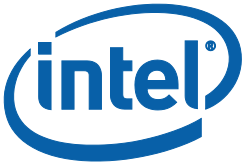


Enhanced Serial Peripheral Interface (eSPI)

Addendum for Server Platforms

December 2013

Revision 0.7

329957

Intel hereby grants you a fully-paid, non-exclusive, non-transferable, worldwide, limited license (without the right to sublicense), under its copyrights to view, download, and reproduce the Enhanced Serial Peripheral Interface (eSPI) Specification ("Specification"). You are not granted any other rights or licenses, by implication, estoppel, or otherwise, and you may not create any derivative works of the Specification.

The Specification is provided "as is," and Intel makes no representations or warranties, express or implied, including warranties of merchantability, fitness for a particular purpose, non-infringement, or title. Intel is not liable for any direct, indirect, special, incidental, or consequential damages arising out of any use of the Specification, or its performance or implementation.

Intel retains ownership of all of its intellectual property rights in the Specification and retains the right to make changes to the Specification at any time. No license is granted to use Intel's name, trademarks, or patents.

If you provide feedback or suggestions on the Specification, you grant Intel a perpetual, non-terminable, fully-paid, nonexclusive, worldwide license, with the right to sublicense, under all applicable intellectual property rights to use the feedback and suggestions, without any notice, consent, or accounting. You represent and warrant that you own, or have sufficient rights from the owner of, the feedback and suggestions, and the intellectual property rights in them, to grant the above license.

This agreement is governed by Delaware law, without reference to choice of law principles. Any disputes relating to this agreement must be resolved in the federal or state courts in Delaware and you consent, and will not object, to the exclusive personal jurisdiction of the courts in Delaware.

This agreement is the entire agreement of the parties regarding the Specification and supersedes all prior agreements or representations.

This agreement is hosted at the following location: http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=21353

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

Except for a limited copyright license to copy this specification for internal use only, no license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Intel Corporation and the authors of this specification disclaim all liability, including liability for infringements of proprietary rights, relating to implementation of information in this document and the specification. Intel Corporation and the authors of this specification also do not warrant or represent that such implementation(s) will not infringe such rights.

Implementations developed using the information provided in this specification may infringe the patent rights of various parties including the parties involved in the development of this specification. Except as expressly granted hereunder, no license, express or implied, by estoppel or otherwise, to any intellectual property rights (including without limitation rights under any party's patents) is granted.

All suggestions or feedback related to this specification become the property of Intel Corporation upon submission.

Intel may make changes to the specifications, product descriptions, and plans at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

This document is an intermediate draft for comment only and is subject to change without notice. Do not finalize a design based on this document.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
329957	0.7	• Initial revision for the eSPI addendum for server platforms	December 2013

§ §

1 Introduction

This addendum to the Enhanced Serial Peripheral Interface (eSPI) Base Specification Rev 0.75 describes an extension intended primarily to support server platforms.

1.1 References

1. Enhanced Serial Peripheral Interface (eSPI) Base Specification Rev 0.75, June 2013
2. Serial Flash Hardening, RPMC Specification Rev 0.72, March 2013

