



# Intel® 8 Series Chipset Family

## SPI Programming Guide

---

*January 2013*

Revision 1.5

**Intel Confidential**



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions 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. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: [http://www.intel.com/products/processor\\_number](http://www.intel.com/products/processor_number).

Intel® Hyper-Threading Technology requires an Intel® HT Technology enabled system, check with your PC manufacturer. Performance will vary depending on the specific hardware and software used. Not available on Intel® Core™ i5-750. For more information including details on which processors support HT Technology, visit <http://www.intel.com/info/hyperthreading>.

Intel® 64 requires a computer system with a processor, chipset, BIOS, operating system, device drivers, and applications enabled for Intel 64. Processor will not operate (including 32-bit operation) without an Intel 64-enabled BIOS. Performance will vary depending on your hardware and software configurations. See <http://www.intel.com/info/em64t> for more information including details on which processors support Intel 64, or consult with your system vendor for more information.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Consult your PC manufacturer. For more information, visit: <http://www.intel.com/go/virtualization>.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2012-2013, Intel Corporation. All rights reserved.



## Contents

---

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Overview	7
1.2	Terminology	8
1.3	Reference Documents	8
	<b>PCH SPI Flash Architecture</b>	<b>9</b>
2.1	Descriptor Mode	9
2.2	SFDP (Serial Flash Discoverable Parameter)	9
2.3	SPI Fast Read	9
2.4	Intel® TPM on SPI Bus	9
2.5	Boot Destination Option	10
2.6	Flash Regions	10
2.7	Hardware vs. Software Sequencing	11
<b>3</b>	<b>PCH SPI Flash Compatibility Requirement</b>	<b>13</b>
3.1	Intel® 8 Series Chipset Family SPI Flash Requirements	13
3.2	Intel® 8 Series Chipset Family SPI AC Electrical Compatibility Guidelines	17
3.3	SPI Flash DC Electrical Compatibility Guidelines	19
<b>4</b>	<b>Descriptor Overview</b>	<b>21</b>
4.1	Flash Descriptor Content	23
4.2	OEM Section	34
4.3	Region Access Control	34
4.4	Intel® Management Engine (Intel® ME) Vendor-Specific Component Capabilities Table	36
<b>5</b>	<b>Serial Flash Discoverable Parameter (SFDP) Overview</b>	<b>39</b>
5.1	Introduction	39
5.2	Discoverable Parameter Opcode and Flash Cycle	39
5.3	Parameter Table Supported on PCH	40
5.4	Detail JEDEC Specification	40
<b>6</b>	<b>Configuring BIOS/GbE for SPI Flash Access</b>	<b>41</b>
6.1	Unlocking SPI Flash Device Protection for Intel® 8 Series Chipset Family Platforms	41
6.2	Locking SPI Flash via Status Register	42
6.3	SPI Protected Range Register Recommendations	42
6.4	Software Sequencing Opcode Recommendations	43
6.5	Recommendations for Flash Configuration Lockdown and Vendor Component Lock Bits	44
6.6	Host Vendor Specific Component Control Registers (VSCC) for Intel® 8 Series Chipset Family Systems	44
6.7	Host VSCC Register Settings for Intel® 8 Series Chipset Family Systems	49
<b>7</b>	<b>Flash Image Tool</b>	<b>51</b>
7.1	Flash Image Details	51
7.2	Modifying the Flash Descriptor Region	52
7.3	PCH Soft Straps	55
7.4	Management Engine VSCC Table	55
<b>8</b>	<b>Flash Programming Tool</b>	<b>59</b>
8.1	BIOS Support	59
8.2	Fparts.txt File	59
8.3	Configuring a Fparts.txt Entry	60
<b>9</b>	<b>SPI Flash Programming Procedures</b>	<b>63</b>
9.1	Updating BIOS	63



<b>10</b>	<b>Intel® Management Engine Disable for Debug/Flash Burning Purposes</b>	<b>65</b>
10.1	Intel® ME Disable	65
<b>11</b>	<b>Recommendations for SPI Flash Programming in Manufacturing Environments for Intel® 8 Series Chipset Family</b>	<b>67</b>
<b>12</b>	<b>FAQ and Troubleshooting</b>	<b>69</b>
12.1	FAQ	69
12.2	Troubleshooting	71
<b>A</b>	<b>APPENDIX A - Descriptor Configuration</b>	<b>73</b>
A.1	Flash Descriptor PCH Soft Strap Section	73
A.2	<a href="#">PCHSTRP0—Strap 0 Record (Flash Descriptor Records)</a>	74
A.3	PCHSTRP1—Strap 1 Record (Flash Descriptor Records)	77
A.4	PCHSTRP2—Strap 2 Record (Flash Descriptor Records)	78
A.5	PCHSTRP3—Strap 3 Record (Flash Descriptor Records)	80
A.6	PCHSTRP4—Strap 4 Record (Flash Descriptor Records)	80
A.7	PCHSTRP5—Strap 5 Record (Flash Descriptor Records)	81
A.8	PCHSTRP6—Strap 6 Record (Flash Descriptor Records)	81
A.9	PCHSTRP7—Strap 7 Record (Flash Descriptor Records)	82
A.10	PCHSTRP8—Strap 8 Record (Flash Descriptor Records)	82
A.11	<a href="#">PCHSTRP9—Strap 9 Record (Flash Descriptor Records)</a>	83
A.12	PCHSTRP10—Strap 10 Record (Flash Descriptor Records)	86
A.13	PCHSTRP11—Strap 11 Record (Flash Descriptor Records)	87
A.14	PCHSTRP12—Strap 12 Record (Flash Descriptor Records)	88
A.15	<a href="#">PCHSTRP13—Strap 13 Record (Flash Descriptor Records)</a>	88
A.16	PCHSTRP14—Strap 14 Record (Flash Descriptor Records)	88
A.17	<a href="#">PCHSTRP15—Strap 15 Record (Flash Descriptor Records)</a>	89
A.18	<a href="#">PCHSTRP16—Strap 16 Record (Flash Descriptor Records)</a>	90
A.19	<a href="#">PCHSTRP17—Strap 17 Record (Flash Descriptor Records)</a>	90
A.20	<a href="#">PCHSTRP18—Strap 18 Record (Flash Descriptor Records)</a>	91
A.21	<a href="#">PCHSTRP19—Strap 19 Record (Flash Descriptor Records)</a>	91
A.22	<a href="#">PCHSTRP20—Strap 20 Record (Flash Descriptor Records)</a>	91
A.23	<a href="#">CPUSTRP0—Strap 0 Record (Flash Descriptor Records)</a>	92
A.24	Softstrap Step Through	93



## Figures

3-1	SPI Timing .....	18
3-2	PCH Test Load .....	19
4-1	Flash Descriptor (Lynx Point) .....	21
5-1	SFDP Read Instruction Sequence.....	39
7-1	Firmware Image Components .....	51
7-2	Editable Flash Image Region List .....	52
7-3	Descriptor Region – Descriptor Map Options .....	53
7-4	Descriptor Region – Fast Read Support Options.....	53
7-5	Descriptor Region - Component Section Options.....	54
7-6	Region Access Control .....	54
7-7	Descriptor Region – Master Access Section Options.....	55
7-8	Add New VSCC Table Entry.....	56
7-9	Add VSCC Table Entry .....	56
7-10	VSCC Table Entry.....	56
7-11	Remove VSCC Table Entry.....	57

