register to clear pending interrupt flag and save value as SavedStatus	
1. Unconditional	→	HFPDMAQ5: CompleteRequests1

HFPDMAQ5: CompleteRequests1	Compare SActive register with stored SActive register from last interrupt to identify completed commands	
1. SActive comparison indicates 1 or more commands are completed	→	HFPDMAQ6: CompleteRequests2
2. SActive comparison indicates no commands are completed	→	HFPDMAQ7: CompleteRequests3

HFPDMAQ6: CompleteRequests2	Retire host requests associated with TAG values corresponding to newly cleared bits in the SActive register and update stored SActive with new value	
1. Unconditional	→	HFPDMAQ7: CompleteRequests3

HFPDMAQ7: CompleteRequests3	Test ERR bit in SavedStatus value	
1. ERR = 0	→	HFPI0: Idle
2. ERR = 1	→	HFPDMAQ8: ResetQueue

HFPDMAQ8: ResetQueue	Issue a READ LOG EXT command with log page = 10h in a Register FIS to device	
1. Unconditional	→	HFPDMAQ9: CleanupACK

HFPDMAQ9: CleanupACK	Wait for DRQ=1 and BSY=0 [1]	
1. DRQ=1 and BSY=0	→	HFPDMAQ10: RetrieveRequest_ Sense
2. DRQ=0 or BSY=1	→	HFPDMAQ9: CleanupACK
NOTE: 1. The host may wait for this condition using any means including awaiting an interrupt and checking the DRQ and BSY status, spinning, or periodic timer.		

HFPDMAQ10: RetrieveRequest_Sense	Receive PIO Data FIS with request sense information	
1. PIO Data FIS reception complete	→	HFPDMAQ11: ErrorFlush

HFPDMAQ11: ErrorFlush	Retire failed queued command with status set to error condition reported by device. Flush all allocated native queued command tags. Flush pending native queued commands from host command queue with system-specific error condition or re-issue pending queued commands.	
1. Unconditional	→	HFPI0: Idle

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

HFPI0: Idle: When in this state, if queuing is supported and enabled, the Command Layer is awaiting a READ FPDMA QUEUED or WRITE FPDMA QUEUED command from the higher level protocol, or awaiting an interrupt from the Device indicating completion of previously queued commands, or waiting for a TAG location to become available for a command waiting in the command queue.

Transition HFPI0:1: If a DMA Read or Write command is pending that has not had a TAG value assigned to it and there is a free TAG location available for assignment then a transition shall be made to the HFPDMAQ2: PresetACTBit state.

Transition HFPI0:2: If a command with assigned TAG value is awaiting issue to the device and BSY=0 and the interface is not in the First-party DMA Data Phase, then a transition shall be made to the HFPDMAQ3: IssueCommand state.

Transition HFPI0:3: If an interrupt is received from the device, indicating status is available for a previously queued command, it shall transition to the HFPDMAQ4: DeviceINT state.

Transition HFPI0:4: If the queuing is supported and enabled, and the command Layer is awaiting a free TAG, a new command, or an interrupt for a previously queued command, it shall transition to the HFPI0: Idle state.

HFPDMAQ1: AddCommandToQueue: The Command Layer enters this state when it has received a READ FPDMA QUEUED or WRITE FPDMA QUEUED command from the higher level protocol, and adds it to the internal host command queue.

Transition HFPDMAQ1:1: After the Command Layer has added the command to the internal host command queue, it shall transition to the HFPI0: Idle state.

HFPDMAQ2: PresetACTBit: When in this state, the Command Layer assigns a free TAG value to the previously queued command and writes the SActive register with the bit position corresponding to the assigned TAG value.

Transition HFPDMAQ2:1: After the Command Layer has assigned a TAG value and written the corresponding bit to the SActive register, it shall transition to the HFPI0: Idle state.

HFPDMAQ3: IssueCommand: When in this state, the Command Layer attempts to issue a command with preassigned TAG to the device by transmitting a Register FIS to the device with the new command and assigned TAG value if the interface state permits it.

Transition HFPDMAQ3:1: After the Command Layer has transmitted the Register FIS, it shall mark the corresponding command as issued and transition to the HFPI0: Idle state.

Transition HFPDMAQ3:2: After the Command Layer has deferred transmission of the Register FIS due to the interface state not permitting it to be delivered, it shall transition to the HFPI0: Idle state. The corresponding command is still considered as not having been issued.

HFPDMAQ4: DeviceINT: When in this state, the Command Layer reads the Device Status Register to reset the pending interrupt flag and save the value as SavedStatus.

Transition HFPDMAQ4:1: After the Command Layer has read the Device Status Register, it shall transition to the HFPDMAQ5: CompleteRequests1 state.

HFPDMAQ5: CompleteRequests1: When in this state, the Command Layer compares the SActive Register with the SavedStatus SActive Register value that resulted from the last interrupt to identify completed commands.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 411 of 587

Transition HFPDMAQ5:1: If the SActive comparison indicates one or more commands have completed, it shall transition to the HFPDMAQ6: CompleteRequests2 state.

Transition HFPDMAQ5:2: If the SActive comparison indicates no commands have completed, it shall transition to the HFPDMAQ7: CompleteRequests3 state.

HFPDMAQ6: CompleteRequests2: When in this state, the Command Layer retires commands in its internal host command queue that are associated with TAG values corresponding to newly cleared bits in the SActive Register and updates the stored SActive register with the new value.

Transition HFPDMAQ6:1: After updating the stored SActive value, it shall transition to the HFPDMAQ7: CompleteRequests3 state.

HFPDMAQ7: CompleteRequests3: When in this state, the Application layer tests the ERR bit in the SavedStatus value to determine whether the queue should be maintained or reset.

Transition HFPDMAQ7:1: If the SavedStatus value ERR=0, the queue is maintained and it shall transition to the HFPI0: Idle state.

Transition HFPDMAQ7:2: If the SavedStatus value ERR=1, an error has been reported by the Device and the Command layer shall transition to the HFPDMAQ8: ResetQueue state.

HFPDMAQ8: ResetQueue: When in this state, the Application layer issues a READ LOG EXT command with Log Page = 10h in a Register FIS to the device.

Transition HFPDMAQ8:1: After the command has been issued, it shall transition to the HFPDMAQ9: CleanupACK state.

HFPDMAQ9: CleanupACK: When in this state, the Command Layer tests the Device for DRQ = 1 and BSY = 0 in preparation for a PIO data FIS transfer.

Transition HFPDMAQ9:1: If DRQ is set to one and BSY is cleared to zero, it shall transition to the HFPDMAQ10: ReceiveRequestSense state.

Transition HFPDMAQ9:2: If DRQ is cleared to zero or BSY is set to one, it shall transition to the HFPDMAQ9: CleanupACK state.

HFPDMAQ10: RetrieveRequest_Sense: When in this state, the Command Layer completes the PIO Data FIS that retrieves the Request Sense information.

Transition HFPDMAQ10:1: After the completion of the PIO Data FIS, it shall transition to the HFPDMAQ11: ErrorFlush state.

HFPDMAQ11: ErrorFlush: When in this state, the Command Layer retires the failed queued command with the error status set to the error condition reported by the device. It flushes all allocated native queued command tags, and flushes pending native commands from the host command queue with system-specific error condition or re-issue pending queued commands

Transition HFPDMAQ11:1: After the error flush actions have been completed, it shall transition to the HFPI0: Idle state.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 412 of 587

13 Application layer

13.1 Parallel ATA Emulation

Emulation of parallel ATA device behavior as perceived by the host BIOS or software driver, is a cooperative effort between the device and the Serial ATA host adapter hardware. The behavior of Command and Control Block registers, PIO and DMA data transfers, resets, and interrupts are all emulated.

The host adapter contains a set of registers that shadow the contents of the traditional device registers, referred to as the Shadow Register Block. All Serial ATA devices behave like Device 0 devices. Devices shall ignore the DEV bit in the Device field of received Register – Host to Device FISes, and it is the responsibility of the host adapter to gate transmission of Register – Host to Device FISes to devices, as appropriate, based on the value of the DEV bit.

After a reset or power-on, the host bus adapter emulates the behavior of a traditional ATA system during device discovery. Immediately after reset, the host adapter shall place the value 7Fh in its Shadow Status register and shall place the value FFh in all the other Shadow Command Block registers (FFFFh in the Data register). In this state the host bus adapter shall not accept writes to the Shadow Command Block Registers. When the Phy detects presence of an attached device, the host bus adapter shall place the value FFh or 80h in the Shadow Status register, and the host bus adapter shall now allow writes to the Shadow Command Block Registers. If a device is present, the Phy shall take no longer than 10 ms to indicate that it has detected the presence of a device, and has set bit 7 in the Shadow Status register. Placing the value 80h in the Shadow Status register is recommended as it provides the highest level of BIOS compatibility. Note that when bit 7 (BSY) is set in the Shadow Status register all other bits in that register have indeterminate values. When the attached device establishes communication with the host bus adapter, it shall send a Register – Device to Host FIS to the host, resulting in the Shadow Command Block Registers being updated with values appropriate for the attached device.

BIOS and system software writers should be aware of the 10 ms latency the interface may incur in determining device presence, and either ensure the Shadow Status register is read no sooner than 10 ms after initialization or ensure the Shadow Status register is re-read 10 ms after having read a value of 7Fh in order to positively determine presence of a device.

The host adapter may present a Master-only emulation to host software, that is, each device is a Device 0, and each Device 0 is accessed at a different set of host bus addresses. The host adapter may optionally present a Master/Slave emulation to host software, that is, two devices on two separate Serial ATA ports are represented to host software as a Device 0 and a Device 1 accessed at the same set of host bus addresses.

13.1.1 Software Reset

According to the ATA/ATAPI-6 standard, issuing a software reset is performed by toggling the SRST bit in the Device Control register. The toggle period is stipulated as being no shorter than 5 us. As a result of the SRST bit changing in the Device Control register, Serial ATA host adapters shall issue at least two Register – Host to Device FISes to the device (one with the SRST bit asserted and a subsequent one with the SRST bit cleared). See section 10.3.4 for a detailed definition of Register – Host to Device FIS. Although host software is required to toggle the SRST bit no faster than 5 us, devices may not rely on the inter-arrival time of received Register – Host to Device FISes also meeting this timing. Because of flow control, frame handshaking, and other protocol interlocks, devices may effectively receive the resulting Register – Device to Host FISes back-to-back.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 413 of 587

Due to flow control, protocol interlocks, power management state, or other transmission latencies, the subsequent Register – Host to Device FIS transmission clearing the SRST bit during a software reset may be triggered prior to the previous Register – Host to Device FIS transmission having been completed. Host adapters are required to allow host software to toggle the SRST with the minimum 5 us timing specified in the ATA/ATAPI-6 standard even if frame transmission latencies result in the first Register FIS transmission taking longer than 5 us. Host adapters are required to ensure transmission of the two resulting Register – Host to Device FISes to the device regardless of the transmission latency of each individual FIS.

13.1.2 Master-only emulation

NOTE: Unlike the remainder of this specification, this section is based on the ATA/ATAPI-5 standard.

A native Serial ATA host adapter behaves the same as if a legacy master only device were attached with no slave present. It is the responsibility of the host adapter to properly interact with host software and present the correct behavior for this type of configuration. All Serial ATA devices, therefore, need not be aware of master / slave issues and ignore legacy task file information that deal with a secondary device. When the DEV bit in the Device register is set to one, selecting the non-existent Device 1, the host adapter shall respond to register reads and writes as specified for a Device 0 with no Device 1 present, as defined in the ATA/ATAPI-5 standard. This includes not setting the BSY bit in the shadow Status register when Device 1 is selected, as described in the ATA/ATAPI-5 standard. When Device 0 is selected, the host adapter shall execute the Serial ATA protocols for managing the shadow register contents as defined in later sections.

When Device 0 is selected and the Command register is written in the Shadow Register Block, the host adapter sets the BSY bit in its shadow Status register. The host adapter then transmits a Register – Host to Device FIS to the device containing the new register contents. When the Device Control register is written in the Shadow Register Block with a change of state of the SRST bit, the host adapter sets the BSY bit in its shadow Status register and transmit a Register – Host to Device FIS to the device containing the new register contents. Transmission of register contents when the Device Control register is written with any value that is not a change of state of the SRST bit shall not set the BSY bit in the shadow Status register, and transmission of a frame to the device containing new register contents is optional. Similarly, the host adapter sets the BSY bit in its shadow Status register to one when a COMRESET is requested or the SRST bit is set to one in the Device Control register. The expected timing for BSY bit assertion is thereby preserved.

The device updates the contents of the host adapter Shadow Register Block by transmitting a Register – Device to Host FIS. This allows the device to set the proper ending status in the host adapter Shadow Register Block at the completion of a command or control request. Specific support is added to ensure proper timing of the DRQ and BSY bits in the Status register for PIO transfers.

Finally the host adapter cooperates with the device by providing an interrupt pending flag in the host adapter. This flag is set by the host adapter when the device sends a Register – Device to Host FIS with the I (Interrupt) bit set to one. The host adapter asserts the interrupt to the host processor any time the interrupt pending flag is set, the DEV bit is cleared to zero in the shadow Device register, and the nIEN bit in the shadow Device Control register is cleared to zero. The host adapter clears the interrupt pending flag any time a COMRESET is requested, the SRST bit in the shadow Device Control register is set to one, the shadow Command register is written with DEV cleared to zero, or the shadow Status register is read with DEV cleared to zero. This allows the emulation of the host interrupt and its proper timing.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 414 of 587

13.1.3 Master/Slave emulation (optional)

NOTE: Unlike the remainder of this specification, this section is based on the ATA/ATAPI-5 standard.

All devices behave as if they are Device 0 devices. However, the host adapter may optionally implement emulation of the behavior of a Master/Slave configuration by pairing two Serial ATA ports, and managing their associated shadow registers accordingly, as though they were a Device 0 and Device 1 at the same set of host bus addresses.

A host adapter that emulates Master/Slave behavior shall manage the two sets of shadow registers (one set for each of the two devices) based on the value of the DEV bit in the shadow Device register. Based on the value of the DEV bit, the host adapter shall direct accesses to the Shadow Register Block Registers, and accesses to the Control Block Registers, to the appropriate set of shadow registers in the correct device. It is the responsibility of the host adapter to ensure that communication with one or both of the attached devices is handled properly, and that information gets routed to the devices correctly. Each device shall process any communication with the host adapter as if it is targeted for the device regardless of the value of the DEV bit.

If a host adapter is emulating Master/Slave behavior, and there is no device attached to the cable designated as the Device 1 cable, the host adapter shall emulate Device 0 behavior with no Device 1 present as described in the ATA/ATAPI-5 standard.

13.1.3.1 Software reset

Host adapters that emulate Master/Slave behavior shall emulate proper behavior for software reset. Based on the Phy initialization status, the host adapter knows whether a device is attached to each of the two ports used in a Master/Slave emulation configuration. Device Control register writes, that have the SRST bit set to one, shall result in the associated Shadow Device Control register being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register – Host to Device FIS being transmitted to each attached device. Similarly, the subsequent write to the Shadow Device Control register that clears the SRST bit shall result in a Register – Host to Device FIS being sent to each attached device. The host adapter shall then await a response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the ATA/ATAPI-5 standard, to produce the Error and Status register values visible to host software.

13.1.3.2 EXECUTE DEVICE DIAGNOSTICS

Host adapters that emulate Master/Slave behavior shall emulate proper behavior for EXECUTE DEVICE DIAGNOSTICS. The host adapter shall detect the EXECUTE DEVICE DIAGNOSTIC command being written to the Command register. Detecting the EXECUTE DEVICE DIAGNOSTICS command shall result in the associated shadow register being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register – Host to Device FIS being transmitted to each attached device. The host adapter shall then await response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the ATA/ATAPI-5 standard to produce the Error and Status register values visible to host software.

13.1.3.3 Restrictions and limitations

Superset capabilities that are unique to Serial ATA and not supported by parallel ATA may not be supported in Master/Slave emulation mode. Such capabilities include but are not limited to support for First-party DMA, hot plug/unplug, interface power management, and superset Status

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 415 of 587

and Control registers. Master/Slave emulation is recommended only in configurations where software written for parallel ATA is used and the number of attached devices exceeds the number of Shadow Command Block register interfaces that software supports.

13.1.3.4 Shadow Command Block Register Access Restrictions

Host software should take measures to ensure that the access restrictions for the Command Block registers (see ATA/ATAPI-5) are observed when accessing the Shadow Command Block registers in order to avoid indeterminate behavior. Some of these measures include:

a) Prior to writing the Shadow Command Block registers to issue a new command, host software should check the Shadow Status register to verify that both BSY and DRQ are cleared to zero.

b) If DRQ is set to one when software expects it to be cleared to zero (e.g. when ERR is set to one), host software should perform an error recovery action (e.g., issue a software reset to the device) prior to writing the Shadow Command register to issue a new command.

   Note: Issuing a software reset by setting the SRST bit in the Device Control register to one while BSY and/or DRQ are set to one is legal behavior as described in the Serial ATA state machines, and completion of a software reset ensures that the device and interface are in a known state.

The ATA/ATAPI-5 standard defines restrictions for writing to the Command Block registers.

It is prohibited to write the Shadow Command register when BSY or DRQ is set to one except for the DEVICE RESET command. The Serial ATA access restrictions differ from parallel ATA where similar access to the Command register is indeterminate. However, the resultant indeterminate behavior may, in some Serial ATA implementations, not be as benign as in some parallel ATA implementations. This is because in Serial ATA the Shadow Command Block registers are cooperatively managed between the host and the device, and the defined coordination is based on the values of BSY and DRQ.

An example situation where parallel ATA and Serial ATA behavior may differ is when the device sets both DRQ and ERR to one in the (Shadow) Status register in a PIO read operation. In parallel ATA, many device implementations allow software to issue a new command without transferring the data from the device. In Serial ATA, issuing a new command without transferring the data may lead to a hang condition. In Serial ATA, when DRQ is asserted there is an associated Data FIS that is being transferred on the Serial ATA interface. Until the Data FIS transfer is completed by the host reading the associated data from the Shadow Data register, the Data FIS remains on the interface in a $HOLD_P$/ $HOLDA_P$ flow-controlled condition. While the Data FIS remains on the interface, a new command cannot be issued to the device.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 416 of 587

13.1.3.5 Parallel ATA interoperability state diagrams

This state diagram defines the protocol of the host adapter to emulate Master only parallel ATA devices as seen from the host BIOS or software driver. The interrupt pending flag (IPF) is an internal state bit in the host adapter that reflects whether or not the device has an interrupt pending to the host.

HA0: HA_SEL/NOINTRQ	The interrupt signal is not asserted to the host.		
1.	COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2.	Device Control register written by the host with SRST=0 and (nIEN=1 or IPF=0).	→	HA_SEL/NOINTRQ
3.	Device Control register written by the host with SRST=0 and (nIEN=0 and IPF=1).	→	HA_SEL/INTRQ
4.	Device Control register written by the host with SRST=1.	→	HA_SEL/NOINTRQ
5.	Command register written by the host.	→	HA_SEL/NOINTRQ
6.	Device register written by the host with DEV = 1.	→	HA_NOTSEL
7.	Any other register read or write by the host	→	HA_SEL/NOINTRQ
8.	Transport layer indicates new register content from the device with no set IPF flag.	→	HA_SEL/NOINTRQ
9.	Transport layer indicates new register content from the device with set IPF flag and nIEN=1.	→	HA_SEL/NOINTRQ
10.	Transport layer indicates new register content from the device with set IPF flag and nIEN=0.	→	HA_SEL/INTRQ

HA1: HA_SEL/INTRQ	The interrupt signal is asserted to the host.		
1.	COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2.	Device Control register written by the host with SRST=0 and (nIEN=1 or IPF=0).	→	HA_SEL/NOINTRQ
3.	Device Control register written by the host with SRST=0 and (nIEN=0 and IPF=1).	→	HA_SEL/INTRQ
4.	Device Control register written by the host with SRST=1.	→	HA_SEL/NOINTRQ
5.	Command register written by the host.	→	HA_SEL/NOINTRQ
6.	Device register written by the host with DEV = 1.	→	HA_NOTSEL
7.	Status register read by the host	→	HA_SEL/NOINTRQ
8.	Any other register write by the host	→	HA_SEL/INTRQ
9.	Any other register read by the host	→	HA_SEL/INTRQ
10.	Transport layer indicates new register content from the device.	→	HA_SEL/INTRQ

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 417 of 587

HA2: HA_NOTSEL	The interrupt signal is not asserted to the host.	
1. COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2. Device Control register written by the host with SRST = 0.	→	HA_NOTSEL
3. Device Control register written by the host with SRST = 1.	→	HA_SEL/NOINTRQ
4. Command register written by the host with command other than EXECUTE DEVICE DIAGNOSTIC.	→	HA_NOTSEL
5. Command register written by the host with command EXECUTE DEVICE DIAGNOSTIC.	→	HA_SEL/NOINTRQ
6. Device register written by the host with DEV = 0 and (nIEN=1 or IPF=0).	→	HA_SEL/NOINTRQ
7. Device register written by the host with DEV = 0 and (nIEN=0 and IPF=1).	→	HA_SEL/INTRQ
8. Any other register write by the host	→	HA_NOTSEL
9. Read of Status or Alternate Status register by the host.	→	HA_NOTSEL
10. Read of any other register by the host.	→	HA_NOTSEL
11. Transport layer indicates new register content from the device.	→	HA_NOTSEL

HA0: HA_SEL/NOINTRQ state: This state is entered when Device 0 is selected and either IPF is cleared to zero or nIEN is set to one.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA0:1: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:2: When the Device Control register is written by the host with SRST cleared to zero and either nIEN set to one or IPF is cleared to zero, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:3: When the Device Control register is written by the host with SRST cleared to zero, nIEN cleared to zero, and IPF is set to one, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA0:4: When the Device Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:5: When the Command register is written by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a Register – Host to Device FIS with C=1 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:6: When the Device register is written by the host with DEV set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 418 of 587

Transition HA0:7: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:8: When the Transport layer indicates new register content from the device with no set IPF flag, the host adapter shall place the new register content into the shadow registers and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:9: When the Transport layer indicates new register content from the device with set IPF flag and nIEN is set to one, the host adapter shall set IPF to one, place the new register content into the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:10: When the Transport layer indicates new register content from the device with set IPF flag and nIEN is cleared to zero, the host adapter shall set IPF to one, place the new register content into the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

HA1: HA_SEL/INTRQ state: This state is when DEVICE 0 is selected, nIEN is cleared to zero, and IPF is set to one.

When in this state, the interrupt signal to the host shall be asserted.

Transition HA1:1: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:2: When the Device Control register is written by the host with SRST cleared to zero, with nIEN set to one or IPF cleared to zero, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:3: When the Device Control register is written by the host with SRST cleared to zero with nIEN cleared to zero and IPF set to one, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:4: When the Device Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:5: When the Command register is written by the host, the host adapter shall set BSY to one in the shadow Status register, clear IPF to zero, notify the Transport layer to send a Register – Host to Device FIS with C=1 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:6: When the Device register is written by the host with DEV set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA1:7: When the Status register is read by the host, the host adapter shall clear IPF to zero and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:8: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:9: When any register is read by the host other than that described above, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:10: When the Transport layer indicates new register content from the device, the host adapter shall place the new register content into the shadow registers and make a transition to the HA1: HA_SEL/INTRQ state.

HA2: HA_NOTSEL state: This state is entered when Device 1 is selected.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA2:1: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the shadow Status register, clear DEV to zero in the shadow Device register, clear IPF to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: SEL/NOINTRQ state.

Transition HA2:2: When the Device Control register is written by the host with SRST cleared to zero, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:3: When the Device Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the shadow Status register, clear DEV to zero in the shadow Device register, clear IPF to zero, notify the Transport layer to send a Register – Host to Device FIS with C=0 with the current content of the shadow registers, and make a transition to the HA0: HA_ SEL/NOINTRQ state.

Transition HA2:4: When the Command register is written by the host with any command code other than EXECUTE DEVICE DIAGNOSTIC, the host adapter shall not set BSY in the shadow Status register and shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:5: When the Command register is written by the host with the EXECUTE DEVICE DIAGNOSTIC command code, the host adapter shall set BSY to one in the shadow Status register, clear DEV to zero in the Device register, clear IPF to zero, notify the Transport layer to send a Register – Host to Device FIS with C=1 with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/INTRQ state.

Transition HA2:6: When the Device register is written by the host with DEV cleared to zero and either nIEN is set to one or IPF is cleared to zero, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA2:7: When the Device register is written by the host with DEV cleared to zero, nIEN cleared to zero, and IPF is set to one, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA2:8: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:9: When the Status or Alternate Status register is read by the host, the host shall return register content 00h to the host and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:10: When any register is read by the host other than that described above, the host adapter shall return the current shadow register content to the host and make a transition to the HA2: HA_NOTSEL state.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 420 of 587

Transition HA2:11: When the Transport layer indicates new register content from the device, the host adapter shall place the new register content into the shadow registers and make a transition to the HA2: HA_NOTSEL state.

13.2 IDENTIFY (PACKET) DEVICE and SET FEATURES

In the IDENTIFY DEVICE command various parameters are communicated to the host from the device. The following sections define those words that are different from and additions to the ATA/ATAPI-6 standard definition of the data contents. Serial ATA features and capabilities include a means by which their presence and support can be determined, and a means for enabling them if optionally supported.