§

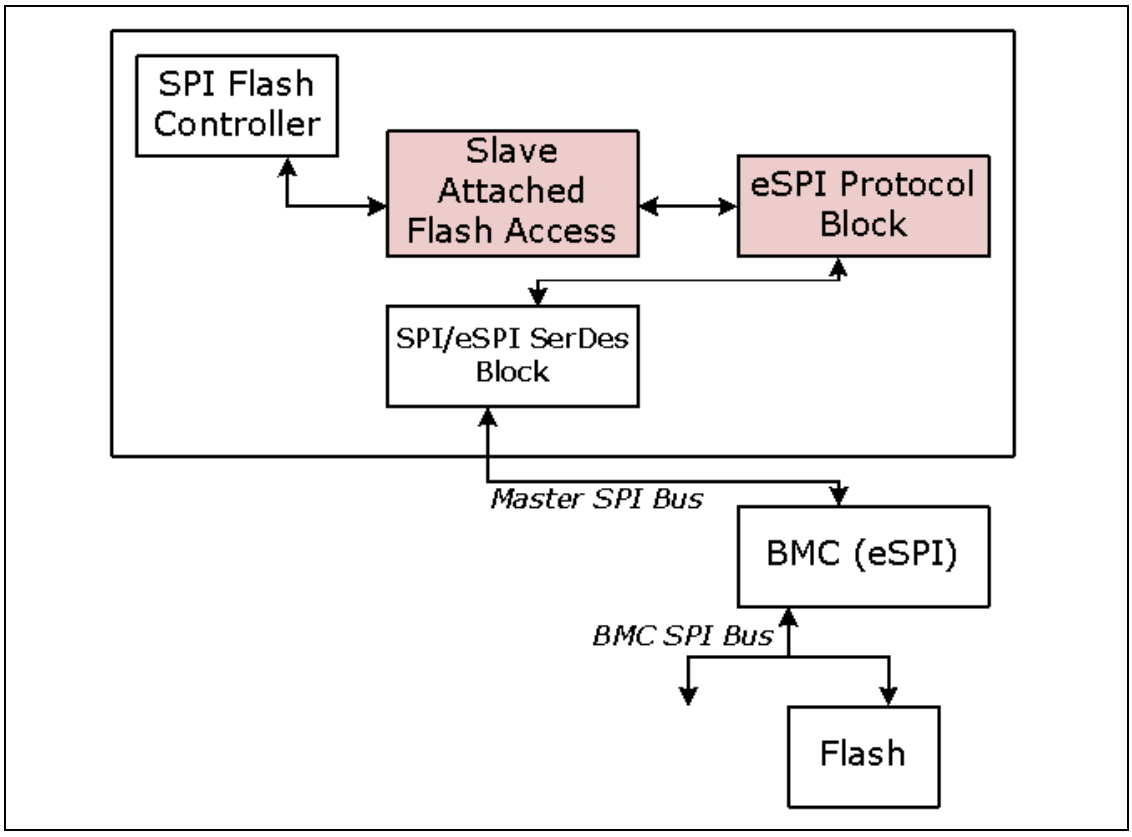
2 Slave Attached Flash Sharing

For server platforms, an alternate configuration for run-time flash access supported by the eSPI protocol is to put the flash device(s) behind the BMC. All the flash accesses by the eSPI Master are tunneled through the BMC over the eSPI protocol (Figure 2-1). The BMC then communicates with the flash device(s) to perform the requested flash operations and return the completion data back to the Master over the eSPI protocol. This allows run-time flash sharing between BMC and the eSPI Master.

The Slave Attached Flash Sharing is mutually exclusive with the Master Attached Flash Sharing for a given eSPI interface. Prior to Flash Access channel being enabled, either the Slave Attached Flash Sharing or the Master Attached Flash Sharing is selected for operation, but not both. The eSPI Master will have access to the flash as soon as the Flash Access channel is enabled on the eSPI Slave side.

The support of Slave attached Trusted Platform Module (TPM) is beyond the scope of this addendum.

Figure 2-1. Slave Attached Flash Sharing



2.1 Slave Register

The eSPI slave register at offset 40h – Channel 3 Capabilities and Configurations register is defined in the eSPI Base Specification for the operation of the Flash Access channel.

There are register fields defined or updated by this addendum. The definition is backward compatible with the eSPI Base Specification where the read value of '0' for these fields previously is consistent with the default intent of these bits defined in this addendum.

The eSPI slave advertises the flash sharing capability supported. The flash sharing scheme in operation is indicated by the Flash Sharing Mode bit which is either a Read-Only (RO) bit, or a Read-Write (RW) bit that configured by the eSPI master accordingly.