## Tables

1-1	Terminology .....	8
1-2	Reference Documents.....	8
2-1	Region Size vs. Erase Granularity of Flash Components .....	11
3-1	SPI Timings (20 MHz) .....	17
3-2	SPI Timings (33 MHz) .....	17
3-3	SPI Timings (50 MHz) .....	18
4-1	Region Access Control Table Options.....	34
4-2	Recommended Read/Write Settings for Platforms Using Intel® ME Firmware .....	35
4-3	Recommended Read/Write Settings for Platforms Using Intel® ME Firmware (Cont'd) .....	35
4-4	Jidn - JEDEC ID Portion of Intel® ME VSCC Table.....	36
4-5	Vscn - Vendor-Specific Component Capabilities Portion of the Lynx Point Family Platforms .....	37
6-1	Recommended Opcodes for FPT Operation.....	43
6-2	Recommended Opcodes for FPT Operation.....	43
6-3	VSCC0 - Vendor-Specific Component Capabilities Register for SPI Component 0 .....	45
6-4	VSCC1 - Vendor Specific Component Capabilities Register for SPI Component 1 .....	47
6-5	Description of How WSR and WEWS is Used.....	49



## Revision History

Document Number	Revision Number	Description	Revision Date
489495	0.5	<ul style="list-style-type: none"> <li>Initial release.</li> </ul>	September 2011
	0.7	<ul style="list-style-type: none"> <li>Lynx Point softstraps updated.</li> </ul>	March 2012
	0.7.1	<ul style="list-style-type: none"> <li>Updated trademarks and branding throughout document (no change bars) Note: Change bars from Rev. 0.7 updates were left in place.</li> </ul>	April 2012
	0.8	<ul style="list-style-type: none"> <li>Updated USB/PCIe mux ports strapping option, added timer t205b into PCHSTRP 9 and added Fidelix into vendor list.</li> <li>Added Quad Enable Requirement (QER) for VSCC0 and VSCCn.</li> </ul>	May 2012
	0.81	<ul style="list-style-type: none"> <li>Updated PCHSTRP 17 bit 1 setting.</li> </ul>	June 2012
	0.9	<ul style="list-style-type: none"> <li>Removed example of VSCC Table Settings from section 4.4.3 and 6.7, SPI programming guide user can refer to VSCCommn.bin Content document.</li> <li>Updated ME related PCHSTRP10 and default settings for PCHSTRP1</li> <li>SMLink1 supports standard mode (100kHz) only, all others reserved.</li> <li>Updated official naming for Lynx Point Chipset.</li> <li>Added Appendix B - Chipset Initialization Table</li> </ul>	Sept 2012
	1.0	<ul style="list-style-type: none"> <li>Included CPUSTRP 0 in Appendix A.</li> <li>Updated FLMAP2 information and default value settings.</li> <li>Updated screenshots of FITC in chapter 7</li> <li>Removed non descriptor operational mode from section 2.1 and section 9.1.2</li> </ul>	Oct 2012
	1.1	<ul style="list-style-type: none"> <li>Corrected data value as 0x4F003F00 for Chipset register 0xE900C17C</li> <li>Note: Change bars from Rev 1.0 updates were left in place.</li> </ul>	Oct 2012
	1.5	<ul style="list-style-type: none"> <li>Update PCHSTRP15 [11:8] timing name to be consistent with EDS.</li> <li>Change PCHSTRP15 [19] to reserved</li> <li>Update PCHSTRP9 [21:18] USB port to be consistent with EDS.</li> <li>Removed Appendix B - Chipset Initialization Table</li> </ul> <p><b>Note:</b> Chipset Initialization Settings have been moved from the descriptor region into the ME Region. Bios will automatically synchronize settings in the ME region when required.</p>	January 2013

§ §



# 1 Introduction

---

## 1.1 Overview

This manual is intended for Original Equipment Manufacturers and software vendors to clarify various aspects of programming SPI flash on PCH family based platforms. The current scope of this document is Intel® 8 Series Chipset Family only. **This Document is not relevant to platforms running on Server Platform Services (SPS) firmware.**

### Chapter , “PCH SPI Flash Architecture”

- Overview of SPI flash, Non-Descriptor vs. Descriptor, Flash Layout, Intel® 8 Series Chipset Family compatible SPI flash.

### Chapter 3, “PCH SPI Flash Compatibility Requirement”

- Overview of compatibility requirements for Intel® 8 Series Chipset Family products.

### Chapter 4, “Descriptor Overview”

- Overview of the descriptor and Descriptor record definition

### Chapter 5, “Serial Flash Discoverable Parameter (SFDP) Overview”

- Overview of the SFDP definition.

### Chapter 6, “Configuring BIOS/GbE for SPI Flash Access”

- Describes how to configure BIOS/GbE for SPI flash access.

### Chapter 7, “Flash Image Tool”

- This tool creates a descriptor and combines the GBE, BIOS, Platform Data Region and Intel® ME (Intel® ME) firmware into one image.

### Chapter 8, “Flash Programming Tool”

- This tool programs the SPI flash device on the Intel® 8 Series Chipset Family platforms. This section will talk about requirements needed for FPT to work.

### Chapter 9, “SPI Flash Programming Procedures”

- Guide on how to program the SPI flash on the Intel CRB and PCH based platforms.

### Chapter 10, “Intel® Management Engine Disable for Debug/Flash Burning Purposes”

- Methods of disabling Intel Management Engine for debug purposes.

### Chapter 11, “Recommendations for SPI Flash Programming in Manufacturing Environments for Intel® 8 Series Chipset Family”

- Recommendations for manufacturing environments.

### Chapter 12, “FAQ and Troubleshooting”

- Frequently asked questions and Troubleshooting tips.



## 1.2 Terminology

Table 1-1. Terminology

Term	Description
BIOS	<u>B</u> asic <u>I</u> nput- <u>O</u> utput <u>S</u> ystem
CRB	<u>C</u> ustomer <u>R</u> eference <u>B</u> oard
FPT	<u>F</u> lash <u>P</u> rogramming Tool - programs the SPI flash
FIT	<u>F</u> lash <u>I</u> mage <u>T</u> ool – creates a flash image from separate binaries
FW	<u>F</u> irm <u>w</u> are
FWH	<u>F</u> irm <u>w</u> are <u>H</u> ub – LPC based flash where BIOS may reside
Intel <sup>®</sup> AMT	Intel <sup>®</sup> Active Management Technology
GbE	Intel Integrated 1000/100/10
HDCP	High-bandwidth Digital Content Protection
Lynx Point	Intel <sup>®</sup> 8 Series Chipset Family, code named Lynx Point
Intel <sup>®</sup> ME Firmware	Intel firmware that adds Intel <sup>®</sup> Active Management Technology, Braidwood Technology, Intel Anti-Theft Technology, Corwin Springs, Castle Peak, Sentry Peak, etc.
Intel PCH	<u>I</u> ntel <u>P</u> latform <u>C</u> ontroller <u>H</u> ub
Intel PCHn family	All PCHn derivatives including PCHn (desktop) and PCHnM (mobile)
LPC	<u>L</u> ow <u>P</u> in <u>C</u> ount Bus- bus on where legacy devices such a FWH reside
SPI	<u>S</u> erial <u>P</u> eripheral <u>I</u> nterface – refers to serial flash memory in this document
VSCC	<u>V</u> endor <u>S</u> pecific <u>C</u> omponent <u>C</u> apabilities
LVSCC	<u>L</u> ower <u>V</u> endor <u>S</u> pecific <u>C</u> omponent <u>C</u> apabilities
UVSCC	<u>U</u> pper <u>V</u> endor <u>S</u> pecific <u>C</u> omponent <u>C</u> apabilities
SFDP	Serial Flash Discoverable Parameter