The IDENTIFY (PACKET) DEVICE settings requirements shall be implemented by native Serial ATA devices. The IDENTIFY (PACKET) DEVICE settings requirements are optional for parallel ATA devices with an external Serial ATA bridge attached.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 421 of 587

13.2.1 IDENTIFY DEVICE

Table 40 – IDENTIFY DEVICE information

Word	O/M	F/V		
0-46			Set as indicated in ATA/ATAPI-6	
47	M		Multiple Count	
		F	15-8	80h
		R	7-0	00h = Reserved 01h-10h = Maximum number of sectors that shall be transferred per interrupt on READ/WRITE MULTIPLE commands 11h-FFh = Reserved
48			Set as indicated in ATA/ATAPI-6	
49	M		Capabilities	
		R	15-14	Reserved for the IDENTIFY PACKET DEVICE command.
		F	13	1=Standby timer values as specified in this standard are supported 0=Standby timer values shall be managed by the device
		R	12	Reserved for the IDENTIFY PACKET DEVICE command.
		F	11	1=IORDY supported 0=IORDY may be supported
		F	10	1=IORDY may be disabled
		R	9	Shall be set to one.
		R	8	Shall be set to one.
		X	7-0	Retired
50-52			Set as indicated in ATA/ATAPI-6	
53	M		Field validity	
		R	15-3	Reserved
		F	2	1=the fields reported in word 88 are valid 0=the fields reported in word 88 are not valid
		F	1	1=the fields reported in words (70:64) are valid 0=the fields reported in words (70:64) are not valid
		F	0	Obsolete
54-62			Set as indicated in ATA/ATAPI-6	
63	M		Multiword DMA transfer	
			15-3	Set as indicated in ATA/ATAPI-6
		F	2	1= Multiword DMA mode 2 and below are supported
		F	1	1= Multiword DMA mode 1 and below are supported
		F	0	1= Multiword DMA mode 0 is supported
64	M		PIO transfer modes supported	
			15-2	Set as indicated in ATA/ATAPI-6
		F	1-0	PIO modes 3 and 4 supported
65	M		Minimum Multiword DMA transfer cycle time per word	
		F	15-0	Cycle time in nanoseconds
66	M		Manufacturer's recommended Multiword DMA transfer cycle time	
		F	15-0	Cycle time in nanoseconds
67	M		Minimum PIO transfer cycle time without flow control	
		F	15-0	Cycle time in nanoseconds
68	M		Minimum PIO transfer cycle time with IORDY flow control	
		F	15-0	Cycle time in nanoseconds
69-74			Set as indicated in ATA/ATAPI-6	
75	O		Queue depth	
		F	15-5	Reserved
		F	4-0	Maximum queue depth - 1
76	O		Serial ATA capabilities	

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 422 of 587

		F	15-11	Reserved
		F	10	Supports Phy event counters
		F	9	Supports receipt of host-initiated interface power management requests
		F	8	Supports Native Command Queuing
		F	7-3	Reserved for future Serial ATA signaling speed grades
		F	2	1 = Supports Serial ATA Gen2 signaling speed (3.0 Gbps)
		F	1	1 = Supports Serial ATA Gen1 signaling speed (1.5 Gbps)
		F	0	Shall be cleared to zero
77			Reserved for future Serial ATA definition	
78	O		Serial ATA features supported	
		F	15-7	Reserved
		F	6	1 = Supports software settings preservation
		F	5	Reserved
		F	4	1 = Supports in-order data delivery
		F	3	1 = Device supports initiating interface power management
		F	2	1 = Supports DMA Setup Auto-Activate optimization
		F	1	1 = Supports non-zero buffer offsets in DMA Setup FIS
		F	0	Shall be cleared to zero
79	O		Serial ATA features enabled	
		F	15-7	Reserved
		V	6	1 = Software settings preservation enabled
		F	5	Reserved
		V	4	1 = In-order data delivery enabled
		V	3	1 = Device initiating interface power management enabled
		V	2	1 = DMA Setup Auto-Activate optimization enabled
		V	1	1 = Non-zero buffer offsets in DMA Setup FIS enabled
		F	0	Shall be cleared to zero
80-87			Set as indicated in ATA/ATAPI-6	
88			15-6	Set as indicated in ATA/ATAPI-6
		F	5	1=Ultra DMA mode 5 and below are supported
		F	4	1=Ultra DMA mode 4 and below are supported
		F	3	1=Ultra DMA mode 3 and below are supported
		F	2	1=Ultra DMA mode 2 and below are supported
		F	1	1=Ultra DMA mode 1 and below are supported
		F	0	1=Ultra DMA mode 0 is supported
89-92			Set as indicated in ATA/ATAPI-6	
93		V	COMRESET result. The contents of this word shall be cleared to zero.	
94-255			Set as indicated in ATA/ATAPI-6	

Key:
M = Support of the word is mandatory.
O = Support of the word is optional.
F = the content of the word is fixed and does not change. For removable media devices, these values may change when media is removed or changed.
V = the contents of the word is variable and may change depending on the state of the device or the commands executed by the device.
X = the content of the word is vendor specific and may be fixed or variable.
R = the content of the word is reserved and shall be zero.

13.2.1.1 Word 0 - 46: Set as indicated in ATA/ATAPI-6

13.2.1.2 Word 47: Multiword PIO transfer

Bits 15 through 8 of word 47 shall be set as indicated in ATA/ATAPI-6.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 423 of 587

Bits 7 through 0 are used to indicate the maximum number of sectors that shall be transferred per interrupt on READ/WRITE MULTIPLE commands. This field shall be set to 16 or less. See section 10.3.10.1.

13.2.1.3 Word 48: Set as indicated in ATA/ATAPI-6

13.2.1.4 Word 49: Capabilities

Bits 15 through 12 of word 49 shall be set as indicated in ATA/ATAPI-6.

Bit 11 of word 49 is used to determine whether a device supports IORDY. This bit shall be set to one, indicating the device supports IORDY operation.

Bit 10 of word 49 is used to indicate a device's ability to enable or disable the use of IORDY. This bit shall be set to one, indicating the device supports the disabling of IORDY. Disabling and enabling of IORDY is accomplished using the SET FEATURES command.

Bits 9 - 0 of word 49 shall be set as indicated in ATA/ATAPI-6.

13.2.1.5 Words 50 - 52: Set as indicated in ATA/ATAPI-6

13.2.1.6 Word 53: Field validity

Bit 0 shall be set to one. Bit 1 of word 53 shall be set to one, the values reported in words 64 through 70 are valid. Any device that supports PIO mode 3 or above, or supports Multiword DMA mode 1 or above, shall set bit 1 of word 53 to one and support the fields contained in words 64 through 70. Bit 2 of word 53 shall be set to one indicating the device supports Ultra DMA and the values reported in word 88 are valid. Bits 15-3 are reserved.

13.2.1.7 Word 54 - 62: Set as indicated in ATA/ATAPI-6

13.2.1.8 Word 63: Multiword DMA transfer

Bits 2 - 0 of word 63 shall be set to one indicating that the device supports Multiword DMA modes 0, 1, and 2. Bits 15 – 3 shall be set as indicated in ATA/ATAPI-6.

13.2.1.9 Word 64: PIO transfer modes supported

Bits 1 - 0 of word 64 shall be set to one indicating that the device supports PIO modes 3 and 4. Bits 15 – 2 shall be set as indicated in ATA/ATAPI-6.

13.2.1.10 Word 65: Minimum Multiword DMA transfer cycle time per word

Shall be set to indicate 120 ns.

13.2.1.11 Word 66: Device recommended Multiword DMA cycle time

Shall be set to indicate 120 ns.

13.2.1.12 Word 67: Minimum PIO transfer cycle time without flow control

Shall be set to indicate 120 ns.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 424 of 587

13.2.1.13 Word 68: Minimum PIO transfer cycle time with IORDY

Shall be set to indicate 120 ns.

13.2.1.14 Words 69-74: Set as indicated in ATA/ATAPI-6

13.2.1.15 Word 75: Queue depth

This word is as defined in the ATA/ATAPI-6 standard. The Native Command Queuing protocol supports at most 32 queued commands, which coincides with the reporting capabilities of the ATA/ATAPI-6 standard. In the native queuing scheme, the host is required to issue only unique tag values for queued commands that have a value less than or equal to the value reflected in this field (i.e. for device reporting a value in this field of 15, corresponding to a maximum of 16 outstanding commands, the host shall never use a tag value greater than 15 when issuing native queued commands).

13.2.1.16 Word 76: Serial ATA capabilities

If not 0000h or FFFFh, the device claims compliance with the Serial ATA specification and supports the signaling rate indicated in bits 1-3. Since Serial ATA supports generational compatibility, multiple bits may be set. Bit 0 is reserved and shall be cleared to zero (thus a Serial ATA device has at least one bit cleared in this field and at least one bit set providing clear differentiation). If this field is not 0000h or FFFFh, words 77 through 79 shall be valid. If this field is 0000h or FFFFh the device does not claim compliance with the Serial ATA specification and Words 76 through 79 are not valid and shall be ignored.

Bit 0 shall be cleared to zero.

Bit 1 when set to one indicates that the device is a Serial ATA device and supports the Gen1 signaling rate of 1.5 Gbps.

Bit 2 when set to one indicates that the device is a Serial ATA device and supports the Gen2 signaling rate of 3.0 Gbps.

Bit 3-7 are reserved for future Serial ATA signaling speed grades and shall be cleared to zero.

Bit 8 when set to one indicates that the Serial ATA device supports the Native Command Queuing scheme defined in section 13.5.

Bit 9 when set to one indicates that the Serial ATA device supports the Partial and Slumber interface power management states when initiated by the host.

Bit 10 when set to one indicates that the Serial ATA device supports Phy event counters. If the device supports Phy event counters, it shall support the Phy event counter READ LOG EXT page 11h.

Bit 11-15 are reserved and shall be cleared to zero

13.2.1.17 Word 77: Reserved

Word 77 is reserved for future Serial ATA definition and shall be cleared to zero.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 425 of 587

13.2.1.18 Word 78: Serial ATA features supported

If Word 76 is not 0000h or FFFFh, Word 78 reports the optional features supported by the device. Support for this word is optional and if not supported the word shall be zero indicating the device has no support for new Serial ATA capabilities.

Bit 0 shall be cleared to zero.

Bit 1 indicates whether the device supports the use of non-zero buffer offsets in the DMA Setup FIS. When set to one, the device supports transmission and reception of DMA Setup FISes with a non-zero value in the Buffer Offset field of the FIS. When cleared to zero, the device supports transmission and reception of the DMA Setup FIS only with the Buffer Offset field cleared to zero.

Bit 2 indicates whether the device supports the use of the DMA Setup FIS Auto-Activate optimization as described in section 10.3.8.3.1. When set to one the device supports use of the Auto-Activate optimization and when cleared to zero the device does not support the Auto-Activate optimization.

Bit 3 indicates whether the device supports initiating power management requests to the host. When set to one the device supports initiating interface power management requests and when cleared to zero the device does not support initiating power management requests. A device may support reception of power management requests initiated by the host as described in the definition of bit 9 of Word 76 without supporting initiating such power management requests as indicated by this bit.

Bit 4 indicates whether the device supports guaranteed in-order data delivery when non-zero buffer offsets are used in the DMA Setup FIS. When set to one, the device guarantees in-order data delivery for READ FPDMA QUEUED or WRITE FPDMA QUEUED commands when non-zero buffer offsets are used with multiple DMA Setup FIS. Target data is delivered in order, starting with the first LBA through command completion. When Bit 4 is cleared to zero, the device does not guarantee in-order data delivery when non-zero buffer offsets are enabled. In this case, data may be interleaved both within a command and across multiple commands. By default this field shall be zero.

Bit 5 is reserved and shall be cleared to zero.

Bit 6 indicates whether the device supports software settings preservation as defined in section 13.4. When set to one the device supports software settings preservation across COMRESET. When cleared to zero the device clears all software settings when a COMRESET occurs.

Bit 7-15 are reserved and shall be cleared to zero

13.2.1.19 Word 79: Serial ATA features enabled

If Word 76 is not 0000h or FFFFh, Word 79 reports which optional features supported by the device are enabled. This word shall be supported if optional Word 78 is supported and shall not be supported if optional Word 78 is not supported.

Bit 0 shall be cleared to zero.

Bit 1 indicates whether device support for use of non-zero buffer offsets in the DMA Setup FIS is enabled. When set to one, device transmission of DMA Setup FISes with a non-zero value in the Buffer Offset field of the FIS is enabled. When cleared to zero, the device is permitted to transmit DMA Setup FIS only with the Buffer Offset field cleared to zero. By default this field shall be zero.

Bit 2 indicates whether device support for use of the DMA Setup FIS Auto-Activate optimization as described in section 10.3.8.3.1 is enabled. When set to one, the device may utilize the Auto-

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 426 of 587

Activate optimization. When cleared to zero the device shall not utilize the Auto-Activate optimization. By default, this field shall be zero.

Bit 3 indicates whether device support for initiating power management requests to the host is enabled. When set to one the device may initiate power management transition requests. When cleared to zero the device shall not initiate interface power management requests to the host. This field shall be zero by default.

Bit 4 indicates whether device support for guaranteed in-order data delivery when non-zero buffer offsets are used in the DMA Setup FIS is enabled. When set to one and non-zero buffer offset is enabled, the device may satisfy a READ FPDMA QUEUED or WRITE FPDMA QUEUED command by transmitting multiple DMA Setup FISes with non-zero buffer offset values where appropriate, provided that the target data is delivered in order, starting with the first LBA through command completion. When Bit 4 is cleared to zero, the device may interleave data both in a command and across multiple commands using non-zero buffer offsets if non-zero buffer offsets are enabled. By default this field shall be zero.

Bit 5 is reserved and shall be cleared to zero.

Bit 6 indicates whether device support for software settings preservation is enabled. When set to one the device shall preserve software settings across COMRESET. When cleared to zero the device shall clear software settings when COMRESET occurs. If the device supports software settings preservation this field shall be one by default. If the device does not support software settings preservation this field shall be zero by default.

Bit 7-15 are reserved and shall be cleared to zero.

13.2.1.20 Words 80-87: Set as indicated in ATA/ATAPI-6

13.2.1.21 Word 88: Ultra DMA modes

Bits 5 - 0 of Word 88 shall be set to one indicating that the device supports Ultra DMA modes 0, 1, 2, 3, 4, and 5. Bits 15 – 5 shall be set as indicated in ATA/ATAPI-6.

13.2.1.22 Word 89 - 92: Set as indicated in ATA/ATAPI-6

13.2.1.23 Word 93: Hardware configuration test results

Word 93 shall be set to 0000h indicating that the word is not supported.

Words 94-255: Set as indicated in ATA/ATAPI-6.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 427 of 587

13.2.2 IDENTIFY PACKET DEVICE

Table 41 – IDENTIFY PACKET DEVICE information

Word	O/M	F/V		
0-48			Set as indicated in ATA/ATAPI-6	
49	M		Capabilities	
		F	15-12	Set as indicated in ATA/ATAPI-6
		F	11	1=IORDY supported 0=IORDY may be supported
		F	10	1=IORDY may be disabled
		R	9	Shall be set to one.
		R	8	1=DMA supported
		X	7-0	Vendor specific
50-52			Set as indicated in ATA/ATAPI-6	
53	M		Field validity	
		R	15-3	Reserved
		F	2	1=the fields reported in word 88 are valid 0=the fields reported in word 88 are not valid
		F	1	1=the fields reported in words (70:64) are valid 0=the fields reported in words (70:64) are not valid
		F	0	Obsolete
54-62			Set as indicated in ATA/ATAPI-6	
63	M		Multiword DMA transfer	
			15-3	Set as indicated in ATA/ATAPI-6
		F	2	1= Multiword DMA mode 2 and below are supported
		F	1	1= Multiword DMA mode 1 and below are supported
		F	0	1= Multiword DMA mode 0 is supported
64	M		PIO transfer modes supported	
			15-2	Set as indicated in ATA/ATAPI-6
		F	1-0	PIO modes 3 and 4 supported
65	M		Minimum Multiword DMA transfer cycle time per word	
		F	15-0	Cycle time in nanoseconds
66	M		Manufacturer's recommended Multiword DMA transfer cycle time	
		F	15-0	Cycle time in nanoseconds
67	M		Minimum PIO transfer cycle time without flow control	
		F	15-0	Cycle time in nanoseconds
68	M		Minimum PIO transfer cycle time with IORDY flow control	
		F	15-0	Cycle time in nanoseconds
69-75			Set as indicated in ATA/ATAPI-6	
76	O		Serial ATA capabilities	
		F	15-11	Reserved
		F	10	Supports Phy event counters
		F	9	Supports receipt of host-initiated interface power management requests
		F	8-4	Reserved
		F	7-3	Reserved for future Serial ATA signaling speed grades
		F	2	1 = Supports Serial ATA Gen2 signaling speed (3.0 Gbps)
		F	1	1 = Supports Serial ATA Gen1 signaling speed (1.5 Gbps)
		F	0	Shall be cleared to zero
77			Reserved for future Serial ATA definition	
78	O		Serial ATA features supported	
		F	15-7	Reserved
		F	6	1 = Supports software settings preservation
		F	5	1 = Supports asynchronous notification
		F	4	Reserved

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 428 of 587

		F	3	1 = Device supports initiating interface power management
		F	2-1	Reserved
		F	0	Shall be cleared to zero
79	O		Serial ATA features enabled	
		F	15-7	Reserved
		V	6	1 = Software settings preservation enabled
		V	5	1 = Asynchronous notification enabled
		F	4	Reserved
		V	3	1 = Device initiating interface power management enabled
		F	2-1	Reserved
		F	0	Shall be cleared to zero
80-87			Set as indicated in ATA/ATAPI-6	
88			15-6	Set as indicated in ATA/ATAPI-6
		F	5	1=Ultra DMA mode 5 and below are supported
		F	4	1=Ultra DMA mode 4 and below are supported
		F	3	1=Ultra DMA mode 3 and below are supported
		F	2	1=Ultra DMA mode 2 and below are supported
		F	1	1=Ultra DMA mode 1 and below are supported
		F	0	1=Ultra DMA mode 0 is supported
89-92			Set as indicated in ATA/ATAPI-6	
93		V	COMRESET result. The contents of this word shall be cleared to zero.	
94-255			Set as indicated in ATA/ATAPI-6	

Key:
M = Support of the word is mandatory.
O = Support of the word is optional.
F = the content of the word is fixed and does not change. For removable media devices, these values may change when media is removed or changed.
V = the contents of the word is variable and may change depending on the state of the device or the commands executed by the device.
X = the content of the word is vendor specific and may be fixed or variable.
R = the content of the word is reserved and shall be zero.

13.2.2.1 Word 0 - 48: Set as indicated in ATA/ATAPI-6

13.2.2.2 Word 49: Capabilities

Bits 15 through 12 of word 49 shall be set as indicated in ATA/ATAPI-6.

Bit 11 of word 49 is used to determine whether a device supports IORDY. This bit shall be set to one, indicating the device supports IORDY operation.

Bit 10 of word 49 is used to indicate a device's ability to enable or disable the use of IORDY. This bit shall be set to one, indicating the device supports the disabling of IORDY. Disabling and enabling of IORDY is accomplished using the SET FEATURES command.

Bit 9 of word 49 shall be set to one.

Bit 8 of word 49 is used to indicate whether the device supports DMA. This bit shall be set to one.

Bits 7 - 0 of word 49 shall be set as indicated in ATA/ATAPI-6.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 429 of 587

13.2.2.3 Words 50 - 52: Set as indicated in ATA/ATAPI-6

13.2.2.4 Word 53: Field validity

Bit 0 shall be set to one. Bit 1 of word 53 shall be set to one, the values reported in words 64 through 70 are valid. Any device that supports PIO mode 3 or above, or supports Multiword DMA mode 1 or above, shall set bit 1 of word 53 to one and support the fields contained in words 64 through 70. Bit 2 of word 53 shall be set to one indicating the device supports Ultra DMA and the values reported in word 88 are valid. Bits 15-3 are reserved.

13.2.2.5 Word 54 - 62: Set as indicated in ATA/ATAPI-6

13.2.2.6 Word 63: Multiword DMA transfer

Bits 2 - 0 of word 63 shall be set to one indicating that the device supports Multiword DMA modes 0, 1, and 2. Bits 15 – 3 shall be set as indicated in ATA/ATAPI-6.

13.2.2.7 Word 64: PIO transfer modes supported

Bits 1 - 0 of word 64 shall be set to one indicating that the device supports PIO modes 3 and 4. Bits 15 – 2 shall be set as indicated in ATA/ATAPI-6.

13.2.2.8 Word 65: Minimum Multiword DMA transfer cycle time per word

Shall be set to indicate 120 ns.

13.2.2.9 Word 66: Device recommended Multiword DMA cycle time

Shall be set to indicate 120 ns.

13.2.2.10 Word 67: Minimum PIO transfer cycle time without flow control

Shall be set to indicate 120 ns.

13.2.2.11 Word 68: Minimum PIO transfer cycle time with IORDY

Shall be set to indicate 120 ns.

13.2.2.12 Words 69-75: Set as indicated in ATA/ATAPI-6

13.2.2.13 Word 76: Serial ATA capabilities

If not 0000h or FFFFh, the device claims compliance with the Serial ATA specification and supports the signaling rate indicated in bits 1-3. Since Serial ATA supports generational compatibility, multiple bits may be set. Bit 0 is reserved and shall be cleared to zero (thus a Serial ATA device has at least one bit cleared in this field and at least one bit set providing clear differentiation). If this field is not 0000h or FFFFh, words 77 through 79 shall be valid. If this field is 0000h or FFFFh the device does not claim compliance with the Serial ATA specification and Words 76 through 79 are not valid and shall be ignored.

Bit 0 shall be cleared to zero.

Bit 1 when set to one indicates that the device is a Serial ATA device and supports the Gen1 signaling rate of 1.5 Gbps.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 430 of 587

Bit 2 when set to one indicates that the device is a Serial ATA device and supports the Gen2 signaling rate of 3.0 Gbps.

Bit 3-7 are reserved for future Serial ATA signaling speed grades and shall be cleared to zero.

Bit 8 is reserved and shall be cleared to zero

Bit 9 when set to one indicates that the Serial ATA device supports the Partial and Slumber interface power management states when initiated by the host.

Bit 10 when set to one indicates that the Serial ATA device supports Phy event counters. If the device supports Phy event counters, it shall support the Phy event counter READ LOG EXT page 11h.

Bit 11-15 are reserved and shall be cleared to zero

13.2.2.14 Word 77: Reserved

Word 77 is reserved for future Serial ATA definition and shall be zero.

13.2.2.15 Word 78: Serial ATA features supported

If Word 76 is not 0000h or FFFFh, Word 78 reports the optional features supported by the device. Support for this word is optional and if not supported the word shall be zero indicating the device has no support for new Serial ATA capabilities.

Bit 0 shall be cleared to zero.

Bit 1-2 are reserved and shall be cleared to zero.

Bit 3 indicates whether the device supports initiating power management requests to the host. When set to one the device supports initiating interface power management requests and when cleared to zero the device does not support initiating power management requests. A device may support reception of power management requests initiated by the host as described in the definition of bit 9 of Word 76 without supporting initiating such power management requests as indicated by this bit.

Bit 4 is reserved and shall be cleared to zero.

Bit 5 indicates whether the device supports asynchronous notification to indicate to the host that attention is required. When set to one the device supports initiating notification events and when cleared to zero the device does not support initiating notification events. An example of an event that the device may need attention for includes a media change. Asynchronous device notification is described in section 13.6.

Bit 6 indicates whether the device supports software settings preservation as defined in section 13.4. When set to one the device supports software settings preservation across COMRESET. When cleared to zero the device clears all software settings when a COMRESET occurs.

Bit 7-15 are reserved and shall be cleared to zero.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 431 of 587

13.2.2.16 Word 79: Serial ATA features enabled

If Word 76 is not 0000h or FFFFh, Word 79 reports which optional features supported by the device are enabled. This word shall be supported if optional Word 78 is supported and shall not be supported if optional Word 78 is not supported.

Bit 0 shall be cleared to zero.

Bit 1-2 are reserved and shall be cleared to zero.

Bit 3 indicates whether device support for initiating power management requests to the host is enabled. When set to one the device may initiate power management transition requests. When cleared to zero the device shall not initiate interface power management requests to the host. This field shall be zero by default.

Bit 4 is reserved and shall be cleared to zero.

Bit 5 indicates whether device support for asynchronous notification to indicate to the host that attention is required is enabled. When set to one the device may initiate notification events. When cleared to zero the device shall not initiate notification events. This field shall be cleared to zero by default. An example of an event that the device may need attention for includes a media change. Asynchronous notification is described in section 13.6.

Bit 6 indicates whether device support for software settings preservation is enabled. When set to one the device shall preserve software settings across COMRESET. When cleared to zero the device shall clear software settings when COMRESET occurs. If the device supports software settings preservation this field shall be one by default. If the device does not support software settings preservation this field shall be zero by default.

Bit 7-15 are reserved and shall be cleared to zero.

13.2.2.17 Words 80-87: Set as indicated in ATA/ATAPI-6

13.2.2.18 Word 88: Ultra DMA modes

Bits 5 - 0 of Word 88 shall be set to one indicating that the device supports Ultra DMA modes 0, 1, 2, 3, 4, and 5. Bits 15 – 5 shall be set as indicated in ATA/ATAPI-6.

13.2.2.19 Word 89 - 92: Set as indicated in ATA/ATAPI-6

13.2.2.20 Word 93: Hardware configuration test results

Word 93 shall be set to 0000h indicating that the word is not supported.

Words 94-255: Set as indicated in ATA/ATAPI-6.

13.2.3 Determining Support for Serial ATA Features

Software should verify a device's Serial ATA capabilities by reading the relevant bits in Words 76-79 of the IDENTIFY (PACKET) DEVICE data. A device claims compliance with the Serial ATA specification by setting IDENTIFY (PACKET) DEVICE Word 76 appropriately.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 432 of 587

Although Serial ATA was first introduced in the ATA/ATAPI specification material in the ATA/ATAPI-7 revision, it cannot be assumed that when Word 80 (Major version number) of the IDENTIFY (PACKET) DEVICE data is read with support of ATA/ATAPI-7 or later that the device supports Serial ATA or any specific Serial ATA features.