Besides, a Channel 3 Capabilities and Configurations 2 register is added at offset 44h. The eSPI slave advertises the Target Maximum Read Request Size Supported, Target Flash Erase Block Size for Master's Regions and Target RPMC Supported in this register.

2.1.1 Offset 40h: Channel 3 Capabilities and Configurations

Table 2-1. Channel 3 Capabilities and Configurations

<table>
<tr><th>Bit</th><th>Type</th><th>Default</th><th>Description</th></tr>
<tr><td>31:18</td><td>RO</td><td>0</td><td>Reserved.</td></tr>
<tr><td>17:16</td><td>RO</td><td>HwInit</td><td>Flash Sharing Capability Supported: This field indicates the flash sharing capability supported by the slave.
<table>
<tr><th>Bits</th><th>Slave Attached Flash Sharing</th><th>Master Attached Flash Sharing</th></tr>
<tr><td>00</td><td rowspan="2">Not Supported</td><td rowspan="2">Supported</td></tr>
<tr><td>01</td></tr>
<tr><td>10</td><td>Supported</td><td>Not Supported</td></tr>
<tr><td>11</td><td>Supported</td><td>Supported</td></tr>
</table></td></tr>
<tr><td>15</td><td>RO</td><td>0</td><td>Reserved.</td></tr>
</table>

Bit	Type	Default	Description
14:12	RW	001b	**Flash Access Channel Maximum Read Request Size:** eSPI master sets the maximum read request size for the Flash Access channel. In the master attached flash sharing configuration, the slave as the Requester must not generate read request with size exceeding the set value. In the slave attached flash sharing configuration, the slave as the Target must handle read request with size as large as the set value. The value set by the eSPI master must never be more than the value advertised in the Target Maximum Read Request Size Supported field. 000b: Reserved. 001b: 64 bytes max read request size. 010b: 128 bytes max read request size. 011b: 256 bytes max read request size. 100b: 512 bytes max read request size. 101b: 1024 bytes max read request size. 110b: 2048 bytes max read request size. 111b: 4096 bytes max read request size.
11	RW/ RO	HwInit	**Flash Sharing Mode:** When Flash Access channel is enabled, this bit indicates the flash sharing scheme in operation. 0b: Master attached flash sharing. 1b: Slave attached flash sharing. If the slave supports only a single flash sharing scheme, this bit is allowed to be implemented as a Read-Only (RO) bit with the value indicates the supported flash sharing scheme. If the slave supports both flash sharing schemes, this bit must be implemented as a Read-Write (RW) bit where eSPI master will configure the bit accordingly to setup the flash sharing scheme.

Bit	Type	Default	Description
10:8	RW	001b	**Flash Access Channel Maximum Payload Size Selected:** eSPI master sets the maximum payload size for the Flash Access channel. The value set by the eSPI master must never be more than the value advertised in the Max Payload Size Supported field. 000b: Reserved. 001b: 64 bytes max payload size. 010b: 128 bytes max payload size. 011b: 256 bytes max payload size. 100b – 111b: Reserved.
7:5	RO	HwInit	**Flash Access Channel Maximum Payload Size Supported:** This field advertises the Maximum Payload Size supported by the slave. 000b: Reserved. 001b: 64 bytes max payload size. 010b: 128 bytes max payload size. 011b: 256 bytes max payload size. 100b – 111b: Reserved.
4:2	RW	01b	**Flash Block Erase Size:** eSPI master sets this field to communicate the block erase size to the slave. This field is applicable only to master attached flash sharing scheme. 000b: Reserved 001b: 4 Kbytes 010b: 64 Kbytes 011b: Both 4 Kbytes and 64 Kbytes are supported 100b: 128 Kbytes 101b: 256 Kbytes 110b – 111b: Reserved
1	RO	0b	**Flash Access Channel Ready:** When this bit is a '1', it indicates that the slave is ready to accept transactions on the Flash Access channel. eSPI master should poll this bit after the channel is enabled before running any transaction on this channel to the slave. 0b: Channel is not ready. 1b: Channel is ready.