## 1.3 Reference Documents

Table 1-2. Reference Documents

Document	Document # / Location
<i>Intel<sup>®</sup> 8 Series Chipset Family External Design Specification (EDS)</i>	Contact Intel field representative
<i>Intel Flash Image Tool (FIT)</i>	\\System Tools\\Flash Image Tool of latest <u>Intel<sup>®</sup> ME</u> kit from VIP/ARMS. The Kit MUST match the platform you intend to use the flash tools for.
<i>Intel Flash Programming Tool (FPT)</i>	\\System Tools\\Flash Programming Tool of latest <u>Intel<sup>®</sup> ME</u> from VIP/ARMS. The Kit MUST match the platform you intend to use the flash tools for.
<i>FW Bring Up Guide</i>	Root directory of latest <u>Intel ME</u> kit from VIP/ARMS. The Kit MUST match the platform you intend to use the flash tools for.

§ §





## 2 PCH SPI Flash Architecture

---

PCH SPI interface consists of clock (CLK), MOSI (Master Out Slave In) MISO (Master In Slave Out), IO2, IO3 (For Quad Fast Read and Quad I/O support) and up to 3 active low chip selects (CSX#) on Intel® 8 Series Chipset Family. Chip Select 3 (CS3#) was for TPM on SPI

Intel® 8 Series Chipset Family can support SPI flash devices up to 64 Mbytes per chip select. Intel® 8 Series Chipset Family can support frequencies of 20 MHz, 33 MHz, and 50 MHz.

### 2.1 Descriptor Mode

**Intel® 8 Series Chipset Family supports descriptor mode only.** Non-descriptor mode is not supported.

Descriptor mode supports up to two SPI flashes, and allows for integrated LAN support, as well as Intel ME firmware to share a single flash. There is also additional security for reads and writes to the flash. Hardware sequencing, heterogeneous flash space, Intel integrated LAN, Intel ME firmware on SPI flash, require descriptor mode. Descriptor mode requires the SPI flash to be hooked up directly to the PCH's SPI bus.

See [SPI Supported Feature Overview](#) of the latest *Intel Platform Controller Hub Family External Design Specification (EDS)* for Intel® 8 Series Chipset Family for more detailed information.

### 2.2 SFDP (Serial Flash Discoverable Parameter)

Intel® 8 Series Chipset Family supports SPI with SFDP. SFDP (Serial Flash Discoverable Parameter) is a JEDEC standard provides a consistent method of describing the functional and feature capabilities of SPI devices in a standard set of internal parameter tables. These parameter tables can be interrogated by PCH to enable adjustment needed to accommodate divergent feature from multiple vendors.

Please refer to [Chapter 5, "Serial Flash Discoverable Parameter \(SFDP\) Overview"](#) for more information.

### 2.3 SPI Fast Read

Intel® 8 Series Chipset Family supports SPI **Dual output, Dual I/O, Quad output and Quad I/O Fast read** instruction with frequencies 20, 33, and 50 MHz.

**Note:** 50Mhz support requires SPI component that meet 66Mhz timing.

### 2.4 Intel® TPM on SPI Bus

Intel® 8 Series Chipset Family supports TPM on the SPI bus. TPM attached to the system may be using LPC or SPI. SPI TPM is accessed much like direct reads and direct writes.



## 2.5 Boot Destination Option

### 2.5.1 Boot Flow for Intel® 8 Series Chipset Family

When booting from Global Reset, the PCH SPI controller will check whether the SPI component is supporting SFDP by sending 5Ah to SPI to CS0 first then CS1. SFDP fetching will triggered when assertion of MEPWROK. If the SPI has a valid SFDP, the controller supports auto discovery of the Component Property Parameter Table (CPPT) which is having CPPT set to 1. CPPT is located at VSCC0 (for SPI 0) and VSCC1 (for SPI 1). Next, the SPI controller will look for a descriptor signature on the SPI flash device on Chip Select 0 at address 0x10. If the signature is present and valid, then the PCH controller will boot in Descriptor mode. It will load up the descriptor into corresponding registers in the PCH. If the signature is NOT present the PCH will boot in non descriptor mode where integrated LAN and all Intel Management Firmware will be disabled. Even if the PCH boot in non descriptor mode, SFDP parameters are available for software use. Whether there is a valid descriptor or not, the PCH will look to the BIOS boot straps to determine the location of BIOS for host boot.

See Boot BIOS strap in the [Functional Straps](#) of the latest *Intel I/O Controller Hub Family External Design Specification (EDS)* for Intel® 8 Series Chipset Family for more detailed information.

If LPC is chosen as the BIOS boot destination, then the PCH will fetch the reset vector on top of the firmware hub flash device.

If SPI is chosen as the BIOS destination, it will either fetch the reset vector on top of the SPI flash device on chip select 0, or if the PCH is in descriptor mode it will determine the location of BIOS through the base address that is defined in the SPI flash descriptor.

See [Chapter 4, "Descriptor Overview"](#) and for more detailed information.

## 2.6 Flash Regions

Flash Regions only exist in Descriptor mode. The controller can divide the SPI flash in up to five separate regions.

Region	Content
0	Descriptor
1	BIOS
2	ME – Intel® Management Engine Firmware
3	GbE – Location for Integrated LAN firmware and MAC address
4	PDR – Platform Data Region

The descriptor (Region 0) must be located in the first sector of component 0 (offset 0x10). Descriptor and ME regions are required for all Intel® 8 Series Chipset Family based platforms.

If Regions 0, 2, 3 or 4 are defined they must be on SPI. BIOS can be on either FWH or SPI. The BIOS that will load on boot will be set by Boot BIOS destination straps.



Only three masters can access the five regions: Host CPU, integrated LAN, and Intel ME.

## 2.6.1 Flash Region Sizes

SPI flash space requirements differ by platform and configuration. Please refer to documentation specific to your platform for BIOS and ME Region flash size estimates.

The Flash Descriptor requires one block. GbE requires two separate blocks. The amount of actual flash space consumed for the above regions are dependent on the erase granularity of the flash part. Assuming 2 Mbyte BIOS, 64 Mb flash part is the target size of flash for largest configuration. BIOS size will determine how small of a flash part can be used for the platform.

## 2.7 Hardware vs. Software Sequencing

**Table 2-1. Region Size vs. Erase Granularity of Flash Components**

Regions	Size with uniform 4 KB blocks
Descriptor	4 KB
GbE	8 KB
Platform Data Region	Varies by platform
BIOS	Varies by platform
ME	Varies by platform and configuration

Hardware and software sequencing are the two methods the PCH uses communicates with the flash via programming registers for each of the three masters.