13.2.4 SET FEATURES

Devices are informed of host capabilities and have optional features enabled/disabled through the SET FEATURES command defined in the ATA/ATAPI-6 standard. Serial ATA features are controlled using a new features value as defined in Figure 166.

Features Value	Description
10h	Enable use of Serial ATA feature
90h	Disable use of Serial ATA feature

Figure 166 – Features enable/disable values

The Sector Count register contains the specific Serial ATA feature to enable or disable. The specific Serial ATA features are defined as defined in Figure 167.

Sector Count Value	Description
01h	Non-zero buffer offset in DMA Setup FIS
02h	DMA Setup FIS Auto-Activate optimization
03h	Device-initiated interface power state transitions
04h	Guaranteed In-Order Data Delivery
05h	Asynchronous Notification
06h	Software Settings Preservation

Figure 167 – Feature identification values

13.2.4.1 Enable/Disable Non-Zero Offsets in DMA Setup

A Sector Count value of 01h is used by the host to enable or disable non-zero buffer offsets in the DMA Setup FIS when the device utilizes the First-party DMA mechanism (see section 13.5.2.2). By default, non-zero buffer offsets in the DMA Setup FIS are disabled. Enabling non-zero buffer offsets in the DMA Setup FIS is useful for performing out of order data delivery within commands, e.g. delivering the last half of the data before the first half of the data, or to support segmentation of large First-party DMA operations into multiple data phases, i.e. to effectively allow sharing of the SATA link during Data I/O, e.g. to provide a mechanism for the host to transmit additional READ FPDMA QUEUED or WRITE FPDMA QUEUED commands to the device for re-order consideration. The enable/disable state for non-zero offsets in DMA Setup FISes shall be preserved across software reset. The enable/disable state for non-zero offsets in DMA Setup FISes shall be reset to its default state upon COMRESET.

13.2.4.2 Enable/Disable DMA Setup FIS Auto-Activate Optimization

A Sector Count value of 02h is used by the host to enable or disable the DMA Setup FIS optimization for automatically activating transfer of the first host-to-device Data FIS following a DMA Setup FIS with a host-to-device transfer direction. For transfers from the host to the device, First-party DMA transfers require a sequence of DMA Setup FIS followed by a DMA Activate FIS to initiate the transfer. The Auto-Activate optimization allows the DMA Setup FIS operation to imply immediate activation thereby eliminating the need for the additional separate DMA Activate FIS to start the transfer. Enabling the optimization notifies the device that the host bus adapter implementation allows the DMA Setup FIS to include the Auto-Activate bit to trigger immediate

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 433 of 587

transfer following receipt and processing of the DMA Setup FIS. By default, the optimization is disabled. See section 10.3.8.3.1 for additional details. The enable/disable state for the auto-activate optimization shall be preserved across software reset. The enable/disable state for the auto-activate optimization shall be reset to its default state upon COMRESET.

13.2.4.3 Enable/Disable Device-Initiated Interface Power State Transitions

A Sector Count value of 03h is used by the host to enable or disable device initiation of interface power state transitions. By default, the device is not permitted to attempt interface power state transitions by issuing the $PMREQ_P$ primitive to the host. The host may enable device initiation of such interface power state transitions for such cases where it may be desirable for the device to attempt initiating such transitions. The enable/disable state for device initiated power management shall persist across software reset. The enable/disable state shall be reset to its default disabled state upon COMRESET.

If device initiated interface power management is enabled, the device shall not attempt to initiate an interface power state transition between reset and the delivery of the device reset signature. Hosts shall not attempt initiating an interface power state transition between an issued reset and the receipt of the device reset signature. Hosts should not attempt initiating an interface power management request without first verifying the device has such capabilities as determined by the information in the device's IDENTIFY (PACKET) DEVICE data structure.

13.2.4.4 Enable/Disable Guaranteed in-Order Data Delivery

A Sector Count value of 04h is used by the host to enable or disable guaranteed in-order data delivery when the device utilizes the First-party DMA mechanism and non-zero buffer offsets in the DMA Setup FIS. By default, guaranteed in-order data delivery is disabled. Enabling guaranteed in-order data delivery is useful for segmenting large I/O processes into multiple atomic data phases using non-zero buffer offsets in the DMA Setup FIS, while minimizing the complexity that may be imposed on the host with out-of-order data delivery. The enable/disable state for guaranteed in-order data delivery shall be preserved across software reset. The enable/disable state for guaranteed in-order data delivery shall be reset to its default state upon COMRESET.

13.2.4.5 Enable/Disable Asynchronous Notification

A Sector Count value of 05h is used by the host to enable or disable asynchronous notification. By default, asynchronous notification is disabled. The host may enable asynchronous notification in order to allow the device to request attention without the host polling. As an example, this may be useful to avoid polling for media change events in ATAPI devices. The enable/disable state for asynchronous notification shall be preserved across software reset. The enable/disable state for asynchronous notification shall be reset to its default state upon COMRESET.

13.2.4.6 Enable/Disable Software Settings Preservation

A Sector Count value of 06h is used by the host to enable or disable software settings preservation, as defined in section 13.4. By default, if the device supports software settings preservation the feature is enabled on power-up. The enable/disable state for software settings preservation shall persist across software reset. The enable/disable state for software settings preservation shall be reset to its default state upon COMRESET. The host may disable software settings preservation in order to not preserve software settings across COMRESET (and make COMRESET equivalent to hard reset in Parallel ATA).

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 434 of 587

13.2.5 READ LOG EXT Log Directory

Devices supporting READ LOG EXT log page 10h reflect this support in the General Purpose Log Directory page (log page 0) by having the value 1 at offset 20h and the value 0 at offset 21h of that log page to indicate existence of a log page at address 10h of 1-page in length.

Devices supporting READ LOG EXT log page 11h reflect this support in the General Purpose Log Directory page (log page 0) by having the value 1 at offset 22h and the value 0 at offset 23h of that log page to indicate existence of a log page at address 11h of 1-page in length.

Byte	Value
0 - 1Fh	As defined in the ATA/ATAPI-6 standard
20h	1 if Native Command Queuing is supported, 0 if Native Command Queuing is not supported
21h	0
22h	1 if Phy Event Counters are supported 0 if Phy Event Counters are not supported
23h	0

13.3 Device Configuration Overlay

Figure 168 defines additional features and capabilities that support can be controlled for using the Device Configuration Overlay feature in the ATA/ATAPI-6 standard. The device is only required to support setting these features if the device reports support for Device Configuration Overlay in either IDENTIFY DEVICE or IDENTIFY PACKET DEVICE, respectively.

Word	Description
0-7	As defined in the ATA/ATAPI-6 standard
8	Serial ATA command / feature sets supported 15-5 Reserved (0) 4 1 = Supports software settings preservation 3 1 = Supports asynchronous notification 2 1 = Supports interface power management 1 1 = Supports non-zero buffer offsets in DMA Setup FIS[1] 0 1 = Supports Native Command Queuing[1]
9	Reserved for Serial ATA
10-255	As defined in the ATA/ATAPI-6 standard
NOTE: 1. Applicable to non-PACKET devices only – i.e. IDENTIFY DEVICE.	

Figure 168 – Device Configuration Overlay data structure

WORD 8: Serial ATA command / feature sets supported

This word enables configuration of command sets and feature sets.

Bit 0 indicates whether Native Command Queuing shall be supported by the device. When set to one, the drive shall support Native Command Queuing. When cleared to zero, drive support for Native Command Queuing shall be disabled and Word 76 bit 8, Word 78 bit 1, Word 78 bit 2, Word 78 bit 4, Word 79 bit 1, Word 79 bit 2, and Word 79 bit 4 of IDENTIFY DEVICE shall all be cleared to zero. If NCQ is disabled and READ FPDMA QUEUED or WRITE FPDMA QUEUED is issued to the device, the device shall abort the command with the ERR bit set to one in the Status register and the ABRT bit

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 435 of 587

set to one in the Error register. The setting of this bit is applicable to non-PACKET devices only.

Bit 1 indicates whether the drive supports non-zero buffer offsets in the DMA Setup FIS. When set to one, the drive shall support non-zero buffer offsets in the DMA Setup FIS. When cleared to zero, drive support for non-zero buffer offsets in the DMA Setup FIS shall be disabled and Word 78 bit 1, Word 78 bit 4, Word 79 bit 1, and Word 79 bit 4 of IDENTIFY DEVICE shall all be cleared to zero. If non-zero buffer offsets in the DMA Setup FIS are disabled, the device shall only issue a DMA Setup FIS that has the DMA Buffer Offset field cleared to zero. The setting of this bit is applicable to non-PACKET devices only.

Bit 2 indicates whether the drive supports interface power management requests. When set to one, the drive shall support receiving host initiated power management requests and shall support device initiated power management requests. When cleared to zero, the drive shall not support receiving host initiated power management requests and shall not support device initiated power management requests and Word 76 bit 9, Word 78 bit 3, and Word 79 bit 3 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE shall all be cleared to zero. If interface power management requests are disabled, the drive shall respond with $PMNAK_P$ to any interface power management requests and the device shall not initiate $PMREQ\_P_P$ or $PMREQ\_S_P$ primitives to the host.

Bit 3 indicates whether the drive supports asynchronous notification. When set to one, the drive shall support asynchronous notification. When cleared to zero, the drive shall not support asynchronous notification and Word 78 bit 5 and Word 79 bit 5 of IDENTIFY PACKET DEVICE shall both be cleared to zero. When asynchronous notification is disabled, the device shall not initiate a Set Device Bits FIS with the Notification bit set to one.

Bit 4 indicates whether the device supports software settings preservation. When set to one, the device shall support software settings preservation. When cleared to zero, the device shall not support software settings preservation and Word 78 bit 6 and Word 79 bit 6 of IDENTIFY (PACKET) DEVICE shall both be cleared to zero. When software settings preservation is disabled, the device shall not preserve any software settings that are normally cleared upon an ATA hardware reset.

Bit 5-15 are reserved and shall be cleared to zero.

WORD 9: Reserved for Serial ATA

This word is reserved for Serial ATA and all bits shall be cleared to zero.

13.4 Software Settings Preservation (Optional)

When a device is enumerated, software configures the device using SET FEATURES and other commands. These software settings are often preserved across software reset but not necessarily across hardware reset. In Parallel ATA, only commanded hardware resets can occur, thus legacy software only reprograms settings that are cleared for the particular type of reset it has issued. In Serial ATA, COMRESET is equivalent to hard reset and a non-commanded COMRESET may occur if there is an asynchronous loss of signal. Since COMRESET is equivalent to hardware reset, in the case of an asynchronous loss of signal some software settings may be lost without legacy software knowledge. In order to avoid losing important software settings without legacy driver knowledge, the software settings preservation ensures that the value of important software settings is maintained across a COMRESET. Software settings preservation may be enabled or disabled using SET FEATURES with a subcommand

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 436 of 587

code of 06h (refer to section 13.2.4.6). If a device supports software settings preservation, the feature shall be enabled by default.

The software settings that shall be preserved across COMRESET are listed below. The device is only required to preserve the indicated software setting if it supports the particular feature/command the setting is associated with.

INITIALIZE DEVICE PARAMETERS: Device settings established with the INITIALIZE DEVICE PARAMETERS command. This command is obsolete in the ATA/ATAPI-6 standard, and was last defined in the ATA/ATAPI-5 standard.

Power Management Feature Set Standby Timer: The Standby timer used in the Power Management feature set.

Read/Write Stream Error Log: The Read Stream Error Log and Write Stream Error Logs (accessed using READ LOG EXT and WRITE LOG EXT).

Security mode state: The security mode state established by Security Mode feature set commands (refer to the ATA/ATAPI-6 standard). The device shall not transition to a different security mode state based on a COMRESET. For example, the device shall not transition from the SEC5: Unlocked / not Frozen state to state SEC4: Security enabled / Locked when a COMRESET occurs, instead the device shall remain in the SEC5: Unlocked / not Frozen state.

SECURITY FREEZE LOCK: The Frozen mode setting established by the SECURITY FREEZE LOCK command.

SECURITY UNLOCK: The unlock counter that is decremented as part of a failed SECURITY UNLOCK command attempt.

SET MAX ADDRESS (EXT): The maximum LBA specified in SET MAX ADDRESS or SET MAX ADDRESS EXT.

SET FEATURES (Write Cache Enable/Disable): The write cache enable/disable setting established by the SET FEATURES command with subcommand code of 02h or 82h.

SET FEATURES (Set Transfer Mode): PIO, Multiword, and UDMA transfer mode settings established by the SET FEATURES command with subcommand code of 03h.

SET FEATURES (Advanced Power Management Enable/Disable): The advanced power management enable/disable setting established by the SET FEATURES command with subcommand code of 05h or 85h. The advanced power management level established in the Sector Count register when advanced power management is enabled (SET FEATURES subcommand code 05h) shall also be preserved.

SET FEATURES (Read Look-Ahead): The read look-ahead enable/disable setting established by the SET FEATURES command with subcommand code of 55h or AAh.

SET FEATURES (Release Interrupt): The release interrupt enable/disable setting established by the SET FEATURES command with a subcommand code of 5Dh or DDh.

SET FEATURES (SERVICE Interrupt): The SERVICE interrupt enable/disable setting established by the SET FEATURES command with a subcommand code of 5Eh or DEh.

SET FEATURES (Reverting to Defaults): The reverting to power-on defaults enable/disable setting established by the SET FEATURES command with a subcommand code of CCh or 66h.

SET MULTIPLE MODE: The block size established with the SET MULTIPLE MODE command.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 437 of 587

13.5 Native Command Queuing (Optional)

This section defines a simple and streamlined command queuing model for Serial ATA. The command queuing model is derived in part from the legacy queuing definition but with some of the legacy queuing aspects modified to take advantage of the capabilities of Serial ATA as well as to streamline the overall mechanism. A race-free status return mechanism is defined that minimizes the required protocol round trips in order to reduce the incurred overhead.

13.5.1 Overview

The native queuing definition utilizes the reserved 32-bit field in the Set Device Bits FIS to convey the pending status for each of up to 32 outstanding commands. The BSY bit in the Status register conveys only the device's readiness to receive another command, and does not convey the completion status of queued commands. Upon receipt of a new command, the device deasserts the BSY bit before proceeding to execute received commands. The 32 reserved bits in the Set Device Bits FIS are handled as a 32-element array of active command bits (referred to as ACT bits), one for each possible outstanding command, and the array is bit significant such that bit "n" in the array corresponds to the pending status of the command with tag "n."

Data returned by the device (or transferred to the device) for queued commands use the First Party DMA mechanism to cause the host controller to select the appropriate destination/source memory buffer for the transfer. The memory handle used for the buffer selection is the same as the tag that is associated with the command. For traditional desktop host controllers, the handle may be used to index into a vector of pointers to pre-constructed scatter/gather lists (often referred to as physical region descriptor tables or simply PRD tables) in order to establish the proper context in the host's DMA engine. The First-party DMA Data Phase is defined as the period from reception of a DMA Setup FIS until either the associated transfer count is exhausted or the ERR bit in the shadow Status register is set. During this period the host may not issue new commands to the device nor may the device signal new command completions to the host.

Status is returned by updating the 32-element bit array in the Set Device Bits FIS for successful completions. For failed commands, the device halts processing commands allowing host software or controller firmware to intervene and resolve the source of the failure, by using the general purpose logging feature set, before processing is again explicitly restarted.

Devices supporting Native Command Queuing shall implement, and report support for, the general purpose logging feature set as defined in ATA/ATAPI-6. In addition, the device shall implement the Serial ATA defined log page 10h.

13.5.2 Definition

13.5.2.1 Command Issue Mechanism

The Serial ATA transmission protocol is sensitive to the state of the BSY bit in the Status shadow register that provides write protection to the shared Shadow Command Block registers. Since the Shadow Command Block registers can be safely written only when the BSY bit is cleared to zero, the BSY bit conventions defined in the Transport layer shall be adhered to, and issuing a new command shall only be attempted when the BSY bit is cleared to zero. When the BSY bit in the Shadow Status register is cleared to zero, another command may be issued to the device.

The state of the BSY bit in the Shadow Status register shall be checked prior to attempting to issue a new queued command. If the BSY bit is set to one, issuing the next command shall be deferred until the BSY bit is cleared to zero. It is desirable to minimize such command issue deferrals, so devices should clear the BSY bit to zero in a timely manner. Host controllers may have internal designs that mitigate the need for host software to block on the state of the BSY bit.

The native queuing commands include a tag value that identifies the command. The tag value is in the range 0 through 31 inclusive (the device queue depth is limited to 32 outstanding entries),

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 438 of 587

and is conveyed in the Register – Host to Device FIS when the command is issued. For devices that report maximum queue depth less than 32 in their IDENTIFY DEVICE word 75, the host shall issue only unique tag values that have a value less than the value reported. For example, for devices reporting a maximum queue depth of 16, the host shall not issue a tag value greater than 15.

Upon issuing a new native queued command, the bit in the SActive register corresponding to the tag value of the command being issued shall be set to one by the HBA prior to the command being transmitted to the device. Section 14.1.4 describes the SActive register and the access conventions for it.

13.5.2.2 Data Delivery Mechanism

The First-party DMA mechanism is used by the device to transmit (or receive) data for an arbitrary queued command. The command's tag value shall also be the DMA Buffer Identifier used to uniquely identify the source/destination memory buffer for the transfer.

The DMA Setup FIS is used by the device to select the proper transfer buffer prior to each data transfer. Only a single DMA Setup FIS is required at the beginning of each transfer and if the transfer spans multiple Data FISes a new DMA Setup FIS is not required before each Data FIS. Serial ATA host controller hardware shall account for the DMA Setup FIS buffer identifier being a value between 0 and 31 and the host controller shall select the proper transfer buffer based on such an index.

For data transfers from the host to the device, an optimization to the First-party DMA mechanism is included to eliminate one transaction by allowing the requested data to immediately be transmitted to the device following such a request without the need for a subsequent DMA Activate FIS for starting the flow of data. This optimization to the First-party DMA mechanism is defined in section 10.3.8.3.1.

If non-zero buffer offsets in the DMA Setup FIS are not enabled (see section 13.2.4.1) or not supported (see section 13.2.1), the data transfer for a command shall be satisfied to completion following a DMA Setup FIS before data transfer for a different command may be started. Host controllers are not required to preserve DMA engine context upon receipt of a new DMA Setup FIS, and if non-zero buffer offsets are not enabled or not supported, a device cannot resume data transfer for a previously abandoned context at the point where it left off.

If the host controller hardware supports non-zero buffer offsets in the DMA Setup FIS and use of non-zero offsets is enabled, and if guaranteed in-order data delivery is either not supported by the device (see section 13.2.1) or is disabled (see section 13.2.4.4), the device may return (or receive) data for a given command out of order (i.e. returning data for the last half of the command first). In this case the device may also interleave partial data delivery for multiple commands provided the device keeps track of the appropriate buffer offsets. For example, data for the first half of command 0 may be delivered followed by data for the first half of command 1 followed by the remaining data for command 0. By default use of non-zero buffer offsets is disabled. See section 13.2.4.1 for information on enabling non-zero buffer offsets for the DMA Setup FIS.

If the host controller hardware supports non-zero buffer offsets in the DMA Setup FIS and use of non-zero offsets is enabled, and if the device supports guaranteed in-order data delivery and guaranteed in-order data delivery is enabled, the device may use multiple DMA Setup FISes to satisfy a particular I/O process, but if multiple DMA Setup FISes are used, the data shall be delivered in-order, starting at the first LBA. In this case the device may not interleave partial data delivery for either individual or multiple commands. For example, data for the first half of a command may be delivered using one DMA Setup FIS and one or more subsequent Data FISes, followed by the remaining data for that command, delivered using a second DMA Setup FIS and

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 439 of 587

one or more subsequent Data FISes. Non-zero buffer offsets are used as in the more general out-of-order data delivery case described above. By default use of guaranteed in-order data delivery is disabled.

For selecting the memory buffer for data transfers, the DMA Setup FIS is issued by the device. The DMA Setup FIS fields are defined in Figure 169 (see also section 10.3.8).

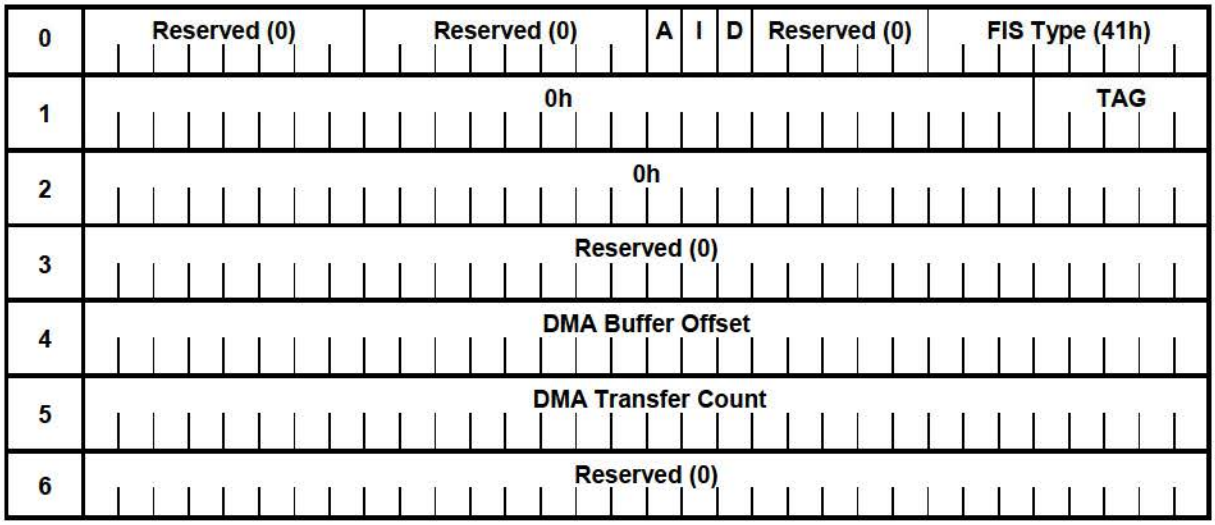


Figure 169 – DMA Setup FIS definition for memory buffer selection

Field Definitions

FIS Type — As defined in section 10.3.8.

D — As defined in section 10.3.8. Since the DMA Setup FIS is only issued by the device for the queuing model defined here, the value in the field is defined as 1 = device to host transfer (write to host memory), 0 = host to device transfer (read from host memory).

A — As defined in section 10.3.8, including additional details in section 10.3.8.3.1. For DMA Setup with transfer direction from device to host, this bit shall be zero.

TAG — This field is used to identify the DMA buffer region in host memory to select for the data transfer. The low order 5 bits of the DMA Buffer Identifier Low field shall be set to the TAG value corresponding to the command TAG for which data is being transferred. The remaining bits of the DMA Buffer Identifier Low/High shall be cleared to zero. The 64-bit Buffer Identifier field defined in the DMA Setup FIS in section 10.3.8 is used to convey a TAG value that occupies the five least-significant bits of the field.

DMA Buffer Offset
As defined in section 10.3.8. The device may specify a non-zero value in this field only if the host indicates support for it through the SET FEATURES mechanism defined in section 13.2.4. Data is transferred to/from sequentially increasing logical addresses starting at the specified offset in the specified buffer.

DMA Transfer Count
As defined in section 10.3.8. The value shall accurately reflect the length of the data transfer to follow. Refer to section 10.3.11.2 for special considerations if the transfer count is for an odd number of words. Devices shall not set this field to 0h; a value of 0h for this field is illegal and results in indeterminate behavior.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

I Interrupt Native Command Queuing does not make use of an interrupt following the data transfer phase (after the transfer count is exhausted). The I bit shall be cleared to zero.

R/Reserved All reserved fields shall be cleared to zero.

13.5.2.3 Success Status Mechanism

For maximum efficiency, the status return mechanism is not interlocked (does not include a handshake) while at the same time ensuring no status notifications are lost or overwritten (i.e. status notifications are race-free). The status return mechanism relies on an array of ACT bits – one ACT bit to convey the active status for each of the 32 possible outstanding commands, resulting in a 32-bit ACT status field. The 32-bit reserved field in the Set Device Bits FIS as defined in section 10.3.6 is defined as the SActive field and is used to convey command completion information for updating the ACT bit array. The zero bit position in the 32-bit field corresponds to the ACT bit for the command with tag value of zero. Host software shall check the SActive register (containing the ACT bit array) when checking status in order to determine which command(s) have completed since the last time the host processed a command completion. It is possible for multiple commands to indicate completion by the time the host checks the status due to the software latencies in the host (i.e. by the time the host responds to one completion notification, another command may have completed). Only successfully completed commands indicate their status using this mechanism – failed commands use an additional mechanism described in section 13.5.2.4 to convey error information as well as the affected command tag.

13.5.2.3.1 Outputs

Register	7	6	5	4	3	2	1	0
Error	0							
Sector Count	na							
Sector Count (exp)	na							
LBA Low	na							
LBA Low (exp)	na							
LBA Mid	na							
LBA Mid (exp)	na							
LBA High	na							
LBA High (exp)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	31	30	29	…	2	1	0
SActive*	ACT 31:0						

Figure 170 – Register values for successful result

*The SActive register is defined in section 14.1.4.

ACT Bit positions set to one for each command TAG still outstanding. Device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.

BSY 0: device is prepared to receive another command
1: device is not prepared to receive another command

DRDY 1

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 441 of 587

DF	0
DRQ	0
ERR	0
na	As defined in the ATA/ATAPI-6 standard

Upon successful completion of an outstanding command, the device shall transmit a Set Device Bits FIS with bits set in the SActive field corresponding to the bit position for each command TAG that has completed since the last status notification was transmitted. The ERR bit in the Status register shall be cleared and the value in the Error register shall be zero.