Bit	Type	Default	Description
0	RW	0b	**Flash Access Channel Enable:** This bit is set to '1' by eSPI master to enable the Flash Access channel. Clearing this bit from '1' to '0' triggers a reset to the Flash Access channel such as during error handling. The channel remains disabled until this bit is set to '1' again. The channel is by default disabled after the eSPI Reset#.

2.1.2 Offset 44h: Channel 3 Capabilities and Configurations 2

Table 2-2. Channel 3 Capabilities and Configurations 2

Bit	Type	Default	Description
31:22	RO	0	**Reserved.**
21:16	RO	HwInit	**Target RPMC Supported:** This field indicates the total number of Replay Protected Monotonic Counters (RPMC) supported by the Slave. It is a 1-based field. 0h: Slave does not support RPMC 1h: Slave supports up to 1 RPMC 2h: Slave supports up to 2 RPMC ... 3Fh: Slave supports up to 63 RPMC

Bit	Type	Default	Description
15:8	RO	HwInit	**Target Flash Erase Block Size for Master's Regions:** This field indicates the sizes of the erase commands the master may issue. If multiple bits are set then the master may issue an erase using any of the indicated sizes. If multiple regions are accessible by the master, this field advertises the common erase block sizes for these regions. This field is only applicable when slave attached flash sharing scheme is selected. Bit 0: 1 Kbytes EBS supported Bit 1: 2 Kbytes EBS supported Bit 2: 4 Kbytes EBS supported Bit 3: 8 Kbytes EBS supported Bit 4: 16 Kbytes EBS supported Bit 5: 32 Kbytes EBS supported Bit 6: 64 Kbytes EBS supported Bit 7: 128 Kbytes EBS supported
7:3	RO	0	**Reserved.**
2:0	RO	HwInit	**Target Maximum Read Request Size Supported:** This field indicates the maximum read request size supported by the slave as the Target on the Flash Access channel. This field is only applicable when slave attached flash sharing scheme is selected. 000b, 001b: 64 bytes max read request size. 010b: 128 bytes max read request size. 011b: 256 bytes max read request size. 100b: 512 bytes max read request size. 101b: 1024 bytes max read request size. 110b: 2048 bytes max read request size. 111b: 4096 bytes max read request size.

2.2 Command Opcodes

The Slave Attached Flash Sharing scheme uses 2 dedicated eSPI command opcodes: PUT_FLASH_NP and GET_FLASH_C (Table 2-3).

These command opcodes are initiated utilizing the standard eSPI command phase protocol as described in the eSPI Base Specification.

Table 2-3. Command Opcode Encodings

CMD Opcode	Encoding[7:0]	Description
Flash Access Channel		
PUT_FLASH_NP	00001010	Put a non-posted Flash Access request. Used in Slave Attached Flash Sharing mode for the master to issue a flash access request to the slave. Note: It is illegal to issue a PUT_FLASH_NP unless the slave has indicated that it is free to take the non-posted Flash Access request.
GET_FLASH_C	00001011	Get a Flash Access completion. Used in Slave Attached Flash Sharing mode for the slave to return a flash access completion to the master. Note: It is illegal to issue a GET_FLASH_C unless the slave has indicated that it has a Flash Access completion available to send.

2.3 Cycle Types

The following cycle types (Table 2-4) are added to the Flash Access channel to support the slave attached flash sharing operation.

These cycle types utilize the standard Flash Access packet formats as described in the eSPI Base Specification. The specifics of how the standard eSPI packet format maps for the various flash access types are detailed later in this document.

The direction of cycle type is specified in the table as “Up” or “Down”. “Up” refers to the direction from eSPI slave to eSPI master and “Down” refers to the direction from eSPI master to eSPI slave.