When utilizing software sequencing, BIOS needs to program the OPTYPE and OPMENU registers respectively with the opcode it needs. It also defines how the system should use each opcode. If the system needs a new opcode that has not been defined, then BIOS can overwrite the OPTYPE and OPMENU register and define new functionality as long as the FLOCKDN bits have not been set.

FPT as well as some BIOS implementation support software sequencing. Note: FPT defaults to hardware sequencing.

Hardware sequencing has a predefined list of opcodes with only the erase opcode being programmable. This mode is only available if the descriptor is present and valid. Intel® ME Firmware and Integrated LAN FW, and integrated LAN drivers all must use HW sequencing, so BIOS must properly set up the PCH to account for this. The Host VSCC registers and Management Engine VSCC table have to be correctly configured for BIOS, GbE and Intel ME Firmware to have read/write access to SPI.

See [Serial Peripheral Interface Memory Mapped Configuration Registers](#) in *Intel® 8 Series Chipset Family External Design Specification (EDS)* for more details.

§ §





## 3 PCH SPI Flash Compatibility Requirement

---

### 3.1 Intel® 8 Series Chipset Family SPI Flash Requirements

- Intel® 8 Series Chipset Family allows for up to two SPI flash devices to store BIOS, Intel ME Firmware and security keys for Platform Data Region and integrated LAN information.
  - **Intel ME FW is required for Intel® 8 Series Chipset Family based platforms!**
  - Each SPI component can support up to 64MB (128MB total addressable) using 26-bit addressing
- 3.3V SPI I/O buffer VCC
- SPI Fast Read instruction is supported and frequency of 20Mhz, 33Mhz and 50Mhz
  - 50 MHz support requires component that meet 66Mhz timing
- SPI Dual Output and Dual I/O Fast Read instruction is supported with frequency of 20Mhz, 33Mhz and 50 MHz
- SPI Quad Output and Quad I/O Fast read instruction is supported with frequency of 20Mhz, 33Mhz and 50Mhz

If there are two SPI components, both components have to support fast read in order to enable Fast Read in PCH.

#### 3.1.1 SPI-based BIOS Requirements

- Erase size capability of: 4 KBytes.
- Serial flash device must ignore the upper address bits such that an address of FFFFFFFh aliases to the top of the flash memory.
- SPI Compatible Mode 0 support: Clock phase is 0 and data is latched on the rising edge of the clock.
- If the device receives a command that is not supported or incomplete (less than 8 bits), the device must discard the cycle gracefully without any impact on the flash content.
- An erase command (page, sector, block, chip, etc.) must set all bits inside the designated area (page, sector, block, chip, etc.) to 1 (Fh).
- Status Register bit 0 must be set to 1 when a write, erase or write to status register is in progress and cleared to 0 when a write or erase is NOT in progress.
- Devices requiring the Write Enable command must automatically clear the Write Enable Latch at the end of Data Program instructions.
- Byte write must be supported. The flexibility to perform a write between 1 byte to 64 bytes is recommended.



- SPI flash parts that do not meet Hardware sequencing command set requirements may work in BIOS only platforms via software sequencing.

### 3.1.2 Integrated LAN Firmware SPI Flash Requirements

A serial flash device that will be used for system BIOS and Integrated LAN or Integrated LAN only must meet all the SPI Based BIOS Requirements plus:

Must support “[Hardware Sequencing Requirements](#)”.

4 KBytes erase capability must be supported.

#### 3.1.2.1 SPI Flash Unlocking Requirements for Integrated LAN

BIOS must ensure there is no SPI flash based read/write/erase protection on the GbE region. GbE firmware and drivers for the integrated LAN need to be able to read, write and erase the GbE region at all times.

### 3.1.3 Intel® Management Engine (Intel® ME) Firmware SPI Flash Requirements

Intel Management Firmware must meet the SPI flash based BIOS Requirements plus:

[3.1.4 SFDP](#)

[3.1.5 JEDEC ID \(Opcode 9Fh\)](#)

[3.1.6 Multiple Page Write Usage Model](#)

[3.1.7 Hardware Sequencing Requirements](#)

Flash part must be uniform 4 KB erasable block throughout the entire part.

Write protection scheme must meet guidelines as defined in [SPI Flash Unlocking Requirements for Management Engine](#).

SPI Flash Unlocking Requirements for Management Engine.

Flash devices must be globally unlocked (read, write and erase access on the ME region) from power on by writing 00h to the flash's status register to disable write protection.

If the status register must be unprotected, it must use the enable write status register command 50h or write enable 06h.

Opcode 01h (write to status register) must then be used to write a single byte of 00h into the status register. This must unlock the entire part. If the SPI flash's status register has non-volatile bits that must be written to, bits [5:2] of the flash's status register must be all 0h to indicate that the flash is unlocked.

If there is no need to execute a write enable on the status register, then opcodes 06h and 50h must be ignored.

After global unlock, BIOS has the ability to lock down small sections of the flash as long as they do not involve the ME or GbE region. See [6.1 Unlocking SPI Flash Device Protection for Intel® 8 Series Chipset Family Platforms](#) and [6.2 Locking SPI Flash via](#)



[Status Register](#) for more information about flash based write/erase protection.

### 3.1.4 SFDP

Serial flash with SFDP have their supported capabilities and commands stored inside the serial flash devices. The controller will discover the attributes needed to operate. Please refer to JEDEC standard Serial Flash Discoverable Parameters in Standard JESD216, for detail instruction and guideline. the document is available on the JEDEC Website [www.jedec.org](http://www.jedec.org).

### 3.1.5 JEDEC ID (Opcode 9Fh)

Since each serial flash device may have unique capabilities and commands, the JEDEC ID is the necessary mechanism for identifying the device so the uniqueness of the device can be comprehended by the controller (master). The JEDEC ID uses the opcode 9Fh and a specified implementation and usage model. This JEDEC Standard Manufacturer and Device ID read method is defined in Standard JESD21-C, PRN03-NV1 and is available on the JEDEC website: [www.jedec.org](http://www.jedec.org).

### 3.1.6 Multiple Page Write Usage Model

Intel platforms have firmware usage models require that the serial flash device support multiple writes to a page (minimum of 512 writes) without requiring a preceding erase command. BIOS commonly uses capabilities such as counters that are used for error logging and system boot progress logging. These counters are typically implemented by using byte-writes to 'increment' the bits within a page that have been designated as the counter. The Intel firmware usage models require the capability for multiple data updates within any given page. These data updates occur via byte-writes without executing a preceding erase to the given page. Both the BIOS and Intel Management Engine firmware multiple page write usage models apply to sequential and non-sequential data writes.

Flash parts must also support the writing of a single bytes 1024 times in a single 256 Byte page without erase. There will be 64 pages where this usage model will occur. These 64 pages will be every 16 Kilo bytes.



### 3.1.7 Hardware Sequencing Requirements

The following table contains a list of commands and the associated opcodes that a SPI-based serial flash device must support in order to be compatible with hardware sequencing.