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged as indicated by "na" in the table.

The SActive field occupies the last 32 bits of the Set Device Bits FIS as defined in Figure 171.

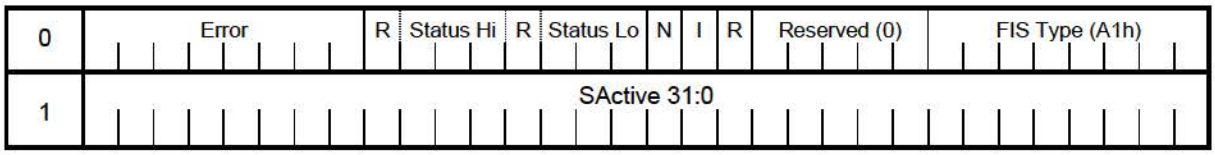


Figure 171 – Set Device Bits FIS and SActive field

SActive	The SActive field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns. The device shall only indicate completion notification for a command if it has completed successfully.

All other fields as defined in section 10.3.6.

13.5.2.4 Error Handling Mechanism

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Register FIS to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field of the FIS and stop all further command processing until host software can respond to the error condition with appropriate action. If all commands have been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Set Device Bits FIS to the host with the ERR bit set to one in the Status field of the FIS and stop all further command processing until host software can respond to the error condition with appropriate action. All outstanding commands issued to the device at the time of an error are aborted as part of the error response and may be re-issued as appropriate by the host. Upon detecting an error when there are one or more queued commands outstanding, the device shall stop processing commands until a READ LOG EXT command with a log page of 10h is issued. When a READ LOG EXT command with a log page of 10h is issued, the device shall abort any outstanding queued commands and perform any necessary cleanup before returning detailed error information for the last failed command including the tag value for the failed command as described in section 13.5.4.3. The READ LOG EXT page reflects the error information for the first recorded erring queued command until such time as another erring queued command is encountered after the execution of a READ LOG EXT command with a log page address of 10h. READ LOG EXT log page 10h is not valid after a software reset or a COMRESET and values read by software after either reset are indeterminate.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

Issuing the READ LOG EXT command thus not only retrieves extended error information but also results in the device aborting any remaining queued commands and performing any cleanup tasks necessary to again be ready to process commands.

The host may perform the following functions to detect and handle an error condition for a queued command:

1. When the host receives status notification for a command, by polling or interrupt, the host checks if any outstanding queued command has completed successfully and indicates successful completion to the operating system for those commands. The host can determine which commands have successfully completed since the last time status was checked by comparing the value in the SActive register with the previously stored value.
2. The host checks the ERR bit in the Status register to determine if any queued command has failed. At this point the host is aware that an error has occurred.
3. The host issues a READ LOG EXT command with a log address of 10h. The READ LOG EXT command causes the device to abort any commands remaining in the queue. The command returns detailed information about the error condition encountered including:
   - The tag value of the queued command that failed
   - The ATA Shadow Register Block image including error code for the queued command that failed

   The information returned by the READ LOG EXT with a log address of 10h is persistent and any subsequent read of this log information returns the same data until such time as a new queued error condition is encountered after the execution of a READ LOG EXT command with a log page address of 10h. READ LOG EXT log page 10h is not valid after a software reset or a COMRESET and values read by software after either reset are indeterminate.
4. The host can now re-issue any aborted commands and/or begin sending new commands to the device.

If the device signals an error condition with a Register FIS, the device shall not send a subsequent Set Device Bits FIS to update the SActive register with any outstanding command completions. When an error is indicated, the host shall treat any outstanding commands that do not have their corresponding SActive register bit cleared as failed. Devices should be aware that if choosing to aggregate status to the point where many of the outstanding commands have actually completed successfully without notification to the host, that an error may cause the final completion status of those commands to be failure. The device should be selective when using status aggregation for outstanding queued commands to ensure the host is made aware of successful completion for outstanding commands in a way that an error would not force a high number of unnecessary command retries.

The device shall only signal completion notification for commands that have completed successfully or for queued commands that are aborted as a result of the host issuing a READ LOG EXT command with log page of 10h. This means that a failed queued command always has its corresponding bit in the SActive shadow register set to one until cleared as a consequence of the host issuing a READ LOG EXT command. A device clears all SActive shadow register bits in response to a received READ LOG EXT command by transmitting a Set Device Bits FIS to the host with all the bits in the SActive field set to one. This policy avoids the host inadvertently completing a failed command with successful status.

13.5.2.4.1 Outputs

Register	7	6	5	4	3	2	1	0
Error	ERROR							
Sector Count	na							
Sector Count (exp)	na							

LBA Low	na							
LBA Low (exp)	na							
LBA Mid	na							
LBA Mid (exp)	na							
LBA High	na							
LBA High (exp)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**	**2**	**1**	**0**
SActive	ACT 31:0						

Figure 172 – Register values for error result

ERROR ATA error code

ACT Bit positions set to one for each command TAG still outstanding. Only successfully completed commands have bit positions zeroed. Device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.

BSY 0

DRDY 1

DF 0

DRQ 0

ERR 1

na As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged as indicated by "na" in the table.

13.5.3 Intermixing Non-Native Queued Commands and Native Queued Commands

The host shall not issue a non-native queued command while a native queued command is outstanding. Upon receiving a non-native queued command while a native queued command is outstanding, the device shall signal the error condition to the host by transmitting a Register FIS to the host with the ERR and ABRT bits set to one and the BSY bit cleared to zero in the Status field of the FIS and halt command processing as described in section 13.5.2.4 except as noted below. Non-native queued commands include all commands other than the READ FPDMA QUEUED and WRITE FPDMA QUEUED commands defined in section 13.5.4.1 and section 13.5.4.2.

Reception of the READ LOG EXT command with a specified log page of 10h after an error has occurred shall cause any outstanding Serial ATA native queued commands to be aborted, and the device shall perform necessary state cleanup to return to a state with no commands pending and shall clear all bits in the SActive shadow register by transmitting a Set Device Bits FIS to the host with bits set to one in the SActive field for each bit position to be cleared in the SActive register. After completing a READ LOG EXT command with a specified log page of 10h, the device shall be prepared to execute subsequently issued queued commands regardless of any previous errors on a queued command.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 444 of 587

In the case that a READ LOG EXT command with a log page of 10h is issued while a native queued command is outstanding AND no error was previously reported by the device, the device shall signal an error condition. The receipt of this command when no error is outstanding shall be handled as any other non-native queued command when a native queued command is outstanding. In this case, a subsequent READ LOG EXT command with log page of 10h is required to recover from the error.

13.5.4 Command Definitions

13.5.4.1 READ FPDMA QUEUED

Queued native read commands make use of a new command. The new command supports LBA mode only and uses 48-bit addressing only. The format of the new command is defined in Figure 173.

13.5.4.1.1 Inputs

Register	7	6	5	4	3	2	1	0
Features	Sector Count 7:0							
Features (exp)	Sector Count 15:8							
Sector Count	TAG					Reserved		
Sector Count (exp)	Reserved							
LBA Low	LBA 7:0							
LBA Low (exp)	LBA 31:24							
LBA Mid	LBA 15:8							
LBA Mid (exp)	LBA 39:32							
LBA High	LBA 23:16							
LBA High (exp)	LBA 47:40							
Device	FUA	1	Res	0	Reserved			
Command	60h							

Figure 173 – READ FPDMA QUEUED command definition

TAG — The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. The assigned TAG value shall not exceed the value specified in IDENTIFY DEVICE word 75.

FUA — When set to one forces the data to be retrieved from the storage media regardless of whether the storage device holds the requested information in its buffers or cache. If the device holds a modified copy of the requested data as a result of having cached writes, the modified data is first written to the media before being retrieved from the storage media as part of this operation. When cleared to zero the data may be retrieved either from the device's storage media or from buffers/cache that the device may include.

Others — All other registers have contents consistent with the READ DMA QUEUED EXT command defined in the ATA/ATAPI-6 standard, including the Sector Count 15:0 convention where a value of zero specifies that 65,536 sectors are to be transferred.

Upon accepting the command, the device shall clear the BSY bit if/when it is prepared to receive another command by transmitting a Register – Device to Host FIS to the host with the BSY bit cleared in the Status field of the FIS. The ability for the device to quickly clear the BSY bit allows the host to issue another queued command without blocking on this bit. The host shall check the BSY bit in the shadow Status register before attempting to issue a new command in order to

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 445 of 587

determine whether the device is ready to receive another command (and determine that the host has write access to the Shadow Register Block Registers). The device shall not trigger an interrupt in response to having successfully received the command, so the Register – Device to Host FIS that the device transmits to clear BSY shall have the I bit cleared to zero.

13.5.4.1.2 Success Outputs

Upon successful completion of an outstanding command, the device shall clear the bit in the SActive shadow register that corresponds to the bit position for the command TAG completing, and alert the host by transmitting a Set Device Bits FIS that has the I bit set to one and has bit positions in the SActive field set to one for each bit position to be cleared in the host SActive shadow register (as defined in 14.1.4). The ERR bit in the Status register shall be cleared and the value in the Error register shall be zero.

Register	7	6	5	4	3	2	1	0
Error	0							
Sector Count	na							
Sector Count (exp)	na							
LBA Low	na							
LBA Low (exp)	na							
LBA Mid	na							
LBA Mid (exp)	na							
LBA High	na							
LBA High (exp)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	31	30	29	…	2	1	0
SActive*	ACT 31:0						

Figure 174 – READ FPDMA QUEUED success status result values

*The SActive register is defined in section 14.1.4.

ACT	Bit positions set to one for each command TAG still outstanding. The device clears bit positions in the host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.
BSY	Unchanged from its previous value (Set Device Bits FIS does not write this field)
DRDY	1
DF	0
DRQ	Unchanged from its previous value (Set Device Bits FIS does not write this field)
ERR	0
na	As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged, as shown by “na” in the table.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 446 of 587

13.5.4.1.3 Error Outputs

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register FIS to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field, the ATA error code in the Error field, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 13.5.2.4. If all commands have been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Set Device Bits FIS to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 13.5.2.4. The device shall leave the bit positions in the SActive shadow register corresponding to the erring tag value and any uncompleted queued commands set.

Register	7	6	5	4	3	2	1	0
Error	ERROR							
Sector Count	na							
Sector Count (exp)	na							
LBA Low	na							
LBA Low (exp)	na							
LBA Mid	na							
LBA Mid (exp)	na							
LBA High	na							
LBA High (exp)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	31	30	29	...	2	1	0
SActive	ACT 31:0						

Figure 175 – READ FPDMA QUEUED error status result values

ACT	Bit positions set to one for each command TAG outstanding. The failed command is considered as still outstanding since it has not completed successfully. The device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set to one for each bit position to be cleared in the SActive register.
ERROR	ATA error code for the failure condition of the failed command
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1
na	As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the shadow Status register is cleared, leaving the other Shadow Register Block Registers unchanged, as shown by “na” in the table. If the device signals an error condition when the BSY bit in the shadow Status register is set, the device clears the BSY bit with a Register FIS which updates all registers in the Shadow Register Block, but the

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 447 of 587

corresponding error information for the command is still retrieved using the READ LOG EXT command with a log page of 10h.

13.5.4.2 WRITE FPDMA QUEUED

Queued native write commands make use of a new command. The format of the new command is defined in Figure 176.

13.5.4.2.1 Inputs

Register	7	6	5	4	3	2	1	0
Features	Sector Count 7:0							
Features (exp)	Sector Count 15:8							
Sector Count	TAG					Reserved		
Sector Count (exp)	Reserved							
LBA Low	LBA 7:0							
LBA Low (exp)	LBA 31:24							
LBA Mid	LBA 15:8							
LBA Mid (exp)	LBA 39:32							
LBA High	LBA 23:16							
LBA High (exp)	LBA 47:40							
Device	FUA	1	0	0	Reserved			
Command	61h							

Figure 176 – WRITE FPDMA QUEUED command definition

TAG — The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. The assigned TAG value shall not exceed the value specified in IDENTIFY DEVICE word 75.

FUA — When set to one forces the data to be written to the storage media before completion status is indicated. When cleared to zero the device may indicate completion status before the data is committed to the media.

Others — All other registers as specified for the WRITE DMA QUEUED EXT command defined in the ATA/ATAPI-6 standard, including the Sector Count 15:0 convention where a value of zero specifies that 65,536 sectors are to be transferred.

Upon accepting the command, the device shall clear the BSY bit if/when it is prepared to receive another command by transmitting a Register FIS to the host with the BSY bit cleared to zero in the Status field of the FIS. The ability for the device to quickly clear the BSY bit allows the host to issue another queued command without blocking on this bit. The host shall check the BSY bit in the shadow Status register before attempting to issue a new command in order to determine that the device is ready to receive another command (and determine that the host has write access to the Shadow Register Block Registers). The device shall not trigger an interrupt in response to having successfully received the command, so the initial status return that clears BSY shall not have an interrupt associated with it.

13.5.4.2.2 Success Outputs

Upon successful completion of an outstanding command, the device shall clear the bit in the SActive field that corresponds to the bit position for the command tag completing by transmitting a Set Device Bits FIS that has bit positions in the SActive field set to one for each bit position to

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 448 of 587

be cleared in the host SActive shadow register as defined in 14.1.4, and shall trigger an interrupt. The ERR bit in the Status register shall be cleared and the value in the Error register shall be zero.

Register	7	6	5	4	3	2	1	0
Error	0							
Sector Count	na							
Sector Count (exp)	na							
LBA Low	na							
LBA Low (exp)	na							
LBA Mid	na							
LBA Mid (exp)	na							
LBA High	na							
LBA High (exp)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	31	30	29	…	2	1	0
SActive*	ACT 31:0						

Figure 177 – WRITE FPDMA QUEUED success status result values

*The SActive register is defined in section 14.1.4.

ACT	Bit positions set to one for each command TAG still outstanding. Device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.
BSY	Unchanged from its previous value (Set Device Bits FIS does not write this field)
DRDY	1
DF	0
DRQ	Unchanged from its previous value (Set Device Bits FIS does not write this field)
ERR	0
na	As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged, as shown by “na” in the table.

13.5.4.2.3 Error Outputs

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register FIS to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field, the ATA error code in the Error field, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 13.5.2.4. If all commands have been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Set Device Bits FIS to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 13.5.2.4. The device shall leave the bit positions in the

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 449 of 587

SActive shadow register corresponding to the erring tag value and any uncompleted queued commands set.

Register	7	6	5	4	3	2	1	0
Error	ERROR							
Sector Count	na							
Sector Count (exp)	na							
LBA Low	na							
LBA Low (exp)	na							
LBA Mid	na							
LBA Mid (exp)	na							
LBA High	na							
LBA High (exp)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	31	30	29	...	2	1	0
SActive	ACT 31:0						

Figure 178 – WRITE FPDMA QUEUED error status result values

ACT	Bit positions set to one for each command TAG outstanding. The failed command is considered as still outstanding since it has not completed successfully. The device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set to one for each bit position to be cleared in the SActive register.
ERROR	ATA error code for the failure condition of the failed command
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1
na	As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the shadow Status register is cleared, leaving the other Shadow Register Block Registers unchanged, as shown by “na” in the table. If the device signals an error condition when the BSY bit in the shadow Status register is set, the device clears the BSY bit with a Register FIS which updates all registers in the Shadow Register Block.

13.5.4.3 READ LOG EXT Command Error Page

The error-handling scheme for native queued commands halts processing of commands after the host is notified of an error on a native queued command. This allows host software to intervene and take appropriate action to resolve the error and avoids the potential for inconsistency due to data dependencies in the outstanding commands. The host explicitly restarts command processing by issuing a specific command to the device that results in the device aborting all remaining outstanding commands. Because the shadow Status and Error registers are not sufficiently large to contain both information about the error condition and the tag identifying the

erring queued command, an additional log page has been added in order for the host to be able to retrieve additional information for erring queued commands.

The READ LOG EXT command is a general facility as defined in the ATA/ATAPI-6 standard. Reading the queued error log page (page 10h) has the additional side effect defined in section 13.5.3 of aborting any outstanding queued commands and returns a device that has halted due to a queued command error to a state where it has no commands outstanding and is again ready to accept commands (i.e. after completion of the READ LOG EXT command the device returns to state DI0:Device_idle state as defined in section 11.2). Log page 10h for READ LOG EXT returns extended command error information.

13.5.4.3.1 READ LOG EXT Log Page 10h

Devices supporting the native queued capability shall support READ LOG EXT log page 10h. Page 10h is one page in length and is defined in Figure 179.

Byte	7	6	5	4	3	2	1	0
0	NQ	Reserved		TAG				
1	Reserved							
2	Status							
3	Error							
4	LBA Low							
5	LBA Mid							
6	LBA High							
7	Device							
8	LBA Low Exp							
9	LBA Mid Exp							
10	LBA High Exp							
11	Reserved							
12	Sector Count							
13	Sector Count Exp							
14	Reserved							
15	Reserved							
16	Reserved							
17	Reserved							
18	Reserved							
19	Reserved							
20	Reserved							
…								
255								
256	Vendor Specific							
…								
510								
511	Data Structure Checksum							

Figure 179 – READ LOG EXT Log Page 10h data structure definition

TAG — If the NQ bit is cleared, the TAG field contains the TAG corresponding to the queued command that failed.

NQ — If set indicates that the error condition was a result of a non-queued command having been issued and that the TAG field is therefore not valid. If cleared

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

indicates that the TAG field is valid and that the error condition applies to a queued command.

BYTE1-19 An image of a device to host Register FIS is embedded in the data structure. The fields correspond to the Shadow Register Block Registers and are encoded with error information as defined in the ATA/ATAPI-6 standard.

ERROR The value corresponding to the ATA ERROR register value for the command that failed. The command-specific error condition of invalid tag value shall be handled as an invalid command parameter and shall be reported as such (i.e. ABRT bit set in the error register and all other bits cleared).

Note that the value returned in the ERROR field of the data structure is separate from the value returned in the Error shadow register when the initial error condition is signaled. The Error shadow register value is used for the purpose of signaling a queued command error, while the value in the ERROR field of the data structure provides specific information about the error condition that the specific queued command encountered.

Vendor Specific
Allocated for vendor specific use.

Data Structure Checksum
The data structure checksum is the 2's complement of the sum of the first 511 bytes in the data structure. Each byte shall be added with unsigned arithmetic and overflow shall be ignored. The sum of all 512 bytes of the data structure is zero when the checksum is correct.

Reserved All reserved fields shall be cleared to zero

13.5.5 First-party DMA HBA Support (Informative)

The Serial ATA native queuing model utilizes the First-party DMA mechanism to allow the device to select the appropriate host memory buffer to transfer data to or from. The First-party DMA mechanism ensures memory protection in order to avoid a rogue or errant device indiscriminately accessing host memory. This is accomplished in Serial ATA by having the device only refer to memory buffers by a DMA Buffer Identifier, rather than through the use of physical memory addresses.

For the Native Command Queuing protocol, the buffer identifier used for selecting memory buffers is the same as the unique tag value used to identify the corresponding command. The tags have a value in the range 0 to 31 inclusive and correspond to the tag values assigned by the host at the time commands are issued to the device.

For mainstream desktop host controllers, upon receipt of a DMA Setup FIS the buffer identifier may be used by the host as an index into a vector of pointers to pre-constructed PRD tables (physical region descriptor tables, also commonly referred to as scatter/gather lists) that correspond to the memory buffers for the various outstanding queued commands. The pointer in the vector table at the appropriate index may be transferred into the DMA engine as the base pointer for the active PRD table, effectively causing the DMA engine to select the corresponding memory buffer for subsequent data transfers. This allows minimal change to the existing host DMA architecture and provides a streamlined and efficient buffer selection mechanism.

For such an implementation, host software would be responsible for pre-constructing corresponding PRD tables and updating the vector table entry prior to issuing a new native queued command. Figure 180 illustrates these concepts (the figure is intended as illustrative and does not exclude other possible host controller implementations).

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 452 of 587

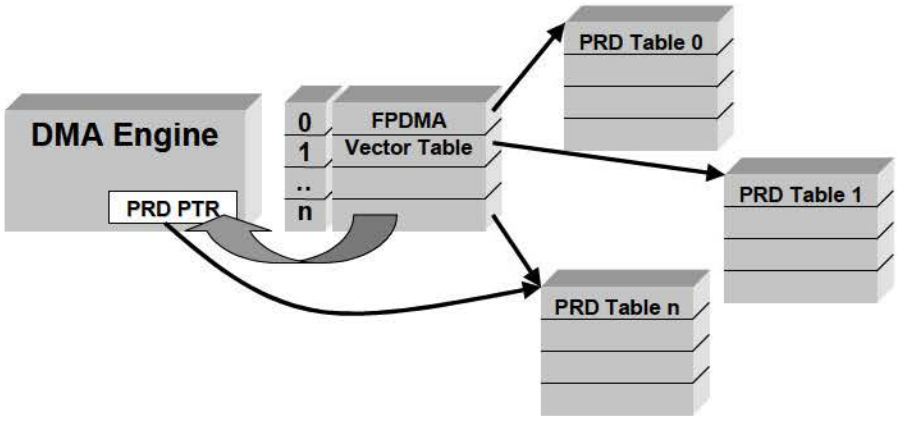


Figure 180 – Example DMA engine indirection for First-party DMA support

This illustrative host controller implementation for supporting First-party DMA has a known shortcoming in that handling non-zero buffer offsets for First-party DMA accesses is cumbersome since the entries in the pre-constructed PRD tables do not necessarily have uniform lengths. For the Native Command Queuing model, there is no requirement for non-zero buffer offset support, however, if out of order data delivery within commands is desired (for example, data for a given command is return by delivering the last half of the data first followed by the first half of the data), support for non-zero buffer offsets is required. See section 13.2.4 for information on non-zero buffer offsets.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

13.6 Asynchronous Notification (Optional)

Asynchronous notification is a mechanism for a device to send a notification to the host that the device requires attention. A few examples of how this mechanism could be used include indicating media has been inserted in an ATAPI device or indicating that a hot plug event has occurred on a Port Multiplier port.

This definition does not list all events that cause a device to generate an asynchronous notification. The mechanism that the host uses to determine the event and the action that is required is outside the scope of this specification, refer to the command set specification for the specific device for more information.

13.6.1 Set Device Bits FIS Notification bit

The Set Device Bits FIS Notification 'N' bit is used by devices to notify the host that attention is needed. The N bit is set to one by a device when the device needs attention, otherwise it is cleared to zero.

By default, a device shall not set the N bit to one in the Set Device Bits FIS. The device shall have the Asynchronous Notification feature enabled by the host before the device may set the N bit to one in the Set Device Bits FIS. After receiving a Set Device Bits with the N bit set, it is the responsibility of the host to interrogate the device and determine what type of action is needed.

13.6.2 Notification Mechanism

To indicate that the device needs attention, the device shall issue a Set Devices Bits FIS to the host with the Interrupt 'I' bit set to one and the Notification 'N' bit set to one. The Error, Status Hi, and Status Lo fields shall accurately reflect the current values of the corresponding register fields in the device.

Reception of a Set Device Bits FIS with the Notification bit set to one may be reflected to the host using the SNotification register, defined in section 14.1.5. A host may support Asynchronous Notification without supporting the SNotification register. The requirement for a host to support Asynchronous Notification is that it can generate an interrupt when the Set Device Bits FIS is received; the SNotification register enables software to be sure that the interrupt was due to a notification event.

13.6.3 State Diagram for Asynchronous Notification

The following state diagram defines the required behavior if the device supports Asynchronous Notification. A device shall only send a Set Device Bits FIS with the Notification bit set when it is in a state that can explicitly send a Set Device Bits FIS to the host as described by the command layer state machines. The state machine is entered from the Device_idle state defined in section 11.2. A device that supports asynchronous notification is also required to support the enhancements for Asynchronous Notification support in the Device_idle and Check_command states in section 11.2.

The state diagram utilizes an internal variable called NotifyPending. The NotifyPending variable indicates that an asynchronous notification has been sent to the host. When NotifyPending is cleared to zero and an event occurs that requires attention, an asynchronous notification may be sent to the host. When NotifyPending is set to one and an event occurs that requires attention, an asynchronous notification shall not be sent to the host since the host has not yet acknowledged reception of the last asynchronous notification received. The host acknowledges reception of an asynchronous notification by sending a Register FIS to the device.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 454 of 587

AN0: Notify_host	Transmit Set Device Bits FIS to host that has 'I' bit set to one, 'N' bit set to one, current Status and Error values, and all Reserved fields cleared to zero. Set NotifyPending to 1.	
1. Unconditional	→	DI0: Device_idle

AN0: Notify_host: This state is entered when the asynchronous notification feature is enabled and an asynchronous notification needs to be sent to the host.

When in this state, the device shall issue a Set Device Bits FIS to the host that has the Interrupt 'I' bit set to one, the Notification 'N' bit set to one, current Status and Error field values, and all Reserved fields cleared to zero. The internal variable NotifyPending shall be set to one to indicate that a notification has been sent to the host that has not yet been acknowledged by the host through the reception of a Register FIS from the host.

Transition AN0:1: The device shall unconditionally transition to the state DI0: Device_idle state.

13.6.4 ATAPI Notification

An ATAPI device shall indicate whether it supports asynchronous notification in Word 78 of IDENTIFY PACKET DEVICE, refer to section 13.2.2. The feature is enabled by using SET FEATURES, as defined in section 13.2.4.5.

13.6.4.1 Event Example (Informative)

An example of an event that may cause the device to generate an asynchronous notification to the host to request attention is the Media Change Event. The Media Change Event occurs when an ATAPI device has detected a change in device state – either media has been inserted or removed.