Table 2-4. Cycle Types

Cycle Type	Encodings[3] [7:0]	Direction	Command Type	Channel Type	Description
Flash Access Channel[4]					

Cycle Type	Encodings[3] [7:0]	Direction	Command Type	Channel Type	Description
Successful Completion Without Data	00000110	Up	Completion	Flash Access Channel	Successful Completion Without Data. Corresponds to Flash Write or Flash Erase.
Successful Completion With Data	00001P_1P_0[1]1	Up	Completion	Flash Access Channel	Successful Completion With Data. Corresponds to Flash Read.
Unsuccessful Completion Without Data	00001P_1P_0[1,2]0	Up	Completion	Flash Access Channel	Unsuccessful Completion Without Data. Corresponds to Flash accesses.
Flash Read	00000000	Down	Non-Posted	Flash Access Channel	Read from Flash.
Flash Write	00000001	Down	Non-Posted	Flash Access Channel	Write to Flash.
Flash Erase	00000010	Down	Non-Posted	Flash Access Channel	Flash Erase instruction. Erase part or the whole partition owned by the corresponding flash master.
RPMC Op.1	00000011	Down	Non-Posted	Flash Access Channel	Replay Protected Monotonic Counter (RPMC) Opcode 1
RPMC Op.2	00000100	Down	Non-Posted	Flash Access Channel	Replay Protected Monotonic Counter (RPMC) Opcode 2

Note:

1. The encoding P_1P_0 has the following definition:

Encoding P_1P_0	Description
00	Indicates the middle completion of a split completion sequence.
01	Indicates the first completion of a split completion sequence
10	Indicates the last completion of a split completion sequence.
11	Indicates the only completion for a split transaction.

2. For Unsuccessful Completion without Data, P_1 must be always a ‘1’ as this is always the last or the only completion.

3. The combination of command opcode and cycle type encoding must be unique. There is no requirement that cycle type encodings must be unique across command opcodes.

4. Refer to the eSPI Base Specification for detail operation of the Flash Access channel.

2.4 Response Modifier

The Response Modifier is a 2-bit field defined for the GET_STATUS with an ACCEPT response only as described in the eSPI Base Specification.

When slave attached flash sharing is supported and in operation, the Response Modifier with a value of 11b allows a Flash Access (channel 3) completion to be appended to the response phase.

Table 2-5. Response Modifier (R_1R_0) for GET_STATUS

RESPONSE	Encoding			Description
	[7:6]	[5:4]	[3:0]	
ACCEPT	R_1R_0[1]	RSV	1000	Command was successfully received If the command was a PUT_NP, a response of ACCEPT means that the non-posted transaction is being completed as a “connected” transaction.

NOTES:

1. The Response Modifier R_1R_0 has the following definition:

Encoding[7:6] R_1R_0	Description
00	No append.
01	A Peripheral (channel 0) completion is appended.
10	A Virtual Wire (channel 1) packet is appended.
11	A Flash Access (channel 3) completion is appended. This is only applicable when slave attached flash sharing is supported and in operation.

2.5 Status

Slave’s Status Register bit[9] (FLASH_NP_FREE) and bit[12] (FLASH_C_AVAIL) are defined to support the Slave Attached Flash Sharing operation.

These bits are don't care and ignored by the eSPI master when Flash Access channel is not enabled, or Slave Attached Flash Sharing scheme is not in operation.