Commands	OPCODE	Notes
Write to Status Register	01h	Writes a byte to SPI flash's status register. Enable Write to Status Register command must be run prior to this command
Program Data	02h	Single byte or 64 byte write as determined by flash part capabilities and software
Read Data	03h	
Write Disable	04h	
Read Status	05h	Outputs contents of SPI flash's status register
Write Enable	06h	
Fast Read	0Bh	
Enable Write to Status Register	50h or 06h	Enables a bit in the status register to allow an update to the status register
Erase	Programmable/ Discoverable	4 Kbyte erase. Uses the value from SFDP (if available) else value from VSCCn Erase Opcode register value
Chip Erase	C7h and/or 60	
JEDEC ID	9Fh	<a href="#">See Section 3.1.5 for more information</a>
Dual Output Fast Read	3Bh/ Discoverable	Discoverable opcodes are obtained from each component's SFDP table
Dual I/O Fast Read	Discoverable	Opcode is obtained from each component's SFDP table
Quad I/O Fast Read	Discoverable	Opcode is obtained from each component's SFDP table





## 3.2 Intel® 8 Series Chipset Family SPI AC Electrical Compatibility Guidelines

**Table 3-1. SPI Timings (20 MHz)**

Sym	Parameter	Min	Max	Units	Notes
t180a	Serial Clock Frequency - 20MHz Operation	17.06	18.73	MHz	1
t183a	Tco of SPI_MOSI with respect to serial clock falling edge at the host	-5	13	ns	
t184a	Setup of SPI_MISO with respect to serial clock falling edge at the host	16	-	ns	
t185a	Hold of SPI_MISO with respect to serial clock falling edge at the host	0	-	ns	
t186a	Setup of SPI_CS[1:0]# assertion with respect to serial clock rising edge at the host	30	-	ns	
t187a	Hold of SPI_CS[1:0]# assertion with respect to serial clock rising edge at the host	30	-	ns	
t188a	SPI_CLK High time	26.37	-	ns	2
t189a	SPI_CLK Low time	26.82	-	ns	2

**Notes:**

1. Typical clock frequency driven by Intel® 8 Series Chipset Family is 17.86 MHz.
2. Measurement point for low time and high time is taken at .5(VccME3\_3).

**Table 3-2. SPI Timings (33 MHz)**

Sym	Parameter	Min	Max	Units	Notes
t180b	Serial Clock Frequency - 33MHz Operation	29.83	32.81	MHz	1
t183b	Tco of SPI_MOSI with respect to serial clock falling edge at the host	-5	5	ns	
t184b	Setup of SPI_MISO with respect to serial clock falling edge at the host	8	-	ns	
t185b	Hold of SPI_MISO with respect to serial clock falling edge at the host	0	-	ns	
t186b	Setup of SPI_CS[1:0]# assertion with respect to serial clock rising edge at the host	30	-	ns	
t187b	Hold of SPI_CS[1:0]# assertion with respect to serial clock rising edge at the host	30	-	ns	
t188b	SPI_CLK High time	14.88	-	ns	2
t189b	SPI_CLK Low time	15.18	-	ns	2

**Notes:**

1. Typical clock frequency driven by Intel® 8 Series Chipset Family is 31.25 MHz.
2. Measurement point for low time and high time is taken at .5(VccME3\_3).

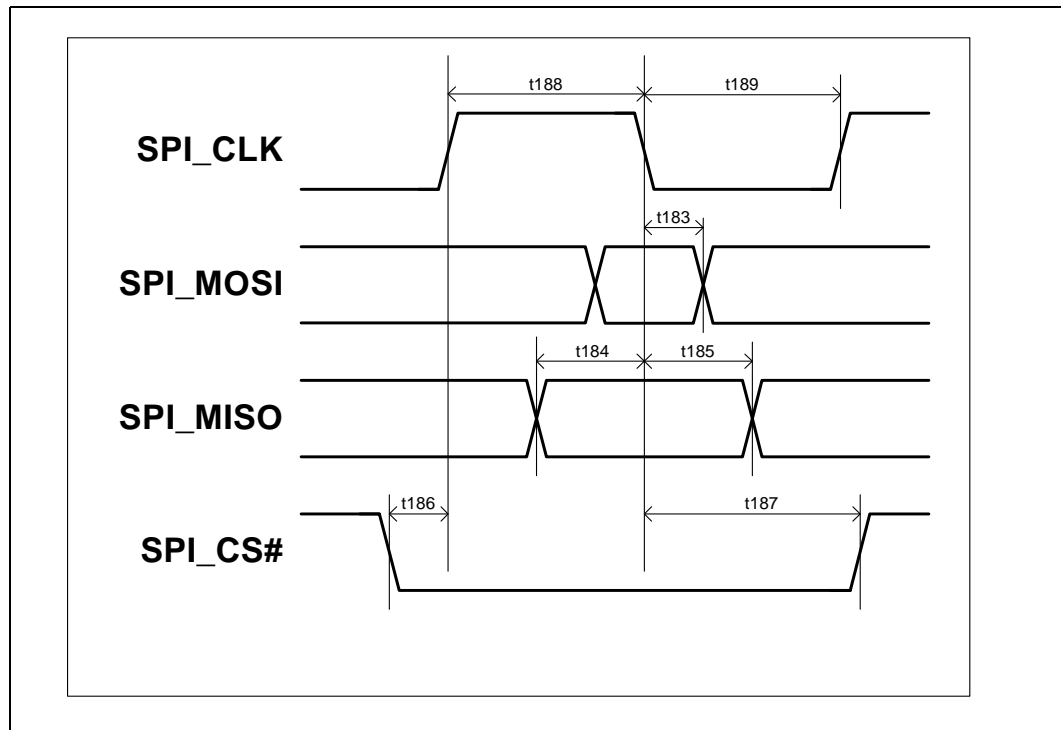
Table 3-3. SPI Timings (50 MHz)

Sym	Parameter	Min	Max	Units	Notes
t180c	Serial Clock Frequency - 50MHz Operation	46.99	53.40	MHz	1
t183c	Tco of SPI_MOSI with respect to serial clock falling edge at the host	-3	3	ns	
t184c	Setup of SPI_MISO with respect to serial clock falling edge at the host	8	-	ns	
t185c	Hold of SPI_MISO with respect to serial clock falling edge at the host	0	-	ns	
t186c	Setup of SPI_CS[1:0]# assertion with respect to serial clock rising edge at the host	30	-	ns	
t187c	Hold of SPI_CS[1:0]# assertion with respect to serial clock rising edge at the host	30	-	ns	
t188c	SPI_CLK High time	7.84	-	ns	2, 3
t189c	SPI_CLK Low time	11.84	-	ns	2, 3

**Notes:**

1. Typical clock frequency driven by Intel® 8 Series Chipset Family is 50 MHz.
2. When using 50 MHz mode ensure target flash component can meet t188c and t189c specifications. Recommended to use SPI flash component rated at 66Mhz or faster.
3. Measurement point for low time and high time is taken at .5(VccME3\_3).