13.7 Phy Event Counters (Optional)

Phy event counters are an optional feature to obtain more information about Phy level events that occur on the interface. This information may aid designers and integrators in testing and evaluating the quality of the interface. A device indicates whether it supports the Phy event counter feature in IDENTIFY (PACKET) DEVICE Word 76, bit 10. The host determines the current values of Phy event counters by issuing the READ LOG EXT command with a log page of 11h. The counter values shall not be retained across power cycles. The counter values shall be preserved across COMRESET and software resets.

The counters defined can be grouped into three basic categories: those that count events that occur during Data FIS transfers, those that count events that occur during non-Data FIS transfers, and events that are unrelated to FIS transfers. Counters related to events that occur during FIS transfers may count events related to host-to-device FIS transfers, device-to-host FIS transfers, or bi-directional FIS transfers. A counter that records bi-directional events is not required to be the sum of the counters that record the same events that occur on device-to-host FIS transfers and host-to-device FIS transfers.

Implementations that support Phy event counters shall implement all mandatory counters, and may support any of the optional counters as shown in Figure 24. Note that some counters may increment differently based on the speed at which non-Data FIS retries are performed by the host and device. Implementations may record CRC and non-CRC error events differently. For example, there is a strong likelihood that a disparity error may cause a CRC error. Thus, the disparity error may cause both the event counter that records non-CRC events and the event counter that records CRC events to be incremented for the same event. Another example

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 455 of 587

implementation difference is how a missing EOF_P event is recorded; a missing EOF_P primitive may imply a bad CRC even though the CRC on the FIS may be correct. These examples illustrate that some Phy event counters are sensitive to the implementation of the counters themselves, and thus these implementation sensitive counters cannot be used as an absolute measure of interface quality between different implementations.

Devices supporting Phy Event Counters shall implement, and report support for, the general purpose logging feature set as defined in the ATA/ATAPI-6 standard. In addition, the device shall implement the Serial ATA defined log page 11h.

13.7.1 Counter Reset Mechanisms

There are two mechanisms by which the host can explicitly cause the Phy counters to be reset. The first mechanism is to issue a BIST Activate FIS to the device. Upon reception of a BIST Activate FIS the device shall reset all Phy event counters to their reset value. The second mechanism uses the READ LOG EXT command. When the device receives a READ LOG EXT command for log page 11h and bit 0 in the Features register is set to one, the device shall return the current counter values for the command and then reset all Phy event counter values.

13.7.2 Counter Identifiers

Each counter begins with a 16-bit identifier. Figure 181 defines the counter value for each identifier. Any unused counter slots in the log page should have a counter identifier value of 0h. Optional counters that are not implemented shall not be returned in log page 11h. A value of '0' returned for a counter means that there have been no instances of that particular event. There is no required ordering for event counters within the log page; the order is arbitrary and selected by the device vendor.

Bits 14:12 of the counter identifier convey the number of significant bits that counter uses. All counter values consume a multiple of 16-bits. The valid values for bits 14:12 and the corresponding counter sizes are:

1h	16-bit counter
2h	32-bit counter
3h	48-bit counter
4h	64-bit counter

Any counter that has an identifier with bit 15 set to one is vendor specific. This creates a vendor specific range of counter identifiers from 8000h to FFFFh. Vendor specific counters shall observe the number of significant bits 14:12 as defined above.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 456 of 587

Identifier (Bits 11:0)	Mandatory/ Optional	Description
000h	Mandatory	No counter value; marks end of counters in the page
001h	Mandatory	Command failed and ICRC error bit set to one in Error register
002h	Optional	R_ERR$_P$ response for Data FIS
003h	Optional	R_ERR$_P$ response for Device-to-Host Data FIS
004h	Optional	R_ERR$_P$ response for Host-to-Device Data FIS
005h	Optional	R_ERR$_P$ response for non-Data FIS
006h	Optional	R_ERR$_P$ response for Device-to-Host non-Data FIS
007h	Optional	R_ERR$_P$ response for Host-to-Device non-Data FIS
008h	Optional	Device-to-Host non-Data FIS retries
009h	Optional	Transitions from drive PHYRDY to drive PHYRDYn
00Ah	Mandatory	Signature Device-to-Host Register FISes sent due to a COMRESET
00Bh	Optional	CRC errors within a Host-to-Device FIS
00Dh	Optional	Non-CRC errors within a Host-to-Device FIS
00Fh	Optional	R_ERR$_P$ response for Host-to-Device Data FIS due to CRC errors
010h	Optional	R_ERR$_P$ response for Host-to-Device Data FIS due to non-CRC errors
012h	Optional	R_ERR$_P$ response for Host-to-Device non-Data FIS due to CRC errors
013h	Optional	R_ERR$_P$ response for Host-to-Device non-Data FIS due to non-CRC errors
C00h	Optional	(Port Multiplier) Host-to-Device non-Data FIS R_ERR$_P$ ending status due to collision
C01h	Optional	(Port Multiplier) Signature Register – Device-to-Host FISes
C02h	Optional	(Port Multiplier) Corrupt CRC propagation of Device-to-Host FISes

Figure 181 – Phy Event Counter Identifiers

FISes that are terminated due to reception of SYNC$_P$ primitives before the end of the FIS (a SYNC Escape) are not counted in the R_ERR$_P$ ending status counters.

13.7.2.1 Counter Definitions

The counter definitions in this section specify the events that a particular counter identifier represents.

13.7.2.1.1 Identifier 000h

There is no counter associated with identifier 000h. A counter identifier of 000h indicates that there are no additional counters in the log page.

13.7.2.1.2 Identifier 001h

The counter with identifier 001h returns the number of commands that returned an ending status with the ERR bit set to one in the Status register and the ICRC bit set to one in the Error register.

13.7.2.1.3 Identifier 002h

The counter with identifier 002h returns the sum of (the number of transmitted Device-to-Host Data FISes to which the host responded with R_ERR$_P$) and (the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$). The count returned for identifier 002h is not required to be equal to the sum of the counters with identifiers 003h and 004h.

13.7.2.1.4 Identifier 003h

The counter with identifier 003h returns the number of transmitted Device-to-Host Data FISes to which the host responded with R_ERR$_P$.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 457 of 587

13.7.2.1.5 Identifier 004h

The counter with identifier 004h returns the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$. The count returned for identifier 004h is not required to be equal to the sum of the counters with identifiers 00Fh and 010h.

13.7.2.1.6 Identifier 005h

The counter with identifier 005h returns the sum of (the number of transmitted Device-to-Host non-Data FISes to which the host responded with R_ERR$_P$) and (the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR$_P$). Retries of non-Data FISes are included in this count.

13.7.2.1.7 Identifier 006h

The counter with identifier 006h returns the number of transmitted Device-to-Host non-Data FISes to which the host responded with R_ERR$_P$. Retries of non-Data FISes are included in this count.

13.7.2.1.8 Identifier 007h

The counter with identifier 007h returns the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR$_P$. Retries of non-Data FISes are included in this count.

13.7.2.1.9 Identifier 008h

The counter with identifier 008h returns the number of transmitted Device-to-Host non-Data FISes which were retried after which the host responded with R_ERR$_P$.

13.7.2.1.10 Identifier 009h

The counter with identifier 009h returns the number of times the device transitioned into the PHYRDYn state from the PHYRDY state, including but not limited to asynchronous signal events, power management events, and COMRESET events. If interface power management is enabled, then this counter may be incremented due to interface power management transitions.

13.7.2.1.11 Identifier 00Ah

The counter with identifier 00Ah returns the number of transmitted Device-to-Host Register FISes with the device reset signature in response to a COMRESET, which were successfully followed by an R_OK$_P$ from the host.

13.7.2.1.12 Identifier 00Bh

The counter with identifier 00Bh returns the number of received Host-to-Device FISes of all types (Data and non-Data) to which the device responded with R_ERR$_P$ due to CRC error. The count returned for identifier 00Bh is not required to be equal to the sum of the counters with identifiers 00Fh and 012h.

13.7.2.1.13 Identifier 00Dh

The counter with identifier 00Dh returns the number of received Host-to-Device FISes of all types (Data and non-Data) to which the devices responded with R_ERR$_P$ for reasons other than CRC error. The count returned for identifier 00Dh is not required to be equal to the sum of the counters with identifiers 010h and 013h.

13.7.2.1.14 Identifier 00Fh

The counter with identifier 00Fh returns the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$ due to CRC error.

13.7.2.1.15 Identifier 010h

The counter with identifier 010h returns the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$ for reasons other than CRC error.

13.7.2.1.16 Identifier 012h

The counter with identifier 012h returns the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR$_P$ due to CRC error.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 458 of 587

13.7.2.1.17 Identifier 013h

The counter with identifier 013h returns the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR_P for reasons other than CRC error.

13.7.3 READ LOG EXT Log Page 11h

READ LOG EXT log page 11h is one page (512 bytes) in length. The first Dword of the log page contains information that applies to the rest of the log page. Software should continue to process counters until a counter identifier with value 0h is found or the entire page has been read. A counter identifier with value 0h indicates that the log page contains no more counter values past that point. Log page 11h is defined in Figure 182.

Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							
3	Reserved							
…	…							
n	Counter n Identifier							
n+1								
n+2	Counter n Value							
n+ Counter n Length								
…	…							
508	Reserved							
509								
510								
511	Data Structure Checksum							

Figure 182 – READ LOG EXT Log Page 11h data structure definition

Counter n Identifier

Phy event counter identifier that corresponds to Counter n Value. Specifies the particular event counter that is being reported. The Identifier is 16 bits in length. Valid identifiers are listed in Figure 181.

Counter n Value

Value of the Phy event counter that corresponds to Counter n Identifier. The number of significant bits is determined by Counter n Identifier bits 14:12 (as defined in section 13.7.2). The length of Counter n Value shall always be a multiple of 16-bits. All counters are one-extended. For example, if a counter is only physically implemented as 8-bits when it reaches the maximum value of FFh, it shall be one-extended to FFFFh. The counter shall stop (and not wrap to zero) after reaching its maximum value.

Counter n Length

Size of the Phy event counter as defined by bits 14:12 of Counter n Identifier. The size of the Phy event counter shall be a multiple of 16-bits.

Data Structure Checksum

The data structure checksum is the 2's complement of the sum of the first 511 bytes in the data structure. Each byte shall be added with unsigned arithmetic and overflow shall be ignored. The sum of all 512 bytes of the data structure is zero when the checksum is correct.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

Reserved All reserved fields shall be cleared to zero

13.8 Staggered Spin-up (Optional)

Storage subsystems that include numerous Serial ATA hard disk drives are presented with power system design issues related to the current load presented during system power-up. It is desirable to provide a simple mechanism by which the storage subsystem controller(s) can sequence disk drive initialization and spin up. Note that Serial ATA disk drive vendors may not always provide the capability to parse or execute ATA commands prior to spinning up a drive and completing drive initialization, therefore this mechanism may not rely on the ATA protocol.

In order to accommodate staggered spin-up of an array of disk drives in an enclosure, disk drives shall not spin up until after successful Phy initialization, that is after the Phy enters the DP7:DR_Ready state. Any of a number of methods may be used by the disk drive to defer spin-up prior to Phy initialization and to maintain correct interface status during drive initialization.

Storage subsystem controllers may employ a variety of methods to sequence Phy initialization across their plurality of Serial ATA ports, including but not limited to staged release of chip-level resets of host-side Serial ATA transceivers, or embedded advanced power management logic.

System implementations should comprehend the various scenarios that may require power management, and the corresponding Phy initialization sequences. For example, upon power up of a populated storage subsystem, Phy communication is initiated with a COMRESET signal generated by the host-side transceiver. The use of the term “host-side transceiver” here refers to the Serial ATA interface located on the storage subsystem controllers. This contrasts with sequences associated with hot-plugging of a Serial ATA disk drive into an operational storage subsystem, wherein COMINIT signals generated by disk-side transceivers initiate Phy communications. In both of these cases, COMRESET or COMINIT signals are followed by exchange of COMWAKE signals. It is the successful entry into the DP7:DR_Ready state that gates disk drive spin-up.

13.8.1 ATA Power States

The ATA/ATAPI-6 standard describes four power states: Active, Idle, Standby, and Sleep. In both the Standby and Sleep modes, the drive is spun-down. According to the power modes state machine, the drive shall only spin up after entering these power modes when media access is required. The drive shall not use the staggered spin-up mechanism to cause the drive to spin-up in cases where the drive was spun down using an ATA command.

13.9 Non-512 Byte Sector Size (Informative)

ATA disk drives today generally support 512-byte sector sizes exclusively. The Serial ATA interface has no inherent sector size dependency and there is nothing in the Serial ATA interface specification that precludes sector sizes other than 512 bytes. Some classes of applications have a need for sector sizes of 520 bytes or 528 bytes.

Since there is no inherent sector size sensitivity in the Serial ATA interface, the command set extensions for supporting the reporting of sector sizes other than 512 bytes is relegated to the INCITS T13 committee for resolution.

Regardless of the physical sector size of storage devices, the maximum Data FIS payload length is 2048 Dword (8KB) as defined in section 10.3.11. This may imply that either the number of physical sectors that are encompassed by a single Data FIS is reduced for devices with larger sector sizes or that Data FISes convey a non-integer number of sectors of data being transferred.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 460 of 587

13.10 Defect Management (Informative)

13.10.1 Overview (Informative)

Storage subsystems that have been based on SCSI disk drives have evolved processes to address disk drive failure and mitigate effects of disk defects. For example, SCSI commands such as READ DEFECT DATA (37h) and REASSIGN BLOCKS (07h) have been used to allow storage administrators or applications to proactively address problematic sectors on a disk surface.

Serial ATA drives leverage the economies of the desktop drive market, and as a consequence inherit both the design philosophy and implementation of desktop drives. From a design philosophy perspective, desktop drives have never yielded the low-level of control, such as reassignment of logical blocks that is commonplace in enterprise-class drives. Instead, desktop drives have been positioned as a "black-box" data repository. This approach has many benefits, such as elimination of defect management as a design task for the computer (O/S, file system, I/O, device driver) designer, at the expense in some cases of deterministic drive performance and awareness of defect management activity in general.

As "black box" providers, Serial ATA drive vendors assume the bulk of the responsibility for defect management and data availability. This section provides a high-level overview of approaches that Serial ATA drive vendors employ in these areas, and of the tools that are available to system designers, storage management application designers and system administrators to maximize storage and data dependability.

In summary, Serial ATA drives provide similar information via SMART (Self-Monitoring, Analysis and Reporting Technology) commands as the information returned with SCSI READ DEFECT DATA commands. Subsystem designers make no provisions for reassignment of blocks; rather reassignment of defective blocks is performed automatically when indicated by desktop-class drives. Additionally, disk drive manufacturers provide a spectrum of tools and embedded features that help prevent occurrence of non-recoverable read errors, that help detect disk drive degradation that might cause catastrophic loss of data, and that help administrators diagnose and isolate the cause of error events.

13.10.2 Typical Serial ATA Reliability Metrics (Informative)

Various methods are employed by disk manufacturers to recover bad bits in a block of data. First and foremost, extensive error correcting codes (ECC) are used "on-the-fly" to detect and correct errors without impacting drive performance. Second, various read-retry schemes may be used when ECC fails to correct an error on the fly. Only after retry processes are exhausted are errors posted to the host.

Design Recommendation: Accommodate inherent non-recoverable read error rate through RAID schemes appropriate for the target market's reliability needs.

13.10.3 An Overview of Serial ATA Defect Management (Informative)

Defects that cause non-recoverable read errors come in two distinct flavors, temporary and permanent. Various schemes are employed by drive manufacturers to determine the nature of a defect. If a defect is determined to be of a permanent nature, drive firmware re-maps the LBA to a predefined spare sector, and marks the defective sector as such. Subsequent accesses to the re-mapped LBA are directed to the spare sector in a fashion transparent to the host. As part of the re-mapping process, the drive preserves the error state of the remapped block until such time as it is written with new data.

The number of spare sectors configured in a disk drive is generally unique across different drive vendors and drive models and reflects tradeoffs between drive capacity and expected defect rates. Generally, spares are associated with defined allocation groups in a disk drive. If the

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 461 of 587

number of spare sectors becomes exhausted over time, subsequent permanent defects result in sectors that cannot be re-mapped. The affected Logical Block Addresses (LBAs) are then recognized as bad by the host operating system or disk utilities such as “scandisk”, and are subsequently not used.

Design Recommendation: If a host or a Serial ATA storage subsystem encounters a non-recoverable read error, that error is managed by the disk drive. The disk drive is responsible for performing extensive read retry processes prior to communicating an error condition, exercising internal drive diagnostics to determine the nature of the error, and re-map the physical sector if required. Subsequent accesses to that logical address are directed to a known good region on the disk.

Care should be taken to assure that spares are not exhausted if write caching is enabled. If this care is not taken, there is a possibility that a write operation may not actually complete successfully and return the appropriate error condition to the host in a timely fashion. Refer to the information on SMART in section 13.10.5 for monitoring spares status.

In especially critical data applications, specific queries of disk drive logs may be performed to determine whether the error event is a random event, or if there exists some degrading condition that warrants additional system or administrator action.

In storage subsystems where known good data can be recovered from redundant sources (such as in a RAID subsystem), it is recommended that known good data be written back to any LBA for which a read error is reported. This behavior ensures that the disk drive has an opportunity to remedy the error condition and write known good data to known good blocks on disk, whether those be remapped blocks resulting from a permanent error condition, or the same blocks that may have been affected by an error of a temporary nature. Subsequent read accesses are assured of being satisfied without error.

13.10.4 Continuous Background Defect Scanning (Informative)

Serial ATA drive vendors may employ schemes called continuous background defect scanning (CBDS), where during idle periods, firmware routines are used to scan sectors on the disk to look for and correct defects. CBDS therefore executes as a background task. The effect of CBDS is to reduce the occurrence of non-recoverable read errors by proactively “cleaning” defects from good sectors or copying data from suspect regions on disk to spare sectors. CBDS is described in more detail in the SMART specification which also provides a means for enabling/disabling the capability.

Design Recommendation: Consider CBDS requirements when evaluating system power-saving schemes that might power-off disk drives during apparent periods of inactivity.

13.10.5 Self-Monitoring, Analysis and Reporting Technology (Informative)

Some disk failures are predictable. Such failure mechanisms are characterized by degradation over time, and in some cases may be effectively monitored by the disk drive and logged for periodic reporting to storage managers or management utilities.

The ATA/ATAPI-6 standard describes SMART command support in detail. Individual drive vendors provide unique SMART capabilities on their drives and documentation on those capabilities so that host or controller developers can effectively use predictive failure information

Design Recommendation: Use capabilities provided through ATA SMART commands to predict disk failure when possible. Predictive knowledge can be used to swap in spare disk drives in RAID configurations, to trigger data backup or protection routines, or to schedule storage subsystem maintenance.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 462 of 587

13.11 Enclosure Services/Management (Optional)

13.11.1 Overview

A means for providing support for industry-standard SAF-TE (SCSI Accessed Fault-Tolerant Enclosures) and SES (SCSI Enclosure Services) enclosure services is provided in order to improve the functionality of Serial ATA storage subsystems. No modification to Serial ATA devices contained within an enclosure is required.

A storage enclosure processor (SEP) is a device which interfaces with various sensors and indicators within a storage enclosure subsystem. A Serial ATA Enclosure Management Bridge (SEMB) is a device that allows a Serial ATA host controller or Port Multiplier to communicate with a SEP, acting as a bridge between the host and the enclosure processor. This section defines a standard interface for software to communicate with SEMB devices via the ATA Command Block Registers. It also defines how SEMB devices communicate with SEP devices using the I^2C IPMI protocol. To allow software to communicate effectively with the SEMB, the SEMB is viewed as another Serial ATA device connected to the storage subsystem. The SEMB may be implemented within a Serial ATA host controller, within a Port Multiplier, or as a hardware bridge between a host and the SEP.

In this specification, the host (the Serial ATA RAID controller or HBA) may use either the SAF-TE or SES command protocol to communicate control/status with the SEP. These protocols provide the necessary features and have the advantage of being well known and widely implemented, and should therefore minimize the impact on RAID controller firmware and host management software. The implementation requires that Serial ATA host controllers or Port Multipliers that support enclosure management have an I^2C interface to communicate with the SEP device.

This specification also addresses the need to support a generic and standardized interface that allows application software from different vendors to communicate with the management device. This is implemented by having the appropriate commands be sent/received using the ATA Command Block Register set (or Serial ATA Register FIS) using the READ SEP/WRITE SEP commands associated with this interface. The ATA Command Block Register interface allows for consistency with the other devices in the subsystem as well as being simple to use and well understood.

13.11.2 Topology

The enclosure services support mechanisms should support the configurations and topologies expected for storage subsystems that require such enclosure services (such as external storage enclosures). Figure 183 illustrates a generic configuration.

The concentrator is controller logic that bridges a host interface to one or more Serial ATA devices. Typically a concentrator would be a RAID controller or a Port Multiplier, but it may be a simple HBA or part of an integrated multi-function chipset. Because a concentrator may be embodied differently depending on implementation, the generic term is used in order to avoid implying any particular implementation.

The host interface is the interface through which the host communicates with the concentrator. For RAID controllers plugged into a PCI slot, the host interface would be PCI, while for external RAID controllers in a storage subsystem, the host interface might be Fibre Channel, InfiniBand Architecture, Ethernet (iSCSI) or any of a number of different external subsystem interconnects.
The SEMB (Serial ATA enclosure management bridge) is logic that bridges enclosure management data from a host interface to an enclosure management bus (indicated as I^2C in the figure). For an intelligent PCI RAID controller, the SEMB may be firmware running on the RAID processor and associated design-specific I^2C controller interface logic, while for a Port Multiplier, the SEMB would be controller logic that bridges the I^2C interface (and transactions) to Serial ATA via a logical ATA Command Block Register interface.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 463 of 587

The SEP (storage enclosure processor) interfaces with the various sensors and indicators in the enclosure such as temperature sensors, fan tachometers, and indicator LEDs.

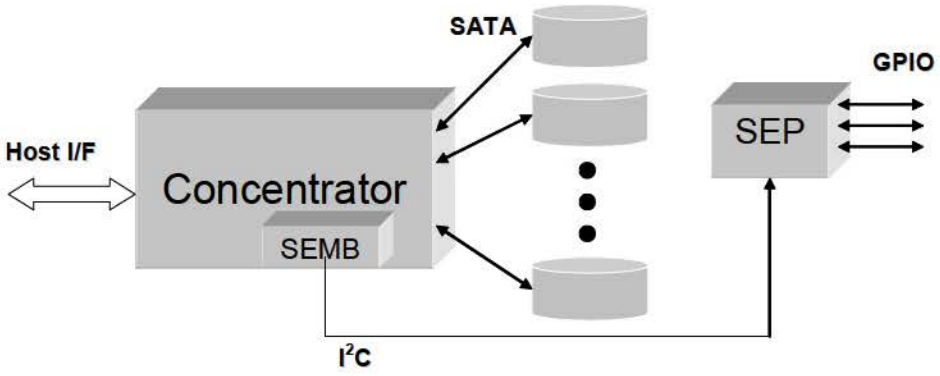


Figure 183 – Generic enclosure services topology

13.11.2.1 Definition Configuration

The definition configuration is distilled from the generic topology in Figure 183 where the extraneous elements have been removed for the sake of definition clarity. For the definition configuration, the host interface is selected as Serial ATA since this results in the maximum number of interfaces/elements being exposed for definition. In practice, the various elements may be integrated into another subsystem element (for instance, the SEMB might be integrated with the RAID controller or Port Multiplier). For the sake of definition, the elements are shown separately and the solution is presented as if it were a Serial ATA target device. In such a configuration, the SEP is being presented as merely another exposed Serial ATA device. This configuration is most analogous to existing SCSI enclosure services schemes where the enclosure services device is implemented using a SCSI target ID. For this configuration the enclosure services bridge and associated storage enclosure processor are exposed as a single-purpose ATA device.



Figure 184 – Simplified view of generic topology

Since the HBA (host bus adapter) in the previous figure is a standard Serial ATA HBA and is therefore fixed, the elements being defined in this section can be further reduced to the definition configuration in Figure 185 that presents the enclosure services facility as an ATA device with a Command Block Register interface.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 464 of 587

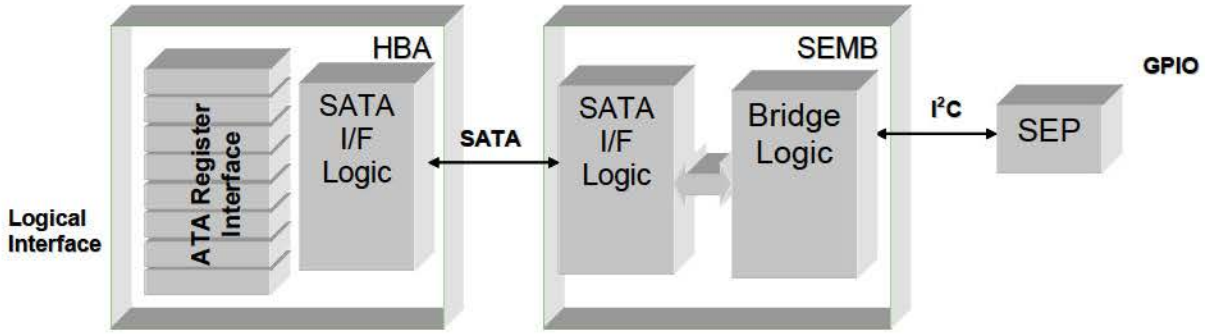


Figure 185 – Enclosure services definition configuration

For implementations where the SEP is not provided packaged with its front-end SEMB or equivalent (i.e. the SEP and SEMB are provided by different vendors), it is recommended that the interconnect between these two elements be I^2C and the command protocol for this interconnect is defined in section 13.11.4.2. For implementations where the SEP is provided by the same supplier as the SEMB, the interconnect between these elements may be vendor-specific (and may be embedded for those cases where the two elements are integrated).