Figure 2-2. Slave's Status Register Definition

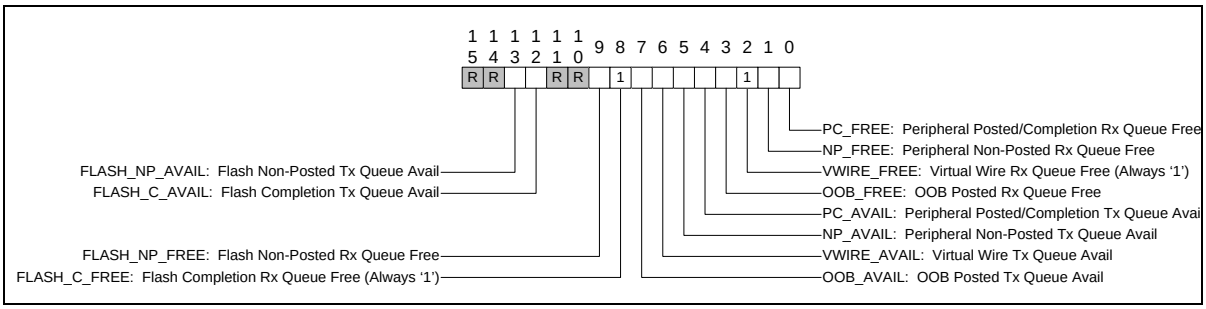


Table 2-6. Status Field Encodings

STATUS	Bits Position	Description
Slave's Rx queues Free		
FLASH_NP_FREE	9	When '1', indicates the slave is free to accept at least one channel 3 Flash Access non-posted header and data up to maximum payload size. This bit is only applicable when slave attached flash sharing is supported and in operation. Otherwise, the bit is a don't care.
Slave's Tx queues Available		
FLASH_C_AVAIL	12	When '1', indicates the slave has a channel 3 Flash Access completion header and data up to maximum payload size available to send. This bit is only applicable when slave attached flash sharing is supported and in operation. Otherwise, the bit is a don't care.

2.6 Slave Attached Flash Sharing Operation

The eSPI Flash Access Channel in this scheme defines a set of Standard flash commands. These commands reflect typical accesses supported by most flash devices.

In addition, up to 16 Platform-Specific flash commands maybe defined for specific flash devices. Platform-Specific commands will be published in a separate Compatibility Specification for a given platform outside the scope of the eSPI specification. Table 2-7 lists the Flash Operations that are supported in Master and Slave Attached flash configurations over the eSPI protocol.

For all the Standard and Platform-Specific commands, the flash device owner is responsible for handling the differences between the different flash vendors and low level flash access operations, making them transparent to the eSPI Master.

The flash device owner is also responsible for managing the various flash access parameters, including, but not limited to flash Single/Dual/Quad IO mode for Opcode/Address/Data phases, flash Mode cycles, flash addressing mode (24-bit vs. 32-bit), flash Wait cycles (bus turnaround time), flash command sequencing (e.g., Write Enable prior to a Write), flash Suspend/Resume (to manage QoS).

The Address and Length fields ({Length1[3:0], Length0[7:0]}) of the standard eSPI packet format will have the definitions as specified in Table 2-7. The Length field will specify the length of the write data to the flash device or requested size of the read data from the flash device (as described in the eSPI Base Specification). The only exception will be for the flash erase block sizes, where the length encodings are as listed in Table 2-7.

The eSPI Master will specify the address for all flash accesses in 32-bit format (4 bytes). The BMC is responsible for determining whether the flash device supports 24-bit or 32-bit addressing and driving the appropriate address length to the flash device. For cases where the flash device supports or is programmed for 24-bit access, or the requested flash operation does not support 32 bit addresses, the BMC will ignore the most significant byte (Byte 3) of the address in the eSPI packet transmitted by the eSPI Master for its flash operation.

All eSPI Master accesses to the flash will specify the physical address to the device based on the allocated flash regions. The exact method of region allocation is platform specific and outside the scope of the eSPI specification.