Figure 3-1. SPI Timing



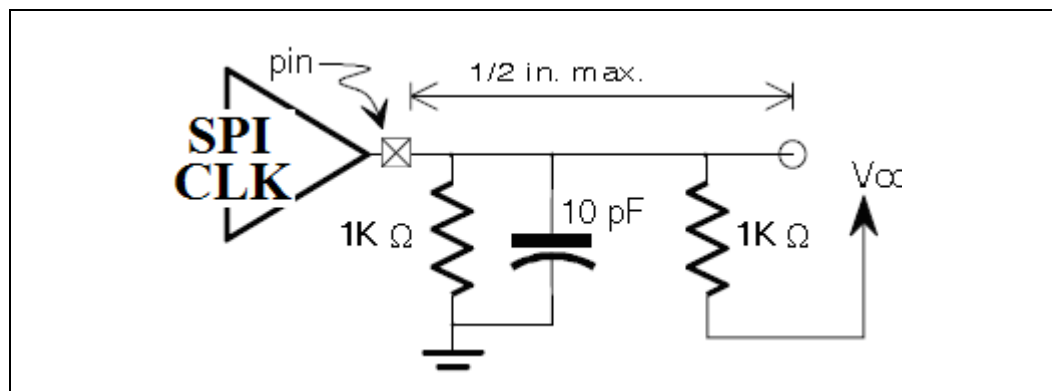
### 3.3 SPI Flash DC Electrical Compatibility Guidelines

Parameter	Min	Max	Units	Notes
Supply Voltage (Vcc)	3.14	3.7	V	
Input High Voltage	$0.5 \cdot V_{CC}$	$V_{CC} + 0.5$	V	
Input Low Voltage	-0.5	$0.3 \cdot V_{CC}$	V	
Output High Characteristics	$0.9 \cdot V_{CC}$	$V_{CC}$	V	$I_{oh} = -0.5\text{mA}$
Output Low Characteristics		$0.1 \cdot V_{CC}$		$I_{ol} = 1.5\text{mA}$
Input Leakage Current	-10	10	$\mu\text{A}$	
Output Rise Slew Rate ( $0.2V_{CC} - 0.6V_{CC}$ )	1	4	V/ns	1
Output Fall Slew Rate ( $0.6V_{CC} - 0.2V_{CC}$ )	1	4	V/ns	1

**Note:**

1. Testing condition: 1K pull up to Vcc, 1kohm pull down and 10pF pull down and 1/2 inch trace See Figure 3.3 for more detail.

**Figure 3-2. PCH Test Load**



§ §



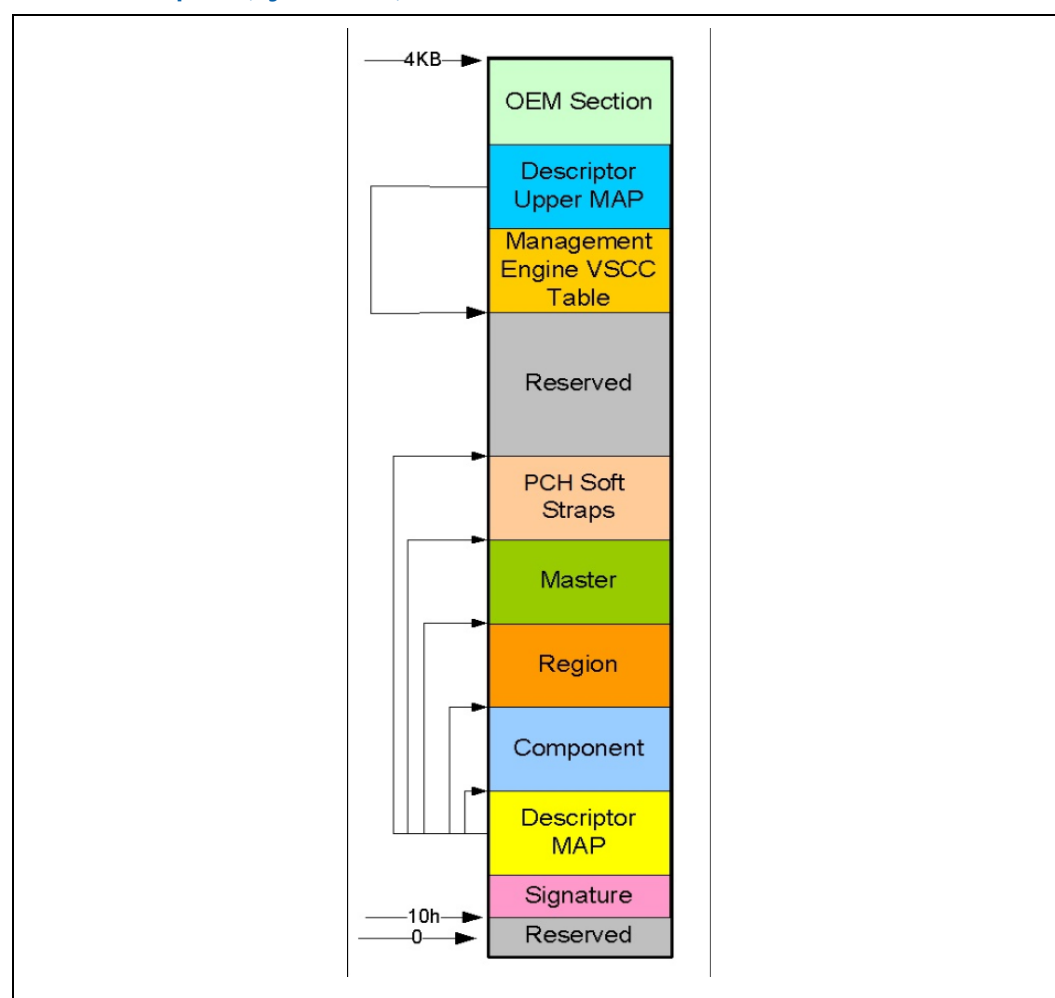
## 4 Descriptor Overview

The Flash Descriptor is a data structure that is programmed on the SPI flash part on Lynx Point-based platforms. The Descriptor data structure describes the layout of the flash as well as defining configuration parameters for the PCH. The descriptor is on the SPI flash itself and is not in memory mapped space like PCH programming registers. The maximum size of the Flash Descriptor is 4 KBytes. It requires its own discrete erase block, so it may need greater than 4 KBytes of flash space depending on the flash architecture that is on the target system.

The information stored in the Flash Descriptor can only be written during the manufacturing process as its read/write permissions must be set to Read Only when the computer leaves the manufacturing floor.

The Descriptor has nine basic parts:

**Figure 4-1. Flash Descriptor (Lynx Point)**





- The Flash signature at the bottom of the flash (offset 10h) must be 0FF0A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the lower five descriptor sections as well as the size of each.
- The Component section has information about the SPI flash part(s) the system. It includes the number of components, density of each component, read, write and erase frequencies and invalid instructions.
- The Flash signature at the bottom of the flash (offset 10h) must be 0FF0A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the lower five descriptor sections as well as the size of each.
- The Component section has information about the SPI flash part(s) the system. It includes the number of components, density of each component, read, write and erase frequencies and invalid instructions.
- The Region section defines the base and the limit of the BIOS, ME and GbE regions as well as their size.
- The master region contains the hardware security settings for the flash, granting read/write permissions for each region and identifying each master.
- PCH chipset soft strap sections contain PCH configurable parameters.
- The Reserved region is for future chipset usage.
- The Descriptor Upper Map determines the length and base address of the Intel ME VSCC Table.
- The Intel ME VSCC Table holds the JEDEC ID and the ME VSCC information for all the SPI Flash part(s) supported by the NVM image. BIOS and GbE write and erase capabilities depend on VSCC0 and VSCC1 registers in SPIBAR memory space.
- OEM Section is 256 Byte section reserved at the top of the Flash Descriptor for use by the OEM.