Similarly, for implementations where the SEMB is packaged with its front-end HBA or equivalent, the interconnect between these elements may be vendor specific and is not required to be Serial ATA (in some configurations, the SEMB may be integrated into the HBA in which case the interconnect between these two logical elements is embedded).

13.11.3 Limitations

In order to accommodate a range of possible implementations, the interfaces and elements defined in this section represent more than would typically be utilized by any one design/implementation. Only those interfaces that are exposed and interconnect ingredients from different vendors are expected to utilize the corresponding definitions and specifications described in this section. For example, an intelligent RAID controller may have the SEMB functionality implemented as firmware running on the embedded RAID processor, and such a solution would not expose the SEMB front-end interface, which would be vendor-specific (i.e internal to the RAID card and managed by the vendor-supplied firmware). However, such a solution probably would expose the I^2C interface if it interconnects with another vendor's SEP, and would therefore need to comply with the specification for the communications between the SEMB and the SEP.

13.11.4 Definition

This specification supports the same enclosure management command/status as the SAF-TE protocol (plus addendums). Since SAF-TE is widely used in current SCSI applications, it is a natural path to try to reuse as much of the data format as possible. This specification also supports the SES enclosure management command/status as defined in the SCSI-3 Enclosure Services Command Set specification. See section 3 for specific references.

For the case of the Serial ATA Register set, a subset of the existing ATA task file is used for creating the commands to send/receive data from the SEP device. Where possible, the same command structure as the existing ATA protocol has been used. All SEP commands are issued to the SEMB with the SEP_ATTN opcode in the Command register and the actual SEP command is passed as a parameter in Features register. Section 13.11.4.3 and 13.11.4.4 define the host-to-SEP and SEP-to-host command protocols.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

13.11.4.1 Discovery

Following a hardware reset or software reset, the SEMB shall place the unique SEMB-specific signature as identified in Figure 186 into the logical Command Block Registers if the SEMB detects the presence of an attached SEP. The signature shall be available in the logical Command Block Registers no later than 3 seconds after a reset operation (whether power-on reset, hard reset, or soft reset). For implementations where the SEMB is separate from the HBA, the Command Block Register signature shall be conveyed over the Serial ATA interconnect using a Register FIS device-to-host.

Register	7	6	5	4	3	2	1	0
Error	00h							
Sector Count	01h							
Sector Count (exp)	00h							
LBA Low	01h							
LBA Low (exp)	00h							
LBA Mid	3Ch							
LBA Mid (exp)	00h							
LBA High	C3h							
LBA High (exp)	00h							
Device	00h							
Status	BSY	DRDY	DF	DSC	DRQ	0	0	ERR

Status =50h

Figure 186 – Register Signature Indicating Presence of Enclosure Services Device

If the SEMB does not detect the presence of an attached SEP, then the SEMB shall place the signature as identified in Figure 187 into the logical Command Block Registers in order to convey to the host that there is no device present at the logical interface, and the SEMB shall not respond to any subsequent Command Block Register accesses or issued commands until the next hardware reset or software reset. For implementations where the SEMB is separate from the HBA, the Command Block Register signature is conveyed over the Serial ATA interconnect using a Register FIS device-to-host.

Register	7	6	5	4	3	2	1	0
Error	FFh							
Sector Count	FFh							
Sector Count (exp)	FFh							
LBA Low	FFh							
LBA Low (exp)	FFh							
LBA Mid	FFh							
LBA Mid (exp)	FFh							
LBA High	FFh							
LBA High (exp)	FFh							
Device	FFh							
Status	0	1	1	1	1	1	1	1

Status=7Fh

Figure 187 – Register Signature for Absent Enclosure Processor

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 466 of 587

In addition to the device reset signature, SEPs shall support the IDENTIFY SEP command described in section 13.11.5.1 in order to allow host software to determine its capabilities and to identify whether it supports the SAF-TE or SES command set.

13.11.4.2 Logical Command Block Registers to I^2C Mapping

The SEMB provides the mapping and translation from transactions between the SEP and SEMB, and the transactions presented to the host via the logical Command Block Register interface.

13.11.4.2.1 Command Delivery

Commands are delivered to the SEP by the SEMB via the Command Block Register interface as a result of the Command or Device Control register being written (if the SEMB logical register interface is closely coupled to the host/HBA) or in response to receipt of a Register FIS (if the SEMB logical register interface is at the far end of a Serial ATA physical interconnect).

Commands are delivered to the SEP over I^2C by the SEMB which extracts the SEP command and command type fields from the Command Block Registers (or received Register FIS) and forwards the fields to the SEP. If the SEP is discrete and connected via an I^2C interconnect, the SEP command fields are packaged as an I^2C frame and forwarded to the SEP via the I^2C interconnect. The SEMB logical Command Block Registers observes the same conventions as Serial ATA for handling of the BSY bit. The BSY bit is set by the interface/SEMB in response to the Command register being written, and the SEP later clears this bit to indicate command completion/status.

For issuing commands, the SEMB shall generate and transmit an I^2C packet based on the contents of the Command Block Registers or Register FIS. The mapping of Command Block Registers to transmitted I^2C packet shall only be done for commands written using the SEP_ATTN opcode, and the SEMB need not respond to any other opcode in the Command register. The Command Block Registers mapping to I^2C packet shall be as indicated in Figure 188.

Register	7	6	5	4	3	2	1	0
Features	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
LBA Low	CMD_TYPE							
LBA Low (exp)	Reserved							
LBA Mid	Reserved							
LBA Mid (exp)	Reserved							
LBA High	Reserved							
LBA High (exp)	Reserved							
Device	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 188 – Command Block Register Fields Used in Enclosure Processor Communications

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

SEP_CMD The SAF-TE or SES command code to be issued in conjunction with the command type specified in CMD_TYPE.

SAF-TE READ BUFFER usage: SEP_CMD is equivalent to the BUFFER ID field of the SCSI READ BUFFER command. See section 3.1 in the SAF-TE specification reference for the command codes and their functions.

SAF-TE WRITE BUFFER usage: SEP_CMD is equivalent OPERATION CODE field transferred in the parameter data of the SCSI WRITE BUFFER command. See section 3.2 in the SAF-TE specification reference for the command codes and their functions.

SES RECEIVE DIAGNOSTIC RESULTS usage: SEP_CMD is equivalent to the PAGE CODE field of the SCSI RECEIVE DIAGNOSTIC RESULTS command. See section 6.1 of the SES specification reference for the command codes and their functions.

SES SEND DIAGNOSTIC usage: SEP_CMD is equivalent to the PAGE CODE field transferred in the parameter data of the SCSI SEND DIAGNOSTIC command. See section 6.1 of the SES specification reference for the command codes and their functions.

IDENTIFY SEP usage: SEP_CMD is equal to ECh.

LEN The transfer length of the data transfer phase of the command in Dword units. Valid values are 1-255 (yielding a maximum transfer length of 1020 bytes). Data transfers that are not a multiple of 4 bytes shall be padded by the transmitter with zeros to the next 4-byte (Dword) granularity.

CMD_TYPE
Flag indicating whether the issued SEP command is a SAF-TE command code or a SES command code and whether the data transfer direction is from SEP-to-host or host-to-SEP. The encoding of the field is as follows:

00h SAF-TE command code with SEP-to-host data transfer, including IDENTIFY SEP

80h SAF-TE command code with host-to-SEP data transfer

02h SES command code with SEP-to-host data transfer, including IDENTIFY SEP

82h SES command code with host-to-SEP data transfer

All other values reserved

The resulting I^2C frame for delivering a command is as indicated in Figure 189.

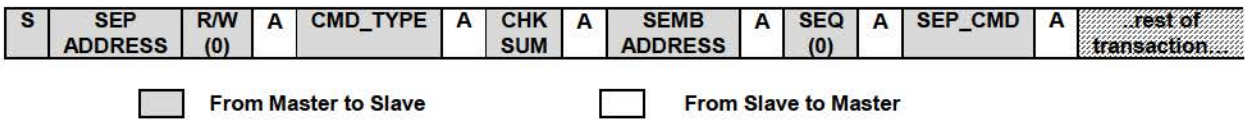


Figure 189 – I^2C Frame for Conveying an Enclosure Services Command

The SEMB need not support any Command register writes with values other than SEP_ATTN. In response to a Command register write of a value other than SEP_ATTN, the SEMB shall set the ERR bit and clear the BSY bit in the Status register. In response to such illegal host behavior, the SEMB shall not generate any I^2C traffic for that illegal command. The SEMB shall support Device

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 468 of 587

Control register writes where the state of the SRST bit changes, but shall take no action in response to Device Control register writes where SRST does not change state.

The SEP need not support both SAF-TE and SES command protocols. In response to a SEP command issues with a protocol not supported by the SEP, the SEP shall return with error status as defined in section 13.11.4.2.2.

13.11.4.2.2 Status Mechanism

The SEMB indicates command completion to the host by clearing the BSY bit in the Status register and by triggering an interrupt. SEP status returns to the SEMB consist only of the Status byte and no other Command Block Registers are used to convey status. If the SEP has encountered some error condition or does not support the issued SEP command, then the ERR bit in the Status register is set to one.

If the SEP is communicating to the SEMB over an I^2C interconnect, then the status byte is included at the end of the transactions as indicated in the SEP read and write definitions that follow. Upon transferring the status value into the Status register, the SEMB shall clear BSY and signal an interrupt. If the SEMB is connected to the host via a Serial ATA interconnect, then the ending status shall be collected in a Register FIS and transmitted to the host. In response to the read and write SEP commands, the SEP_CMD and CMD_TYPE byte values are returned.

Upon successful completion of a command, the status value in the Status register shall be 50h. Upon an error condition, the status value shall be 51h.

13.11.4.3 Host-to-SEP Data Commands

All host-to-SEP data transfers transfer a data payload with length as indicated in the LEN field for the command. The SAF-TE and SES references define the commands and functions supported as well as the format of the transferred data structures.

All SEP commands are issued using the SEP_ATTN command (opcode 67h) in the Command register and the SEP command code in the Features register as illustrated in the Command Block Registers image of Figure 190. The CMD_TYPE field identifies whether the issued SEP command is a read or a write and whether the command protocol is SAF-TE or SES.

Register	7	6	5	4	3	2	1	0
Features	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
LBA Low	CMD_TYPE (80h or 82h)							
LBA Low (exp)	Reserved							
LBA Mid	Reserved							
LBA Mid (exp)	Reserved							
LBA High	Reserved							
LBA High (exp)	Reserved							
Device	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 190 – WRITE SEP Command Block Registers

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

Host-to-SEP data transfer commands shall be followed by a data transfer from the host to complete the SEP command delivery. If the command is delivered to the SEP over an I^2C interface, the transmitted I^2C packets shall be of the form indicated in Figure 191.

SEMB (master) to SEP (slave) transfer – transfers both the command and data

S	SEP ADDRESS	R/W (0)	A	CMD_TYPE	A	CHK SUM	A	SEMB ADDRESS	A	SEQ (0)	A	SEP_CMD	A	...

...	Data[0]	A	Data[1]	A	...	Data[LEN*4-1]	A	CHKSUM	A	P

SEP (master) to SEMB (slave) transfer – transfers status

S	SEMB ADDRESS	R/W (0)	A	CMD_ TYPE	A	CHK SUM	A	SEP ADDRESS	A	SEQ (0)	A	SEP_ CMD	A	SEP STATUS	A	CHK SUM	A	P

From Master to Slave From Slave to Master

Figure 191 – I^2C Transactions Corresponding to a WRITE SEP Command

The I^2C transport for Serial ATA enclosure service traffic shall be Master to Slave (IPMB) compliant. See the IPMB and I^2C references for details on Master to Slave transport conventions for IPMB.

The host shall use the DMA protocol for transferring data from the host to the SEMB, and if the interface between the SEMB and the host is a Serial ATA interface, the SEMB shall trigger the DMA data transfer by transmitting a DMA Activate FIS to the host.

13.11.4.4 SEP-to-Host Data Commands

All SEP-to-Host data transfers transfer a data payload. The LEN field for the command indicates the length of the data payload being transferred. The SAF-TE and SES references define the commands and functions supported as well as the format of the transferred data structures. Both SAF-TE and SES SEPs shall support the IDENTIFY SEP command, which has the same format for both command sets.

All SEP commands are issued using the SEP_ATTN command (opcode 67h) in the Command register and the SEP command code in the Features register as illustrated in the Command Block Registers image of Figure 192. The CMD_TYPE field identifies whether the issued SEP command is a read or a write and whether the command protocol is SAF-TE or SES.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 470 of 587

Register	7	6	5	4	3	2	1	0
Features	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
LBA Low	CMD_TYPE (00h or 02h)							
LBA Low (exp)	Reserved							
LBA Mid	Reserved							
LBA Mid (exp)	Reserved							
LBA High	Reserved							
LBA High (exp)	Reserved							
Device	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 192 – READ SEP Command Block Registers

SEP-to-host data transfer commands shall be followed by a subsequent data transfer from the device to the host to complete the SEP command. If the command is delivered to the SEP over an I^2C interface, the transmitted I^2C packets shall be of the form indicated in Figure 193.

SEMB (master) to SEP (slave) transfer – transfer the command

S	SEP ADDRESS	R/W (0)	A	CMD_ TYPE	A	CHK SUM	A	SEMB ADDRESS	A	SEQ (0)	A	SEP_C MD	A	CHK SUM	A	P

SEP (master) to SEMB (slave) transfer – transfers both the data and status

S	SEMB ADDRESS	R/W (0)	A	CMD_TYPE	A	CHK SUM	A	SEP ADDRESS	A	SEQ (0)	A	SEP_CMD	A	...

...	SEP STATUS	A	Data[0]	A	Data[1]	A	...	Data[LEN*4-1]	A	CHKSUM	A	P

From Master to Slave From Slave to Master

Figure 193 – I^2C Transactions Corresponding to READ SEP Command

The host shall use the DMA protocol for transferring data from the SEMB to the host.

13.11.5 SES and SAF-TE Extensions

13.11.5.1 IDENTIFY SEP Command (ECh)

Both SAF-TE and SES SEPs shall support the IDENTIFY SEP command as defined here. The IDENTIFY SEP command is a SEP-to-host command code used as the SEP command argument for the SEP_ATTN command with the CMD_TYPE value set to either 00h or 02h depending on the command protocol being used (see section 13.11.4.2.1). The command returns a data structure that describes the capabilities and attributes of the attached SEP. The IDENTIFY SEP command requests that the SEP return enclosure specific information (not device or environmental information). The command is roughly analogous to a SCSI INQUIRY command.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 471 of 587

For parameters defined as a string of ASCII characters, the ASCII data fields shall contain only graphic codes (i.e., code values 20h through 7Eh) and all strings shall be padded with space characters to the full width of the field. For the string “Copyright”, the character “C” is the first byte, the character “o” is the second byte, etc.

13.11.5.1.1 IDENTIFY SEP Data Structure

Figure 194 describes the data structure that is returned by the SEP in response to the IDENTIFY_SEP command. All reserved fields shall be cleared to zero. By setting the LEN field of the issued Read SEP command, the amount of returned data can be controlled (i.e. the transfer time can be reduced by not transferring the reserved and vendor specific bytes at the end of the data structure).

The IDENTIFY SEP data structure is normally 64 bytes long and may be extended in order to support larger VENDOR SPECIFIC ENCLOSURE INFORMATION.

The data structure does not include a list of elements in an enclosure.

This data block provides enclosure descriptor information and parameters.

Identify Data Page

Bytes	Field name
0	ENCLOSURE DESCRIPTOR LENGTH
1	SUB-ENCLOSURE IDENTIFIER
2-9	ENCLOSURE LOGICAL IDENTIFIER
10-17	ENCLOSURE VENDOR IDENTIFICATION
18-33	PRODUCT IDENTIFICATION
34-37	PRODUCT REVISION LEVEL
38	CHANNEL IDENTIFIER
39-42	FIRMWARE REVISION LEVEL
43-48	INTERFACE IDENTIFICATION STRING
49-52	INTERFACE SPECIFICATION REVISION LEVEL
53-63	VENDOR SPECIFIC ENCLOSURE INFORMATION

Figure 194 – IDENTIFY SEP data structure definition

ENCLOSURE DESCRIPTOR LENGTH

The ENCLOSURE DESCRIPTOR LENGTH field specifies the number of valid bytes contained in the IDENTIFY SEP data structure.

SUB-ENCLOSURE IDENTIFIER

As defined in the SES reference. Unless sub-enclosures are defined this field shall be cleared to zero.

ENCLOSURE LOGICAL IDENTIFIER

The ENCLOSURE LOGICAL IDENTIFIER field contains a unique logical identifier for the subenclosure. It shall use an 8-byte NAA identifier, the format of which is defined in the SCSI Primary Commands reference. The ENCLOSURE LOGICAL IDENTIFIER is unique to the enclosure and may be different from the worldwide name of the device providing the enclosure services. The combination of this field, along with the ENCLOSURE VENDOR IDENTIFICATION and PRODUCT IDENTIFICATION fields, uniquely identifies any SEP unit from any

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 472 of 587

manufacturer. The worldwide name should have an NAA field of 5h, indicating that IEEE is the naming authority.

ENCLOSURE VENDOR IDENTIFICATION

The ENCLOSURE VENDOR IDENTIFICATION field shall contain the identification string for the vendor of the enclosure in the same format as specified for the vendor identification field of the standard SCSI INQUIRY data (see SCSI Primary Commands reference). The ENCLOSURE VENDOR IDENTIFICATION may be different from the vendor identification of the device providing the enclosure services.

PRODUCT IDENTIFICATION

The PRODUCT IDENTIFICATION field shall contain the product identification string for the enclosure in the same format as specified for the product identification field of the standard SCSI INQUIRY data (see SCSI Primary Commands reference). The PRODUCT IDENTIFICATION field may be different from the product identification of the device providing the enclosure services.

PRODUCT REVISION LEVEL

The PRODUCT REVISION LEVEL field shall contain the product revision level string for the enclosure in the same format as specified for the product revision level field of the standard SCSI INQUIRY data (see SCSI Primary Commands reference). The PRODUCT REVISION LEVEL may be different from the product revision level of the device providing the enclosure services.

CHANNEL IDENTIFIER

The CHANNEL IDENTIFIER field is used to distinguish between separate HBA channels supported by a single enclosure (through multiple/redundant host connections for example). The value in this field is unique for each channel. This field is optional and if not used shall be cleared to zero.

FIRMWARE REVISION LEVEL

The FIRMWARE REVISION LEVEL field is a 4-byte ASCII string that identifies the current firmware revision of the SEP device.

INTERFACE IDENTIFICATION STRING

The INTERFACE IDENTIFICATION STRING field is a 6-byte field that holds the constant ASCII string “SAF-TE” (if the command is issued with the SAF-TE protocol bit set in the CMD_TYPE field and the SEP supports this protocol) or “S-E-S” (if the command is issued with the SES protocol bit set in the CMD_TYPE field and the SEP supports this protocol). This serves to identify that the enclosure is compliant with the command protocol indicated by the CMD_TYPE field used in issuing the IDENTIFY SEP command.

INTERFACE SPECIFICATION REVISION LEVEL

The INTERFACE SPECIFICATION REVISION LEVEL field is a 4-byte field that holds an ASCII string of the format “x.xx”, which identifies the revision of the Interface Specification to which this SEP device claims compliance. ASCII string data is stored with the most significant (leftmost) character stored at the lowest byte offset of the field.

VENDOR-SPECIFIC ENCLOSURE INFORMATION

The VENDOR-SPECIFIC ENCLOSURE INFORMATION field is available for vendor-specific definition and use.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 473 of 587

13.11.5.2 Activity LED Control

The SES and SAF-TE protocols provide commands that are used to inform the SEP device of the state of each of its associated slots and the devices potentially inserted. This information is used to drive the enclosure status signals (LEDs, LCD, audible alarm, etc.) to some meaningful state, or to force the SEP to respond with a preprogrammed response as required; depending on the vendor's implementation.

Since Serial ATA devices do not necessarily provide an activity indication, extensions to the SAF-TE and SES facilities are defined in section 13.11.5.2.1 and 13.11.5.2.2 to accommodate a means by which an enclosure processor can be used to provide operator activity indication.

13.11.5.2.1 SAF-TE - Write Device Slot Status Modification

The length of the valid data for the SAF-TE Write Device Slot Status depends on the number of device slots *(*d) on this channel. There are three bytes of data for each drive slot on the channel and the associated data structure is defined in Figure 195.

Bit/Byte	7	6	5	4	3	2	1	0
0	Slot 0 Byte 0							
1	Slot 0 Byte 1							
2	Slot 0 Byte 2							
…								
d*3-3	Slot d-1 Byte 0							
d*3-2	Slot d-1 Byte 1							
d*3-1	Slot d-1 Byte 2							
d*3	Vendor Specific							
…63								

Figure 195 – SAF-TE Write Device Slot Status data structure

The Serial ATA modification to the definition of the fields is as follows:

Slot *d* Byte 0		As defined in SAF-TE
Slot *d* Byte 1		Modified from definition in SAF-TE
	Bit 0-1	As defined in SAF-TE
	Bit 2	DR_ACT: Set to one by the host to indicate device activity when issuing commands to the device associated with this slot. (Defined as Reserved in the SAF-TE reference)
	Bit 3-7	As defined in SAF-TE
Slot *d* Byte 2		As defined in SAF-TE

If no flags are set in any byte for a device slot this is a NO CHANGE FROM CURRENT STATE indication. This allows a Host to change the state of one particular device slot without having to be aware of the current state of all device slots. Setting one or more flags requires that all flags be written to the correct binary value. Thus, changes should be preceded by a read of the corresponding device slot status and the Write Device Slot Status implemented as a "read-modify-write" operation.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 474 of 587

13.11.5.2.2 SES- Device Element Definition Modification

The format of the CONTROL INFORMATION field for a device element type in the enclosure control page is defined in Figure 196. The data structure is modified with the addition of the drive activity control bit (DR_ACT) to bit 7 of byte 2 (this bit is defined as Reserved in the SES reference).

Bits / Bytes	7	6	5	4	3	2	1	0
0	COMMON CONTROL							
1	Reserved							
2	DR_ACT	DO NOT REMOVE	Reserved		RQST INSERT	RQST REMOVE	RQST IDENT	Rsrvd
3	Reserved		RQST FAULT	DEVICE OFF	ENABLE BYP A	ENABLE BYP B	Reserved	

Figure 196 – SES Device Element data structure

13.11.5.2.3 Activity Indication Behavior and Operation

The host writes a 1 to the DR_ACT bit to set the external DRIVE ACTIVITY LED to 'on'. In response, the SEP shall blink the associated LED at a vendor-specific rate. The LED shall blink for a duration of approximately 0.5 seconds after which this bit shall be automatically cleared to zero by the SEP (and the LED stops blinking).

13.11.5.3 Slot – to – Port Correspondence

For storage subsystems that do not have a direct one-to-one correspondence between host connection and device slot, a correspondence convention is required in order to ensure the host has a means for accurately controlling the proper enclosure slot for a particular storage device. Configurations that do not have direct correspondence include those that have intervening elements between the host and the device that compromise direct correspondence such as would be the case if Serial ATA Port Multipliers were used.

Figure 197 illustrates a configuration where there is no direct correspondence between host connection and enclosure device slot.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 475 of 587

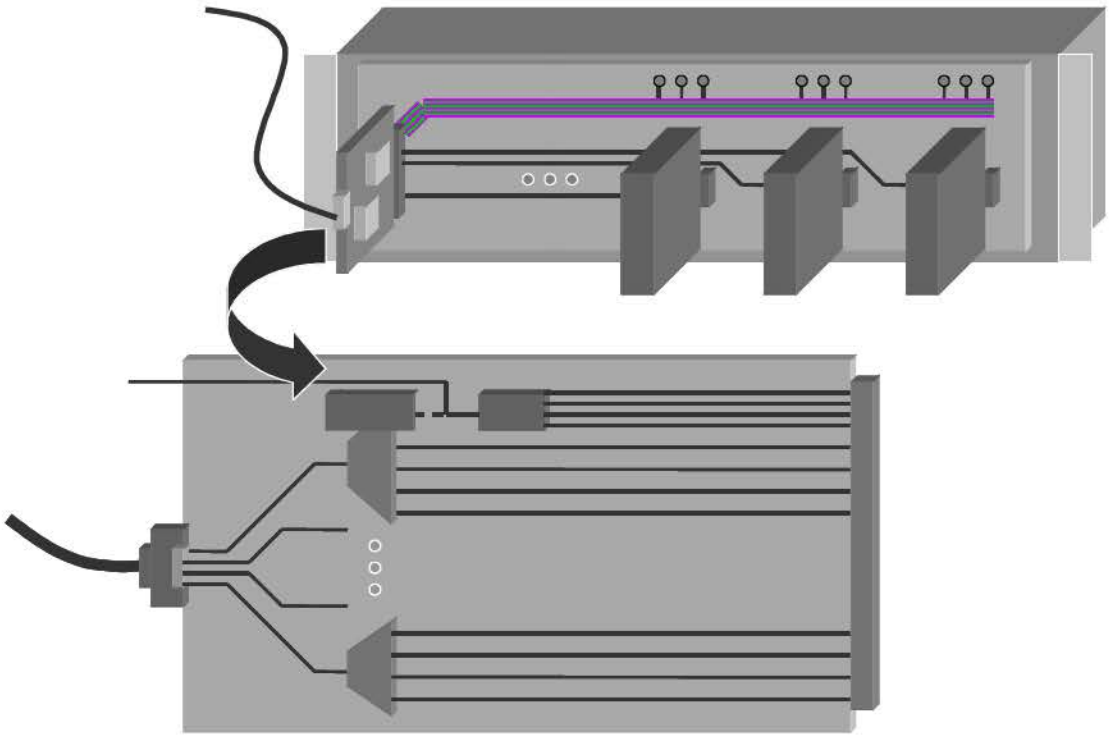


Figure 197 – Example Subsystem

13.11.5.3.1 SAF-TE Correspondence Definition (Optional)

Figure 198 defines the SAF-TE SCSI ID field convention for establishing correspondence between host connections and enclosure slots. SEPs that support SAF-TE may optionally adopt this convention. If an enclosure complies with this correspondence definition, it shall set bit 0 of byte 7 in the data returned for the Read Enclosure Configuration command to '1'. If an enclosure does not comply with this correspondence definition, it shall clear bit 0 of byte 7 in the data returned for the Read Enclosure Configuration command to '0'.