Table 2-7. eSPI Flash Access Channel Packet Format for Master and Slave Attached Flash Configurations

Cycle Type [7:0]	Flash Command Type	Flash Operation	Address Size	Length [11:0] (3)	Slave Attached Flash Supported	Master Attached Flash Supported
00h	Standard	Read	4 B	≤ Max Read Request Size	Yes	Yes

01h	Standard	Write	4 B	≤ Max Payload Size	Yes (1)	Yes (1)
02h	Standard	Erase	4 B	0h: 4 KB 1h: 32 KB 2h: 64 KB 3h – FFFh: Reserved	Yes (1)	Yes (1)
03h	Standard	RPMC Op.1	N/A	≤ Max Payload Size	Yes (2)	No
04h	Standard	RPMC Op.2	N/A	≤ Max Read Request Size	Yes	No
05h, 07h, 09h – 2Fh	Standard	Reserved	Command Specific	Command Specific	Command Specific	Command Specific
30h – 3Fh	Platform-Specific	Reserved	Command Specific	Command Specific	Command Specific	Command Specific

1. The device that owns the flash is responsible for performing the Write Enable prior and the Read Status polling after the Write/Erase operation. The eSPI completion response for a flash Write/Erase operation will indicate the result after the Read Status polling has completed.
2. The device that owns the flash is responsible for performing the Read Status polling atomically after the RPMC Op.1 operation (i.e. no other flash operations can be performed between the RPMC Op.1 and until the Read Status polling has completed). The eSPI completion response for a flash RPMC Op.1 operation will indicate the result after the Read Status polling has completed.
3. In slave attached flash sharing configuration, the Max Read Request Size must never be more than the value advertised by the slave in the Target Maximum Read Request Size Supported field. The length field with a value of '0' indicates 4KB of length.

All the flash operations from eSPI Master are non-posted transactions. Each of the transactions will have a corresponding completion which indicates the status of the requested operation, together with data if the cycle is a flash access that returns data from the flash. The status of the completion will be conveyed back to the eSPI Master.

The BMC shall opportunistically exercise flash Suspend/Resume capability to speed up low latency commands such as reads by interleaving them within high latency operations such as writes and erases when the address ranges are non-overlapping.

The BMC is required to maintain a separate queue for flash access commands from eSPI Master to potentially improve the QoS for the flash access operations. The recommended queue depth will be specified separately in the Compatibility Specification for a given platform (typically a 2 to 4 deep queue will suffice). When the eSPI Master issues a flash access command, the Slave (BMC) will return its status for

FLASH_NP_FREE as True unless its corresponding queue is full. This will allow the eSPI Master to issue multiple outstanding flash operations into the queue.

An example of the flash access command queue within a BMC is shown in Table 2-8.

Table 2-8. Example eSPI Slave Attached Flash Access Command Sequence

Command #	Flash Access Command from eSPI Master
4	Write
3	RPMC Op.1
2	Read 4 B, Address A2
1	Erase 64 KB, Address A1

Since commands #1 and #2 do not overlap in the flash, the BMC can schedule them to the flash as it deems best for improving the eSPI Master's flash access QoS. For example, if command #2 is received soon after command #1 has been issued to the flash, the BMC can choose to put the Flash in Suspend/Resume to interrupt the Erase 64 KB command and service the Read 4 B read command. For the RPMC Op.1 command (#3), the BMC must issue and complete the subsequent Read Status polling atomically before allowing any other flash operations (from the Master or the BMC) to be issued to the flash. Similarly for the Write, the BMC must first issue a Write Enable to the flash device, complete the Write and the subsequent Read Status polling before sending the completion back to the Master.

Flash Access Channel Maximum Read Request Size parameter in the Channel Capability and Configuration register is defined to allow the eSPI Master to limit the Flash Read request.

Similarly, Flash Access Channel Maximum Payload Size parameter in the Channel Capability and Configuration register is defined to allow the eSPI Master to limit the Flash Write data payload size.

If the flash access command from eSPI master is not supported due to invalid addressing mode (32-bit versus 24/26-bit addressing), unsupported command, unsupported block erase size or any other reasons, the eSPI Slave (BMC) is responsible for the error detection and handling as described in the eSPI Base Specification.

§