See [SPI Supported Feature Overview](#) and [Flash Descriptor Records](#) in the *Lynx Point Family External Design Specification (EDS)*.



## 4.1 Flash Descriptor Content

The following sections describe the data structure of the Flash Descriptor on the SPI device. These are not registers or memory space within PCH. FDBAR - is address 0x0 on the SPI flash device on chip select 0.

### 4.1.1 Descriptor Signature and Map

#### 4.1.1.1 FLVALSIG - Flash Valid Signature (Flash Descriptor Records)

Memory Address: FDBAR + 010h

Size: 32 bits

Recommended Value: 0FF0A55Ah

Bits	Description
31:0	<b>Flash Valid Signature.</b> This field identifies the Flash Descriptor sector as valid. If the contents at this location contain 0FF0A55Ah, then the Flash Descriptor is considered valid and it will operate in Descriptor Mode, else it will operate in Non-Descriptor Mode.

#### 4.1.1.2 FLMAPO - Flash Map 0 Register (Flash Descriptor Records)

Memory Address: FDBAR + 014h

Size: 32 bits

Bits	Description
31:27	Reserved
26:24	<b>Number Of Regions (NR).</b> This field identifies the total number of Flash Regions. This number is 0's based, so a setting of all 0's indicates that the only Flash region is region 0, the Flash Descriptor region.
23:16	<b>Flash Region Base Address (FRBA).</b> This identifies address bits [11:4] for the Region portion of the Flash Descriptor. Bits [24:12] and bits [3:0] are 0.  Set this value to 04h. This will define FRBA as 40h.
15:10	Reserved
9:8	<b>Number Of Components (NC).</b> This field identifies the total number of Flash Components. Each supported Flash Component requires a separate chip select. 00 = 1 Component 01 = 2 Components All other settings = Reserved
7:0	<b>Flash Component Base Address (FCBA).</b> This identifies address bits [11:4] for the Component portion of the Flash Descriptor. Bits [24:12] and bits [3:0] are 0.  set this field to 03h. This will define FCBA as 30h



#### 4.1.1.3 FLMAP1—Flash Map 1 Register (Flash Descriptor Records)

Memory Address: FDBAR + 018h

Size: 32 bits

Recommended Value: 15100206h

Bits	Description
31:24	<b>PCH Strap Length (PSL)</b> . Identifies the 1s based number of Dwords of PCH Straps to be read, up to 255 DWs (1KB) max. A setting of all 0's indicates there are no PCH DW straps.  This field <b>MUST</b> be set to 15h
23:16	<b>Flash PCH Strap Base Address (FPSBA)</b> . This identifies address bits [11:4] for the PCH Strap portion of the Flash Descriptor. Bits [24:12] and bits [3:0] are 0.  Set this field to 10h. This will define FPSBA to 100h
15:10	Reserved
9:8	<b>Number Of Masters (NM)</b> . This field identifies the total number of Flash Masters.  Set this field to 10b
7:0	<b>Flash Master Base Address (FMBA)</b> . This identifies address bits [11:4] for the Master portion of the Flash Descriptor. Bits [24:12] and bits [3:0] are 0.  Set this field to 06h. This will define FMBA as 60h

#### 4.1.1.4 FLMAP2—Flash Map 2 Register (Flash Descriptor Records)

Memory Address: FDBAR + 01Ch

Size: 32 bits

Bits	Description
31:24	<b>Register Init Length (RIL)</b> : Identifies the 1's based number of register initialization entries. If this field is set to 0, then there are no Register Init entries to send. Each register init entry is 2DW in length. Note: Refer to Appendix B for chipset register init table details.  Set this field to 0Eh.
23:16	Reserved. Set this field to 21h.
15:08	<b>CPU Strap Length (CPUSL)</b> . Identifies the 1's based number of Dwords of Processor Straps to be read, up to 255 DWs (1KB) max. A setting of all 0's indicates there are no Processor DW straps.  Set this field to 01h.
7:0	<b>Flash CPU Strap Base Address (FCPUSBA)</b> . This identifies address bits [11:4] for the Processor Strap portion of the Flash Descriptor. Bits [24:12] and bits [3:0] are 0.  Set this field to 20h. This will define FCPUSBA as 200h





## 4.1.2 Flash Descriptor Component Section

### 4.1.2.1 The following section of the Flash Descriptor is used to identify the different SPI Flash Components and their capabilities. **FLCOMP—Flash Components Record (Flash Descriptor Records)**

Memory Address: FCBA + 000h

Size: 32 bits

Bits	Description
31	Reserved
30	<b>Dual Output Fast Read Support</b> 0 : Dual Output Fast Read is not supported 1 : Dual Output Fast Read is supported  <b>Notes:</b> <ol style="list-style-type: none"> <li>If the Dual Output Fast Read Support bit is set to 1b, the Dual Output Fast Read instruction is issued in all cases where the Fast Read would have been issue</li> <li>The Frequencies supported for the Dual Output Fast Read are the same as those supported by the Fast Read Instruction</li> <li>If more than one Flash component exists, this field can only be set to "1" if both component support Dual Output Fast Read</li> <li>The Dual output Fast Read is only supported using the 3Bh opcode and dual read only affect the read data, not the address phase.</li> <li>this field only has effect if the SFDP parameter table is not detected. If the SFDP parameter table is detected, this field is ignored and SFDP discovered parameter is used instead</li> </ol>
29:27	<b>Read ID and Read Status Clock Frequency.</b> 000 = 20 MHz 001 = 33 MHz 100 = 50 MHz All other Settings = Reserved  <b>Notes:</b> <ol style="list-style-type: none"> <li>If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components.</li> <li>If setting to 50 MHz, ensure flash meets timing requirements defined in <a href="#">Table 3-3</a></li> </ol>
26:24	<b>Write and Erase Clock Frequency.</b> 000 = 20 MHz 001 = 33 MHz 100 = 50 MHz All other Settings = Reserved  <b>Notes:</b> <ol style="list-style-type: none"> <li>If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components.</li> <li>If setting to 50 MHz, ensure flash meets timing requirements defined in <a href="#">Table 3-3</a></li> </ol>
23:21	<b>Fast Read Clock Frequency.</b> This field identifies the frequency that can be used with the Fast Read instruction. This field is undefined if the Fast Read Support field is '0'. 000 = 20 MHz 001 = 33 MHz 100 = 50 MHz All other Settings = Reserved  <b>Notes:</b> <ol style="list-style-type: none"> <li>If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components.</li> <li>If setting to 50 MHz, ensure flash meets timing requirements defined in <a href="#">Table 3-3</a></li> </ol>