This correspondence convention, if implemented, shall be used for the SCSI ID in the Read Enclosure Status command where the SCSI ID for each device slot is returned to the host. The correspondence shall also be used by the host for the SCSI ID specified in the Set SCSI ID command. An empty or unused slot shall have a SCSI ID of FFh; a SCSI ID of FFh shall not be used to identify an active device slot.

Bits Field	7	6	5	4	3	2	1	0
SCSI ID	Port				Channel			

Figure 198 – SAF-TE Device ID field convention

Channel — The Channel field corresponds to a host connection. For a given enclosure slot, it indicates which host connection the slot is associated with. In the instance where there is an intervening Port Multiplier, the Channel field indicates which host channel the upstream port of the associated Port Multiplier is connected to. The Channel value is assigned serially starting at zero and is indicated on the host connections packaging accordingly.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 476 of 587

Port — The Port field corresponds to a device connection. For a given enclosure slot, it indicates which port of an intervening multi-port controller the slot is associated with. In the instance where the intervening controller is a Port Multiplier, the Port field indicates which port of the Port Multiplier the slot is connected to and corresponds to the PM Port field used in the FIS to address the device. In the absence of an intervening multi-port controller, this field is zero.

For the configuration in Figure 197, the SCSI ID value corresponding to Slot 0 in the enclosure would be 33h, while the SCSI ID value corresponding to Slot 3 in the enclosure would be 03h.

13.11.5.3.2 SES Correspondence Definition (Optional)

Figure 199 defines the SES Slot Address field convention for establishing correspondence between host connections and enclosure slots. SEPs that support SES may optionally adopt this convention. The method for determining whether an enclosure complies with this correspondence definition is vendor specific or as defined by the SES-2 specification.

The SEP shall use this convention, if implemented, in the Status Information field for a device element type (element type 01h) in the enclosure status page. The first byte in this field for an element of device type is the Slot Address and shall be set as specified in Figure 199. The enclosure status page can be read with the Receive Diagnostic Results command.

Bits Field	7	6	5	4	3	2	1	0
Slot Address	Port				Channel			

Figure 199 – SES Slot Address field convention

Channel — The Channel field corresponds to a host connection. For a given enclosure slot, it indicates which host connection the slot is associated with. In the instance where there is an intervening Port Multiplier, the Channel field indicates which host channel the upstream port of the associated Port Multiplier is connected to. The Channel value is assigned serially starting at zero and is indicated on the host connections packaging accordingly.

Port — The Port field corresponds to a device connection. For a given enclosure slot, it indicates which port of an intervening multi-port controller the slot is associated with. In the instance where the intervening controller is a Port Multiplier, the Port field indicates which port of the Port Multiplier the slot is connected to and corresponds to the PM Port field used in the FIS to address the device. In the absence of an intervening multi-port controller, this field is zero.

For the configuration in Figure 197, the Slot Address value corresponding to Slot 0 in the enclosure would be 33h, while the Slot Address value corresponding to Slot 3 in the enclosure would be 03h.

13.11.6 Enclosure Services Hardware Interface

For implementations where the enclosure/backplane is not bundled with the storage subsystem controller, it is recommended that the out-of-band enclosure services interface used by both the enclosure and the controller be I^2C. Section 13.11.6.1 defines the connector and cable interconnect between the enclosure/backplane and the storage subsystem controller.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 477 of 587

13.11.6.1 I^2C Cable/Connector definition

Implementations where the storage subsystem controller is not directly connected to the enclosure/backplane require an interconnect between the backplane and the controller for the I^2C enclosure services bus. For example, a self-contained system that uses a PCI-based Serial ATA RAID controller and houses a backplane with front-panel accessible devices, requires a means by which the I^2C interface originating on the PCI controller can be connected to the backplane that interfaces with the various storage devices and operator indicators (LEDs).

Products that utilize I^2C for enclosure services and desire interoperability/interchangeability with others' products shall use Molex Part # 22-43-6030 or equivalent as the bus connector. This connector has the same footprint/dimensions as the IPMB connector but its color is white instead of yellow as used by the actual IPMB connector.

13.11.6.2 SEP Discovery and Enumeration

For implementations where the SEP is an I^2C device and is not provided as part of a complete storage solution (i.e. is intended to communicate with a SEMB provided by another vendor), the SEP should have a dedicated I^2C interface with the SEMB and should not share the I^2C interface with enclosure sensors and indicators.

In order to enable the SEMB to efficiently discover and enumerate an attached SEP, the first SEP on an I^2C bus attached to a SEMB in a storage subsystem shall be assigned I^2C address C0h (note that the LSB of the I^2C address field is the R/W bit effectively resulting in the I^2C address field being 7 bits in length). A SEMB directly supports only a single SEP, so use of more than one SEP attached to the same SEMB in a storage subsystem is vendor specific. Any additional SEP(s) on the same I^2C interface in a storage subsystem shall be assigned consecutively higher I^2C addresses.

13.12 HDD Activity Indication (Optional)

Operator notification/indication of storage device status and activity may be driven by the host through the enclosure services facilities defined in Section 13.11 or through other host-driven means.
Operator notification/indication of storage device status and activity may also be driven through an intelligent processor such as an IO Processor. Reference the SAF-TE Write Device Slot Status command in the SAF-TE reference and section 13.11.5.2.1 for methods to support a SEP controlling HDD Activity Indication. As this reference suggests, these implementations may be vendor specific.

13.12.1 HDD Activity Emulation of Desktop Behavior

If the host controller optionally implements the desktop activity LED functionality, with the desired behavior to be compatible with current parallel ATA solutions, the host controller shall generate such a signal with the behavior defined in Figure 200.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 478 of 587

```
//          POR - Power On Reset
//          HRESET - Hardware RESET
RESET = POR  || HRESET;


if ((BSY || SActive) && DEVICE_TYPE != ATAPI)       // How !ATAPI determined is implementation specific
          ACTIVITY_LED = ON;
else
          ACTIVITY_LED = OFF;


if (MASTER_SLAVE_EMULATION_ENABLED && SLAVE_PRESENT)
{
          if (RESET)
                    SLAVE_LED = ON;
}
else
          SLAVE_LED = OFF;


If (REGISTER_FIS_TRANSMITTED_ON_SLAVE_CHANNEL)
          SLAVE_LED = OFF;


LED = ACTIVITY_LED || SLAVE_LED;
```

Figure 200 – Activity LED definition for desktop behavior emulation

In the LED behavioral logic, the means by which an implementation may determine that the attached device type is ATAPI (see logic expression DEVICE_TYPE != ATAPI in logic behavioral definition) is implementation specific, and may include detection of device type based on reset signature, detection of the command opcodes issued to the device (i.e. ATAPI devices have command issued using the ATAPI command codes of A0h and A1h), or through other means. The required behavior is that activity to ATAPI devices not generate LED activity indication.

For implementations that have multiple Serial ATA channels, the controller should provide an aggregate activity signal that is the wired-OR of the individual activity signals from each channel.

13.12.1.1 Desktop HDD Activity Signal Electrical Requirements

For implementations that provide an activity signal in accordance with section 13.12.1, the signal shall be active low and shall be of an open collector/drain design. The voltage and current requirements/capabilities of the signal is vendor-specific.

13.12.2 Activity/Status Indication Reference (Informative)

Serial ATA controller devices may include discrete physical pins for the purpose of connecting device activity LEDs. Such controllers may provide one device activity pin per port and/or an aggregate activity signal. They may be included so that the host (or IO Processor) software need not supply a device activity status, or so that a SEP need not be burdened with blinking a LED when writes/reads to/from a device are occurring. These signals allow drive activity LEDs without need for changes to the Serial ATA specification. It is likely that the methods outlined in this section may become obsolete as enclosure management facilities for Serial ATA mature and become readily available.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 479 of 587

Figure 201 shows an example configuration for implementing device activity LEDs within an enclosure. In this example, it would be likely that the solution would use standard 0.1" headers that are common and widely available. Depending upon the number of devices, the standard header would vary in size, but would generally require two pins per device activity LED in the configuration. Thus, a 4-port solution with 4-device status LEDs would require a 2x4 pin standard header.

Figure 202 is identical in configuration to that shown in Figure 201, except that it shows the use of a ribbon cable for ease of assembly.

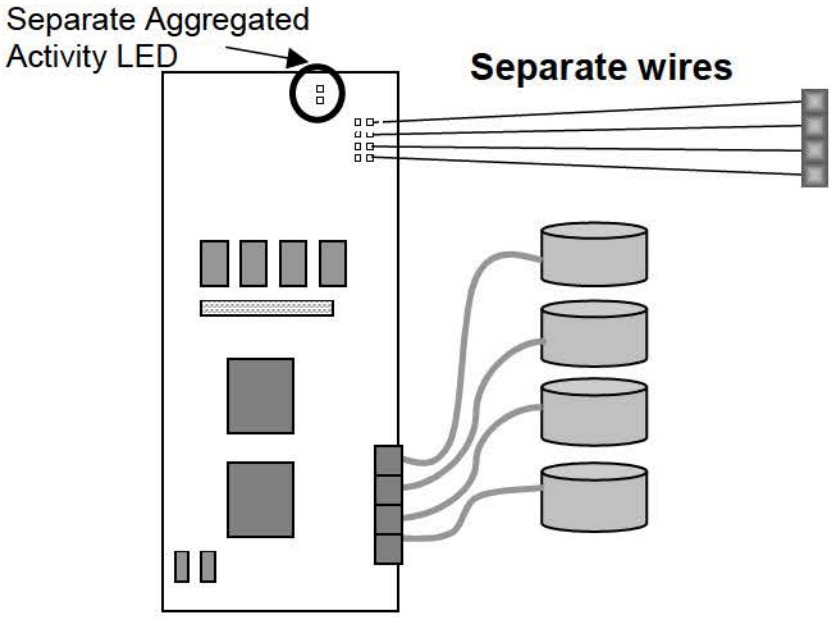


Figure 201 – Device Activity LEDs with Separate Wires

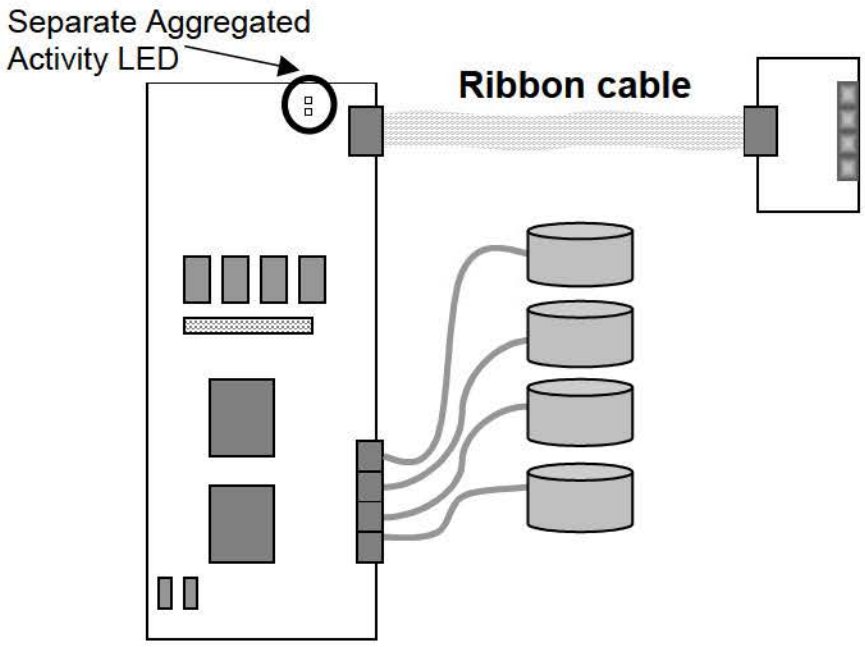


Figure 202 – Device Activity LEDs with Ribbon Cable

Note also the activity outputs have been designed to support wired-OR configurations. They can then support a configuration where all the outputs can be connected together for the purpose of generating an aggregate activity indication.

Taking advantage of the wired-OR functionality, Figure 203 shows a more complex configuration that would support additional needs that may be required by an integrated system. In this example, the devices plug straight into a backplane. The device activity LEDs are placed on an separate, small mezzanine card that could wire-OR the LED activity signals from both the Serial ATA controller on the I/O Controller and the SEP, which in this example is located on the backplane. This would then allow normal device activity to be indicated as described in the preceding paragraphs, but would also support the concept of the SEP generating distinguishable visual patterns to the LEDs for other purposes. For example, an LED could be placed into a steady state ON by the SEP to indicate a failed card or a card that is targeted for hot-plug.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 481 of 587

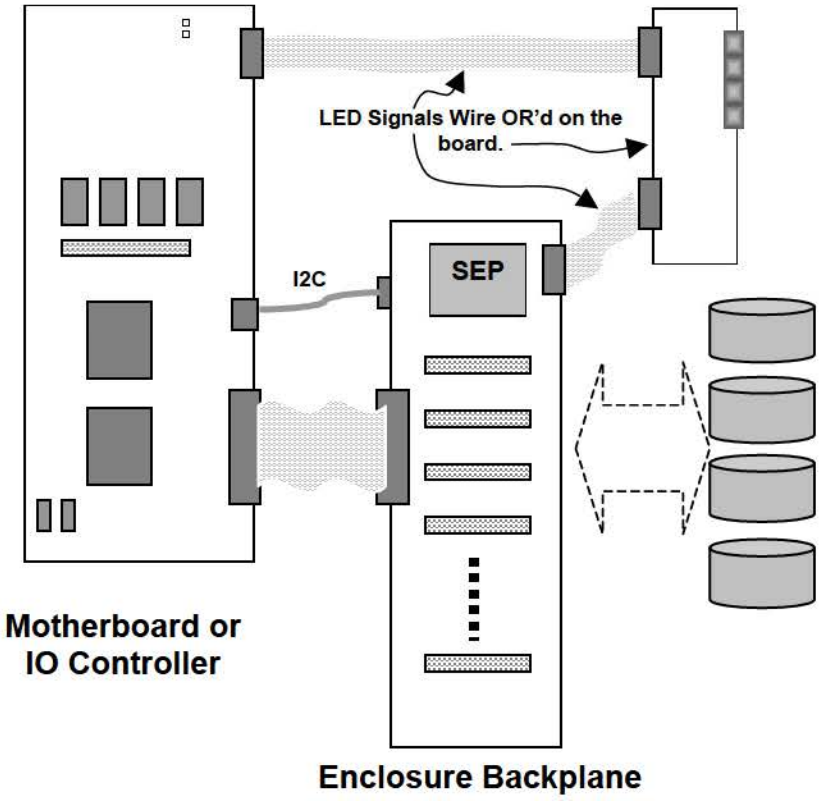


Figure 203 – Device Activity LEDs in a Storage Subsystem

Designers should be aware of some of the limitations of the solutions described in this section. If a solution is using a Port Multiplier, for example, then the LED might only be useful for general front end port activity, and might not identify a specific device on a specific Port Multiplier port for which an operator may be needing to take an action upon (such as device failed). Another consideration is that the signal generated by the Serial ATA controller may change states based upon certain states, and these state transitions may be too rapid to be readily detected by the operator. The LED pin may be asserted if the BSY bit is asserted or if the transport state machine is active for example. Thus, in certain configurations where many short packets are being transmitted/received, the device may be active when the LED appears to be off if no additional methods are implemented to prevent this case from occurring.

13.13 Port Multiplier Discovery and Enumeration

13.13.1 Power-up

On power-up, the Port Multiplier shall enter state HPHP1:NoComm in the hot plug state machine for the host port, specified in section 16.3.3.5.1. Upon entering this state, the Port Multiplier shall:

1. Clear any internal state and reset all parts of the Port Multiplier hardware.
2. Place the reset values in all Port Multiplier registers, including port specific registers. The reset values shall disable all device ports.

After performing this sequence, the Port Multiplier shall proceed with the actions described in the hot plug state machine for the host port.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 482 of 587

If the Port Multiplier receives a COMRESET from the host before issuing the initial COMINIT signal to the host, the Port Multiplier shall immediately perform the actions detailed in section 17.5.2.1 and cease performing the power-up sequence listed above.

13.13.2 Resets

There are three mechanisms to reset a device: COMRESET, software reset, and the DEVICE RESET command for PACKET devices.

To reset a Port Multiplier, the host shall issue a COMRESET. A Port Multiplier does not reset in response to a software reset or a DEVICE RESET command. The specific actions that a Port Multiplier takes in response to each reset type is detailed in the following sections.

13.13.2.1 COMRESET

When the Port Multiplier receives a COMRESET over the host port, the Port Multiplier shall enter state HPHP1:NoComm in the hot plug state machine for the host port, specified in section 16.3.3.5.1. Upon entering this state, the Port Multiplier shall:

1. Clear any internal state and reset all parts of the Port Multiplier hardware.
2. Place the reset values in all Port Multiplier registers, including port specific registers. The reset values shall disable all device ports.

After performing this sequence, the Port Multiplier shall proceed with the actions described in the hot plug state machine for the host port.

13.13.2.2 Software Reset

When the host issues a software reset to the control port, two Registers FISes are sent to the control port as a result. In the first Register FIS, the SRST bit in the Device Control register is set to one. In the second Register FIS, the SRST bit in the Device Control register is cleared to zero.

Upon receiving the Register FIS with the SRST bit asserted, the Port Multiplier shall wait for the Register FIS that has the SRST bit cleared before issuing a Register FIS with the Port Multiplier signature to the host. The Port Multiplier's behavior shall be consistent with the Software reset protocol described in section 11.3. The values to be placed in the Register FIS are listed in Figure 204.

Register	7	6	5	4	3	2	1	0
Error	00h							
Sector Count	01h							
Sector Count (exp)	00h							
LBA Low	01h							
LBA Low (exp)	00h							
LBA Mid	69h							
LBA Mid (exp)	00h							
LBA High	96h							
LBA High (exp)	00h							
Device	00h							
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Figure 204 – Software reset to control port result values

BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 0

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 483 of 587

The Port Multiplier shall take no reset actions based on the reception of a software reset. The only action that a Port Multiplier shall take is to respond with a Register FIS that includes the Port Multiplier signature. To cause a general Port Multiplier reset, the COMRESET mechanism is used.

13.13.2.3 Device Reset

A device reset command issued to the control port shall be treated as an unsupported command by the Port Multiplier. Refer to section 16.3.3.8.6.

13.13.3 Software Initialization Sequences (Informative)

This section details the sequences that host software should take to initialize a Port Multiplier device.

13.13.3.1 Port Multiplier Aware Software

Port Multiplier aware software checks the host's SStatus register to determine if a device is connected to the port. If a device is connected to the port, the host then issues a software reset to the control port. If the Port Multiplier signature is returned, then a Port Multiplier is attached to the port. Then the host proceeds with enumeration of devices on Port Multiplier ports as detailed in section 13.13.4.2.

Port Multiplier aware software shall not require a device to be present on device port 0h in order to determine if a Port Multiplier is present.

13.13.3.2 Non-Port Multiplier Aware Software

Non-Port Multiplier aware software waits for the signature of the attached device to be returned to the host. If a device is present on device port 0h, the device connected to device port 0h returns a Register FIS to the host that contains its signature. If a device is not present on device port 0h, non-Port Multiplier aware software times out waiting for the signature to be returned and assumes that a device failure has occurred. If fast boot is a requirement, the system should have a Port Multiplier aware BIOS and Port Multiplier aware OS driver.

When non-Port Multiplier aware software is loaded, all device ports other than device port 0h are disabled. The host will only receive FISes from the device attached to device port 0h.

13.13.3.3 Boot Devices Connected to Port Multiplier

System designers should only connect multiple boot devices to a Port Multiplier if the BIOS is Port Multiplier aware. For example, in a system that contains three bootable devices (hard drive, CD-ROM, and DVD) these devices should only be attached to the Port Multiplier if the BIOS is Port Multiplier aware or the user cannot boot off of the devices that are not connected to device port 0h.

13.13.4 Port Multiplier Discovery and Device Enumeration (Informative)

13.13.4.1 Port Multiplier Discovery

To determine if a Port Multiplier is present, the host performs the following procedure. The host determines if communication is established on the host's Serial ATA port by checking the host's SStatus register. If a device is present, the host issues a software reset with the PM Port field set to the control port. The host checks the signature value returned and if it corresponds to the Port Multiplier Signature, the host knows that a Port Multiplier is present. If the signature value does not correspond to a Port Multiplier, the host may proceed with the normal initialization sequence for that device type. The host shall not rely on a device being attached to device port 0h to determine that a Port Multiplier is present.

When a Port Multiplier receives a software reset to the control port, the Port Multiplier shall issue a Register FIS to the host according to the procedure detailed in section 17.5.2.2. The signature value contained in the Register FIS is shown in Figure 205 .

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 484 of 587

Register	7	6	5	4	3	2	1	0
Error	na							
Sector Count	01h							
Sector Count (exp)	na							
LBA Low	01h							
LBA Low (exp)	na							
LBA Mid	69h							
LBA Mid (exp)	na							
LBA High	96h							
LBA High (exp)	na							
Device	00h							
Status	na							

Figure 205 – Port Multiplier Signature

13.13.4.2 Device Enumeration

After discovering a Port Multiplier, the host enumerates all devices connected to the Port Multiplier. The host reads GSCR[2], defined in section 16.4.1.1, to determine the number of device ports on the Port Multiplier.

For each device port on the Port Multiplier, the host performs the following procedure to enumerate a device connected to that port:

1. The host enables the device port. The host can enable a device port by setting the DET field appropriately in the device port's PSCR[2] (SControl) register, as specified in section 14.1.3. The host uses the WRITE PORT MULTIPLIER command to write PSCR[2] (SControl) for the device port to be enabled.

2. The host should allow for communication to be established and device presence to be detected after enabling a device port. Refer to section 8.3.1 that describes the host PHY initialization sequence.

3. The host reads PSCR[0] (SStatus) for the device port using the READ PORT MULTIPLIER command. If PSCR[0] (SStatus) indicates that a device is present, the host queries PSCR[1] (SError) for the device port and clears the X bit indicating device presence has changed.

4. The signature generated by the device as a consequence of the initial COMRESET to the device port may be discarded if the host does not support context switching because the BSY bit may be clear when the Register FIS is received by the host. Therefore, to determine the signature of the attached device, the host should issue a software reset to the device port. If a valid signature is returned for a recognized device, the host may then proceed with normal initialization for that device type.

Cascading Port Multipliers shall not be supported.

14 Host adapter register interface

Serial ATA host adapters include an additional block of registers mapped separately and independently from the ATA Command Block Registers for reporting additional status and error information and to allow control of capabilities unique to Serial ATA. These additional registers, referred to as the Serial ATA Status and Control Registers (SCR's) are organized as 16 contiguous 32-bit registers. The base address and mapping scheme for these registers is defined by the specific host adapter implementation – for example, PCI controller implementations may map the SCR's using the PCI mapping capabilities. Table 42 illustrates the overall organization of the Serial ATA register interface including both the ATA Command Block Registers and the Status and Control registers. Legacy software does not make use of the Serial ATA Status and Control registers. The Serial ATA Status and Control register are associated with the serial interface and are independent of any master/slave emulation the host adapter may implement.

Table 42 – SCR definition

ATA Command Block and Control Block registers

	A2	A1	A0	Read	Write
CS 0	0	0	0	Data	Data
	0	0	1	Error	Features
	0	1	0	Sector Count [15:8] [7:0]	Sector Count [15:8] [7:0]
	0	1	1	LBA Low [31:24] [7:0]	LBA Low [31:24] [7:0]
	1	0	0	LBA Mid [39:32] [15:8]	LBA Mid [39:32] [15:8]
	1	0	1	LBA High [47:40] [23:16]	LBA High [47:40] [23:16]
	1	1	0	Device	Device
	1	1	1	Status	Command
CS 1	1	1	0	Alternate Status	Device Control

Serial ATA Status and Control registers

SATA register	0	SATA Status/Control
SATA register	1	SATA Status/Control
…	…	SATA Status/Control
SATA register	14	SATA Status/Control
SATA register	15	SATA Status/Control

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 486 of 587

14.1 Status and Control Registers

Serial ATA provides an additional block of registers to control the interface and to retrieve interface state information. There are 16 contiguous registers allocated of which the first five are defined and the remaining 11 are reserved for future definition. Table 43 defines the Serial ATA Status and Control registers.

Table 43 – SCR Definition

SCR[0]	SStatus register
SCR[1]	SError register
SCR[2]	SControl register
SCR[3]	SActive register
SCR[4]	SNotification register
SCR[5]	Reserved
…	…
SCR[15]	Reserved

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 487 of 587

14.1.1 SStatus register

The Serial ATA interface Status register - SStatus - is a 32-bit read-only register that conveys the current state of the interface and host adapter. The register conveys the interface state at the time it is read and is updated continuously and asynchronously by the host adapter. Writes to the register have no effect.



DET The DET value indicates the interface device detection and Phy state.
- 0000b No device detected and Phy communication not established
- 0001b Device presence detected but Phy communication not established
- 0011b Device presence detected and Phy communication established
- 0100b Phy in offline mode as a result of the interface being disabled or running in a BIST loopback mode

All other values reserved

SPD The SPD value indicates the negotiated interface communication speed established
- 0000b No negotiated speed (device not present or communication not established)
- 0001b Generation 1 communication rate negotiated
- 0010b Generation 2 communication rate negotiated

All other values reserved

IPM The IPM value indicates the current interface power management state
- 0000b Device not present or communication not established
- 0001b Interface in active state
- 0010b Interface in Partial power management state
- 0110b Interface in Slumber power management state

All other values reserved

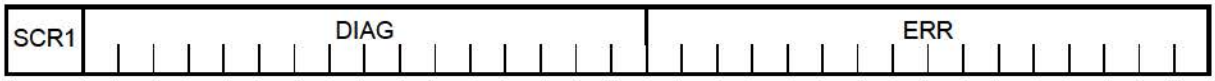
Reserved All reserved fields shall be cleared to zero.

NOTE – The interface needs to be in the active state for the interface device detection value (DET field) to be accurate. When the interface is in the Partial or Slumber state no communication between the host and target is established resulting in a DET value corresponding to no device present or no communication established. As a result the insertion or removal of a device may not be accurately detected under all conditions such as when the interface is quiescent as a result of being in the Partial or Slumber state. This field alone may therefore be insufficient to satisfy all the requirements for device attach or detatch detection during all possible interface states.

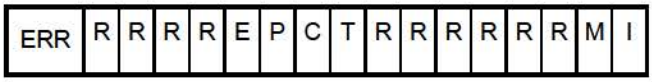
Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199

14.1.2 SError register

The Serial ATA interface Error register - SError - is a 32-bit register that conveys supplemental Interface error information to complement the error information available in the Shadow Register Block Error register. The register represents all the detected errors accumulated since the last time the SError register was cleared (whether recovered by the interface of not). Set bits in the error register are explicitly cleared by a write operation to the SError register, or a reset operation. The value written to clear set error bits shall have 1's encoded in the bit positions corresponding to the bits that are to be cleared. Host software should clear the Interface SError register at appropriate checkpoints in order to best isolate error conditions and the commands they impact.



ERR The ERR field contains error information for use by host software in determining the appropriate response to the error condition. The field is bit significant as defined in the following figure.



C Non-recovered persistent communication or data integrity error: A communication error that was not recovered occurred that is expected to be persistent. Since the error condition is expected to be persistent the operation need not be retried by host software. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.

E Internal error: The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. Host software should reset the interface before re-trying the operation. If the condition persists, the host bus adapter may suffer from a design issue rendering it incompatible with the attached device.

I Recovered data integrity error: A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This can arise from a noise burst in the transmission, a voltage supply variation, or from other causes. No action is required by host software since the operation ultimately succeeded, however, host software may elect to track such recovered errors in order to gauge overall communications integrity and potentially step down the negotiated communication speed.

M Recovered communications error: Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes and may be derived from the PHYRDYn signal between the Phy and Link layers. No action is required by the host software since the operation ultimately succeeded, however, host software may elect to track such recovered errors in order to gauge overall communications integrity and potentially step down the negotiated communication speed.

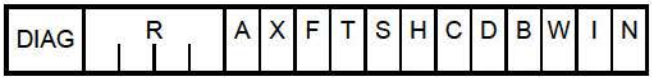
P Protocol error: A violation of the Serial ATA protocol was detected. This can arise from invalid or poorly formed FISes being received, from invalid state transitions, or from other causes. Host software should reset the interface and retry the corresponding operation. If such an error persists, the attached device may have a design issue rendering it incompatible with the host bus adapter.

R Reserved bit for future use: Shall be cleared to zero.

T Non-recovered transient data integrity error: A data integrity error occurred that was not recovered by the interface. Since the error condition is not expected to be persistent the operation should be retried by host software.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 489 of 587

DIAG The DIAG field contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. The field is bit significant as defined in the following figure.



A Port Selector presence detected: This bit is set to one when COMWAKE is received while the host is in state HP2: HR_AwaitCOMINIT. On power-up reset this bit is cleared to zero. The bit is cleared to zero when the host writes a one to this bit location.

B 10b to 8b Decode error: When set to a one, this bit indicates that one or more 10b to 8b decoding errors occurred since the bit was last cleared.

C CRC Error: When set to one, this bit indicates that one or more CRC errors occurred with the Link Layer since the bit was last cleared.

D Disparity Error: When set to one, this bit indicates that incorrect disparity was detected one or more times since the last time the bit was cleared.

F Unrecognized FIS type: When set to one, this bit indicates that since the bit was last cleared one or more FISes were received by the Transport layer with good CRC, but had a type field that was not recognized.

I Phy Internal Error: When set to one, this bit indicates that the Phy detected some internal error since the last time this bit was cleared.

N PHYRDY change: When set to one, this bit indicates that the PHYRDY signal changed state since the last time this bit was cleared.

H Handshake error: When set to one, this bit indicates that one or more R_ERR_P handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 10b/8b decoding error, or other error condition leading to a negative handshake on a transmitted frame.

R Reserved bit for future use: Shall be cleared to zero.

S Link Sequence Error: When set to one, this bit indicates that one or more Link state machine error conditions was encountered since the last time this bit was cleared. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition.

T Transport state transition error: When set to one, this bit indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared.

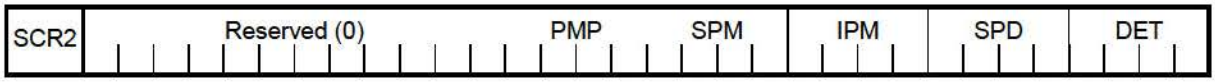
W COMWAKE Detected: When set to one this bit indicates that a COMWAKE signal was detected by the Phy since the last time this bit was cleared.

X Exchanged: When set to one this bit indicates that device presence has changed since the last time this bit was cleared. The means by which the implementation determines that the device presence has changed is vendor specific. This bit may be set anytime a Phy reset initialization sequence occurs as determined by reception of the COMINIT signal whether in response to a new device being inserted, in response to a COMRESET having been issued, or in response to power-up.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 490 of 587

14.1.3 SControl register

The Serial ATA interface Control register - SControl - is a 32-bit read-write register that provides the interface by which software controls Serial ATA interface capabilities. Writes to the SControl register result in an action being taken by the host adapter or interface. Reads from the register return the last value written to it.



DET The DET field controls the host adapter device detection and interface initialization.
- 0000b No device detection or initialization action requested
- 0001b Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. Upon a write to the SControl register that sets the DET field to 0001b, the host interface shall transition to the HP1: HR_Reset state and shall remain in that state until the DET field is set to a value other than 0001b by a subsequent write to the SControl register.
- 0100b Disable the Serial ATA interface and put Phy in offline mode.
- All other values reserved

SPD The SPD field represents the highest allowed communication speed the interface is allowed to negotiate when interface communication speed is established
- 0000b No speed negotiation restrictions
- 0001b Limit speed negotiation to a rate not greater than Gen 1 communication rate
- 0010b Limit speed negotiation to a rate not greater than Gen 2 communication rate
- All other values reserved

IPM The IPM field represents the enabled interface power management states that can be invoked via the Serial ATA interface power management capabilities
- 0000b No interface power management state restrictions
- 0001b Transitions to the Partial power management state disabled
- 0010b Transitions to the Slumber power management state disabled
- 0011b Transitions to both the Partial and Slumber power management states disabled
- All other values reserved

SPM The Select Power Management (SPM) field is used to select a power management state. A non-zero value written to this field shall cause the power management state specified to be initiated. A value written to this field is treated as a one-shot. This field shall be read as 0000b.
- 0000b No power management state transition requested
- 0001b Transition to the Partial power management state initiated
- 0010b Transition to the Slumber power management state initiated
- 0100b Transition to the active power management state initiated
- All other values reserved

PMP The Port Multiplier Port (PMP) field represents the 4-bit value to be placed in the PM Port field of all transmitted FISes. This field is '0000' upon power-up. This field is optional and an HBA implementation may choose to ignore this field if the FIS to be transmitted is constructed via an alternative method.

Reserved All reserved fields shall be cleared to zero.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 491 of 587

14.1.4 SActive register

The SActive register is a 32-bit register that conveys the information returned in the SActive field of the Set Device Bits FIS. If NCQ is not supported, then the SActive register does not need to be implemented.

The host may set bits in the SActive register by a write operation to the SActive register. The value written to set bits shall have ones encoded in the bit positions corresponding to the bits that are to be set. Bits in the SActive register are not cleared as a result of a register write operation by the host, and host software cannot directly clear bits in the SActive register.

Set bits in the SActive register are cleared as a result of data returned by the device in the SActive field of the Set Device Bits FIS. The value returned in the SActive field of the Set Device Bits FIS shall have ones encoded in the bit positions corresponding to the bits that are to be cleared in the SActive register. The device cannot set bits in the SActive register.

The host controller shall clear all bits in the SActive register upon issuing a hard reset (COMRESET) signal or as a result of issuing a software reset by transmitting a Control Register FIS with the SRST bit asserted.

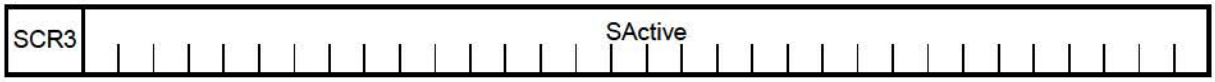


Figure 206 – SActive register definition

SActive — For the Native Command Queuing protocol, the SActive value represents the set of outstanding queued commands that have not completed successfully yet. See section 13.5.

14.1.5 SNotification register (Optional)

The Serial ATA interface notification register - SNotification - is a 32-bit register that conveys the devices that have sent the host a Set Device Bits FIS with the Notification bit set, as specified in section 10.3.6. When the host receives a Set Device Bits FIS with the Notification bit set to one, the host shall set the bit in the SNotification register corresponding to the value of the PM Port field in the received FIS. For example, if the PM Port field is set to 7 then the host shall set bit 7 in the SNotification register to one. After setting the bit in the SNotification register, the host shall generate an interrupt if the I bit is set to one in the FIS and interrupts are enabled.

Set bits in the SNotification register are explicitly cleared by a write operation to the SNotification register, or a power-on reset operation. The register is not cleared due to a COMRESET, software is responsible for clearing the register as appropriate. The value written to clear set bits shall have ones encoded in the bit positions corresponding to the bits that are to be cleared.

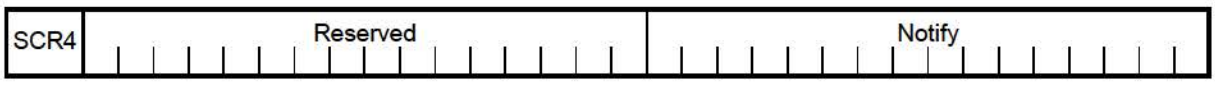


Figure 207 – SNotification register definition

Notify — The field represents whether a particular device with the corresponding PM Port number has sent a Set Device Bits FIS to the host with the Notification bit set.

Reserved — The reserved field shall be cleared to zero.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 492 of 587

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 493 of 587

15 Error handling

15.1 Architecture

The layered architecture of Serial ATA extends to error handling as well. As indicated in Figure 208, each layer in the Serial ATA stack has as inputs error indication from the next lower layer (except for the Phy layer which has no lower layer associated with it), data from the next lower layer in the stack, and data from the next higher layer in the stack. Each layer has its local error detection capability to identify errors specific to that layer based on the received data from the lower and higher layers. Each layer performs local recovery and control actions and may forward error information to the next higher layer in the stack.

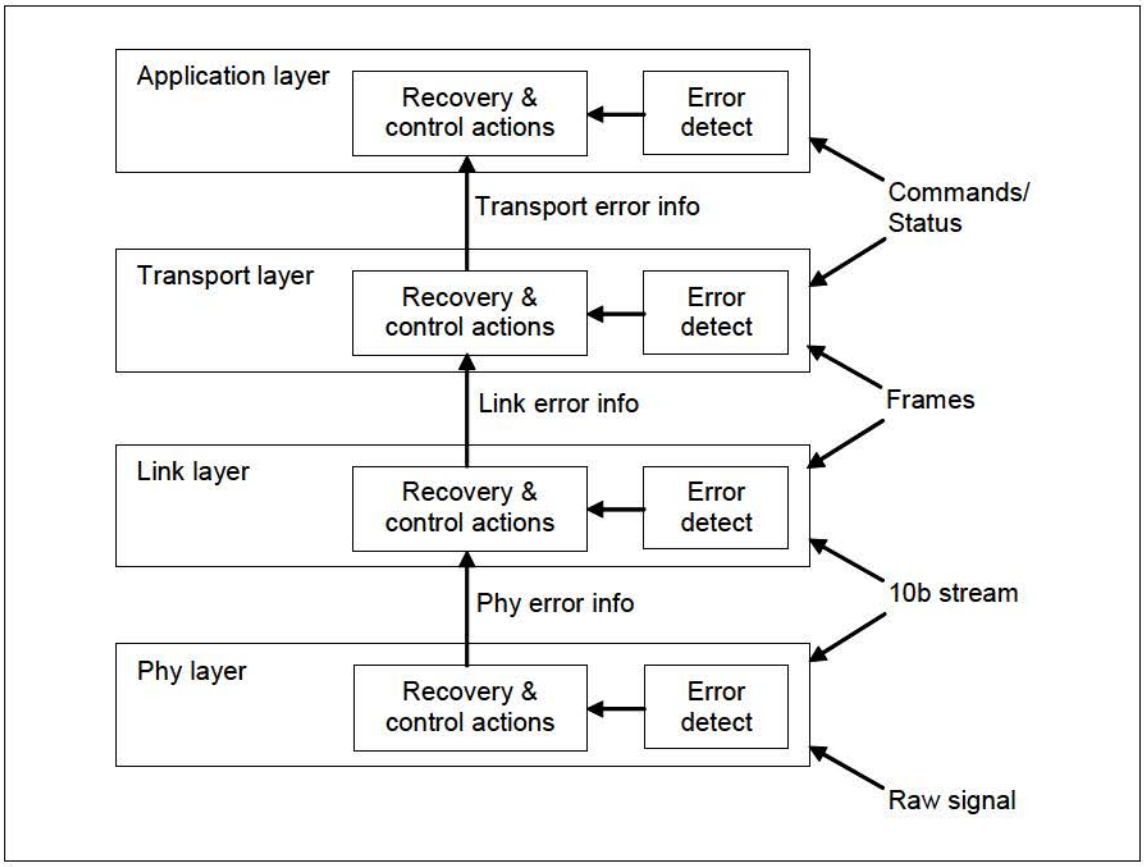


Figure 208 – Error handling architecture

Error responses are generally classified into four categories

Freeze
Abort
Retry
Track/ignore

The error handling responses described in this section are not comprehensive and are included to cover specific known error scenarios as well as to illustrate typical error control and recovery actions. This section is therefore descriptive and supplemental to the error reporting interface defined in section 14.1.2 and implementations may vary in their internal error recovery and control actions.

For the most severe error conditions in which state has been critically perturbed in a way that it is not recoverable, the appropriate error response is to freeze and rely on a reset or similar operation to restore all necessary state to return to normal operation.

For error conditions that are expected to be persistent, the appropriate error response is to abort and fail the attempted operation. Such failures usually imply notification up the stack in order to inform host software of the condition.

For error conditions that are transient and not expected to persist, the appropriate response is to retry the failed operation. Only failed operations that have not perturbed system state can be retried. Such retries may either be handled directly by the recovery and error control actions in the relevant layer or may be handled by host software in response to error information being conveyed to it.

Non-critical recoverable conditions can either be tracked or ignored. Such conditions include those that were recovered through a retry or other recovery operation at a lower layer in the stack. Tracking such errors can often be beneficial in identifying a marginally operating component or other imminent failure.

15.2 Phy error handling overview

15.2.1 Error detection

There are three primary categories of error that the Phy layer detects internally:

- No device present
- OOB signaling sequence failure
- Phy internal error (loss of synchronization of communications link)

A no device present condition results from a physical disconnection in the media between the host controller and device, whether intermittent or persistent. The host controller shall detect device presence as part of the interface reset sequence defined in section 7.5.

An OOB signaling failure condition arises when the sequence of OOB signaling events cannot be completed, prior to the PHYRDY signal being asserted. OOB signaling sequences are required when emerging from a power management Partial or Slumber state, from a loopback BIST test state, or from initial power-up. OOB signaling sequences are used to achieve a specifically ordered exchange of COMRESET, COMINIT, COMWAKE, and $ALIGN_P$ patterns to bring the communications link up between host controller and device. The specific sequences are detailed in section 7.5.

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 495 of 587

A Phy Internal Error can arise from a number of conditions, whether it is caused by the characteristics of the input signal, or an internal error unique to the implementation, it always results in the loss of the synchronization of the communications link.

Fixed local receive PLL frequency architectures (oversampling, tracking/non-tracking) are sensitive to input data rate frequency variations from the nominal expected rate, thus they usually have elasticity buffers. Elasticity buffers are used to accommodate the difference between input data rate frequency, and the local receive PLL frequency -- overrun/underrun conditions may occur if the Tx difference in frequency is too high/low with respect to the Rx local PLL frequency, commonly declared as a Phy Internal Error.

VCO-based (PLL) clock recovery architectures are also sensitive to input data rate frequency variations, high frequency jitter, and may have trouble achieving lock -- which may be declared as a Phy Internal Error.

A number of state machine, impedance compensation, and serializer/deserializer (SerDes) circuits make up a typical Serial ATA interface Phy, and the various types of error conditions can be grouped together to make up the declared "Phy Internal Error". These errors are usually specific to each implementation.

15.2.2 Error control actions

15.2.2.1 No device present

Due to the nature of the physical interface, it is possible for the Phy to determine that a device is attached to the cable at various times. As a direct result, the Phy is responsible for detecting presence of an attached device and this presence shall be reported in the SStatus register such that host software can respond appropriately.

During the interface initialization sequence, an internal state bit “Device Detect” shall be cleared in the host controller when the COMRESET signal is issued. The “Device Detect” state bit shall be set in the host controller when the host controller detects a COMINIT signal from the attached device. The “Device Detect” state bit may be set in the host controller when the host controller detects a COMWAKE signal while recovering from the Partial or Slumber state. The “Device Detect” state bit corresponds to the device presence detect information in the SStatus register defined in section 14.1.1.

Note that device presence and communications established are separately reported in the SStatus register in order to encompass situations in which an attached device is detected by the Phy, but the Phy is unable to establish communications with it.

15.2.2.2 OOB signaling sequence failure

The Phy does not have any timeout conditions for the interface OOB reset signaling sequence as defined in section 8.1. If a device is present, the Phy shall detect device presence within 10 ms of a power-on reset (i.e. COMINIT shall be returned within 10 ms of an issued COMRESET). If a device is not present, the Phy is not required to time-out and may remain in the reset state indefinitely until host software intervenes. Upon successful completion of the interface initialization sequence, the Phy shall be ready, active, and synchronized, and the SStatus register bits shall reflect this as defined in section 14.1.1.

15.2.2.3 Phy internal error

As described in section 15.2.1, there are several potential sources of errors categorized as “Phy Internal Errors.” In order to accommodate a range of implementations without making the software error handling approach implementation dependent, all the different potential sources of

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 496 of 587

internal Phy errors are combined for the purpose of reporting the condition in the SError register as defined in section 14.1.2. This requirement does not preclude each vendor from implementing their own level of error diagnostic bits, but those bits shall reside in vendor specific register locations.

Phy internal errors shall result in the Phy becoming not ready (the PHYRDY signal being deasserted) and the corresponding SStatus register and SError register bits shall be updated as defined in section 14.1.

The "PHYRDY change" bit, as defined in the SError register, shall be updated as per its definition in section 14.1.2

15.2.3 Error reporting

Phy errors are generally reported to the Link layer in addition to being reflected in the SStatus register and SError register as defined in section 14.1.

15.3 Link layer error handling overview

15.3.1 Error detection

There are two primary categories of errors that the Link layer detects internally:

Invalid state transitions
Data integrity errors

Invalid state transition errors can arise from a number of sources and the Link layer responses to many such error conditions are defined in section 9. Data integrity errors generally arise from noise in the physical interconnect.

15.3.2 Error control actions

Errors detected by the Link layer during a transmission are generally handled by accumulating the errors until the end of the transmission and reflecting the reception error condition in the final R_ERR_P /R_OK_P handshake. Specific scenarios are listed in the following sections.

15.3.2.1 Invalid state transitions

Invalid state transitions are generally handled through the return to a known state, for example where the Link state diagrams in section 9.6 define the responses for invalid state transition attempts. Returning to a known state is generally achieved through two recovery paths depending on the state of the system.

If the invalid state transitions are attempted during the transmission of a frame (after the receipt of an SOF_P), the Link layer shall signal negative acknowledgement (R_ERR_P) to the transmitting agent.

If the invalid state transition is not during a frame transmission, the Link shall go directly to the idle state, and await the next operation.

The following paragraphs outline the requirements during specific state transition scenarios, and their respective Link error control actions:

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 497 of 587

Following reception of one or more consecutive X_RDY_P at the receiver interface, if the next control character received is not SOF_P, the Link layer shall notify the Transport layer of the condition and transition to the idle state.

Following transmission of X_RDY_P, if there is no returned R_RDY_P received, no Link layer recovery action shall be attempted. The higher-level layers should eventually time out, and reset the interface.

On receipt of an unexpected SOF_P, when the receiving interface had not yet signaled readiness to receive data with R_RDY_P, that receiving interface shall remain in the idle state issuing $SYNC_P$ primitives until the transmitting interface terminates the transmission and also returns to the idle state.

If the transmitter closes a frame with EOF_P and $WTRM_P$, and receives neither a R_OK_P nor an R_ERR_P within a predetermined timeout, no Link layer recovery action shall be attempted. The higher-level layers should eventually time out, and reset the interface.

If the transmitter signals EOF_P, and a primitive other than $SYNC_P$, R_OK_P, or R_ERR_P is received, the Link layer shall persistently continue to await reception of a proper terminating primitive.

Data integrity errors:
Data integrity errors are generally handled by signaling the Transport layer in order to potentially trigger a transmission retry operation, or to convey failed status information to the host software. In order to return to a known state, data integrity errors are usually signaled via the frame acknowledgement handshake, at the end of a frame transmission, before returning to the idle state.

The following paragraphs outline the requirements during specific data integrity error scenarios, and the respective Link error control actions:

On detection of a CRC error at the end of receiving a frame (at EOF_P), the Link layer shall notify the Transport layer that the received frame contains a CRC error. Furthermore, the Link layer shall issue the negative acknowledgement, R_ERR_P, as the frame status handshake, and shall return to the idle state.

On detection of a disparity error or other 8b/10b coding violation during the receipt of a frame, the Link layer shall retain this error information, and at the close of the received frame the Link layer shall provide the negative acknowledgement, R_ERR_P, as the frame handshake, and shall notify the Transport layer of the error.

The control actions are essentially the same for coding violations as for CRC errors.

15.3.3 Error reporting

Link layer error conditions are reported to the Transport layer via a private interface between the Link layer and Transport layer. Additionally, Link layer errors are reported in the SError register as defined section 14.1.2.

15.4 Transport layer error handling overview

The Transport layer is the highest level layer in the Serial ATA interface. The Transport layer communicates errors to the software and/or performs local error recovery, and initiates control actions, such as retrying a class of FIS transmissions

Silicon Motion - Exhibit 1020
Silicon Motion v. UTL - IPR2024-00199
Page 498 of 587

The Transport layer informs the Link layer of detected errors so that the Link layer can reflect Transport errors in the R_ERR_P/R_OK_P handshake at the end of each frame. Devices shall reflect any R_ERR_P frame handshakes in the command ending status reflected in the transmitted Register FIS that conveys the operation ending status. The Transport layer also reflects any encountered error information in the SError register.

The Transport layer may retry any FIS transmission, provided the system state has not changed as a result of the corresponding failure, and may retry any number of times. For scenarios where repeated retry operations persistently fail, host software should eventually time out the corresponding command and perform recovery operations.

15.4.1 Error detection

In addition to the error information passed to it by the Link layer, the Transport layer internally detects the following categories of errors:

Internal errors
Frame errors
Protocol errors & state errors

There are several kinds of internal errors to the Transport layer, including overflow/underflow of the various speed matching FIFOs. Internal errors are generally handled by failing the corresponding transaction, and returning to a state equivalent to a failed transaction (e. g. the state that would result from a bad CRC).

The Transport layer detects several kinds of frame errors including reception of frames with incorrect CRC, reception of frames with invalid TYPE field, and reception of ill-formed frames (such as a register frames that are not the correct length). Frame errors are generally handled by failing the corresponding transaction and returning to a state equivalent to a failed transaction (such as the state that would result from a bad CRC).

Protocol and state transition errors often stem from ill-behaved devices not following the proper Serial ATA protocol, and include errors such as the PIO count value not matching the number of data characters subsequently transferred and errors in the sequence of events.

Protocol and state transition errors are generally handled by failing the corresponding transaction and returning to a state equivalent to a failed transaction (such as the state that would result from a frame being received with a bad CRC).

15.4.2 Error control actions

15.4.2.1 Internal errors

Internal errors are normally handled by failing the corresponding transaction and either re-trying the transaction or notifying host software of the failure condition in order to ultimately generate a host software retry response. The following are specific internal error scenarios and their corresponding Transport layer error control actions.

If the receive FIFO overflows, the Transport layer shall signal frame reception negative acknowledgement, by signaling the Link layer to return R_ERR_P during the frame acknowledgement handshake. Subsequent actions are equivalent to a frame reception with erroneous CRC.

If the transmit FIFO underruns, the Transport layer shall close the transmitting frame with an EOF_P and CRC value that is forced to be incorrect in order to ensure the receiver of the corrupted frame also executes appropriate error control actions.