Bits	Description
20	<p><b>Fast Read Support.</b>            0 = Fast Read is not Supported            1 = Fast Read is supported</p> <p>If the Fast Read Support bit is a '1' and a device issues a Direct Read or issues a read command from the Hardware Sequencer and the length is greater than 4 bytes, then the SPI Flash instruction should be "Fast Read". If the Fast Read Support is a '0' or the length is 1-4 bytes, then the SPI Flash instruction should be "Read".</p> <p>Reads to the Flash Descriptor always use the Read command independent of the setting of this bit.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. If more than one Flash component exists, this field can only be set to '1' if both components support Fast Read.</li> <li>2. It is strongly recommended to set this bit to 1b</li> </ol>
19:17	<p><b>Read Clock Frequency.</b>            000 = 20 MHz            All other Settings = Reserved</p> <p><b>Note:</b></p> <ol style="list-style-type: none"> <li>1. If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components.</li> </ol>
16:8	Reserved
7:4	<p><b>Component 1 Density. (C1DEN)</b> This field identifies the size of the 2nd Flash component connected directly to the PCH. If there is not 2nd Flash component, the contents of this field should be read as "1111b"</p> <p>0000 = 512 KB            0001 = 1 MB            0010 = 2 MB            0011 = 4 MB            0100 = 8 MB            0101 = 16 MB            0110 = 32 MB            0111 = 64 MB            1000 - 1110 = Reserved            1111 = 2nd flash component not present</p> <p><b>Note:</b> This field is defaulted to "1111b" after reset  <b>Note:</b> C1DEN field will be <b>ignored</b> if FLMAPO.NC bit [9:8] is set to 00 i.e. 1 component only.</p>
3:0	<p><b>Component 0 Density (CODEN).</b> This field identifies the size of the 1st or only Flash component connected directly to the PCH.</p> <p>0000 = 512 KB            0001 = 1 MB            0010 = 2 MB            0011 = 4 MB            0100 = 8 MB            0101 = 16 MB            0110 = 32 MB            0111 = 64 MB            1000 - 1111 = Reserved</p> <p><b>Note:</b> This field is defaulted to "0101b" (16MB_) after reset.</p>



#### 4.1.2.2 FLILL—Flash Invalid Instructions Record (Flash Descriptor Records)

Memory Address: FCBA + 004h

Size: 32 bits

Bits	Description
31:24	<b>Invalid Instruction 3.</b> See definition of Invalid Instruction 0
23:16	<b>Invalid Instruction 2.</b> See definition of Invalid Instruction 0
15:8	<b>Invalid Instruction 1.</b> See definition of Invalid Instruction 0
7:0	<b>Invalid Instruction 0.</b> Op-code for an instruction that the Flash Controller should protect against, such as Chip Erase. This byte should be set to 0 if there are no invalid instructions to protect against for this field. Op-codes programmed in the Software Sequencing Opcode Menu Configuration and Prefix-Opcode Configuration are not allowed to use any of the Invalid Instructions listed in this register.

### 4.1.3 Flash Descriptor Region Section

The following section of the Flash Descriptor is used to identify the different Regions of the NVM image on the SPI flash.

Flash Regions:

- If a particular region is not using SPI Flash, the particular region should be disabled by setting the Region Base to all 1's, and the Region Limit to all 0's (base is higher than the limit)
- For each region except FLREG0, the Flash Controller must have a default Region Base of 7FFFh and the Region Limit to 0000h within the Flash Controller in case the Number of Regions specifies that a region is not used.

#### 4.1.3.1 FLREG0—Flash Region 0 (Flash Descriptor) Register (Flash Descriptor Records)

Memory Address: FRBA + 000h

Size: 32 bits

Recommended Value: 00000000h

Bits	Description
31	Reserved
30:16	<b>Region Limit.</b> This specifies bits 26:12 of the ending address for this Region.  <b>Notes:</b> <ol style="list-style-type: none"> <li>1. Set this field to 0b. This defines the ending address of descriptor as being FFFh.</li> <li>2. Region limit address Bits[11:0] are assumed to be FFFh</li> </ol>
15	Reserved
14:0	<b>Region Base.</b> This specifies address bits 26:12 for the Region Base.  <b>Note:</b> Set this field to all 0s. This defines the descriptor address beginning at 0h.



#### 4.1.3.2 FLREG1—Flash Region 1 (BIOS) Register (Flash Descriptor Records)

Memory Address: FRBA + 004h

Size: 32 bits

Bits	Description
31	Reserved
30:16	<b>Region Limit.</b> This specifies bits 26:12 of the ending address for this Region.  <b>Notes:</b> 1. Must be set to 0000h if BIOS region is unused (on Firmware hub) 2. Ensure BIOS region size is a correct reflection of actual BIOS image that will be used in the platform 3. Region limit address Bits[11:0] are assumed to be FFFh
15	Reserved
14:0	<b>Region Base.</b> This specifies address bits 26:12 for the Region Base.  <b>Note:</b> If the BIOS region is not used, the Region Base must be programmed to 7FFFh

#### 4.1.3.3 FLREG2—Flash Region 2 (Intel ME) Register (Flash Descriptor Records)

Memory Address: FRBA + 008h

Size: 32 bits

Bits	Description
31	Reserved
30:16	<b>Region Limit.</b> This specifies bits 26:12 of the ending address for this Region.  <b>Notes:</b> 1. Ensure size is a correct reflection of actual Intel ME firmware size that will be used in the platform 2. Region limit address Bits[11:0] are assumed to be FFFh
15	Reserved
14:0	<b>Region Base.</b> This specifies address bits 26:12 for the Region Base.

#### 4.1.3.4 FLREG3—Flash Region 3 (GbE) Register (Flash Descriptor Records)

Memory Address: FRBA + 00Ch

Size: 32 bits

Bits	Description
31	Reserved
30:16	<b>Region Limit.</b> This specifies bits 26:12 of the ending address for this Region.  <b>Notes:</b> 1. The maximum Region Limit is 128KB above the region base. 2. If the GbE region is not used, the Region Limit must be programmed to 0000h 3. Region limit address Bits[11:0] are assumed to be FFFh
15	Reserved
14:0	<b>Region Base.</b> This specifies address bits 26:12 for the Region Base.  <b>Note:</b> If the GbE region is not used, the Region Base must be programmed to 7FFFh



#### 4.1.3.5 FLREG4—Flash Region 4 (Platform Data) Register (Flash Descriptor Records)

Memory Address: FRBA + 010h

Size: 32 bits

Bits	Description
31	Reserved
30:16	<b>Region Limit.</b> This specifies bits 26:12 of the ending address for this Region. <b>Notes:</b> <ol style="list-style-type: none"> <li>1. If PDR Region is not used, the Region Limit must be programmed to 0000h</li> <li>2. Ensure BIOS region size is a correct reflection of actual BIOS image that will be used in the platform</li> <li>3. Region limit address Bits[11:0] are assumed to be FFFh</li> </ol>
15	Reserved
14:0	<b>Region Base.</b> This specifies address bits 26:12 for the Region Base. <b>Note:</b> If the Platform Data region is not used, the Region Base must be programmed to 7FFFh

**Note:** Flash Region 5 (FRBA + 014h) and Region 6 (FRBA + 018h) is reserved in client platform and should set to 7FFFh

### 4.1.4 Flash Descriptor Master Section

#### 4.1.4.1 FLMSTR1—Flash Master 1 (Host CPU/ BIOS) (Flash Descriptor Records)

Memory Address: FMBA + 000h

Size: 32 bits

Bits	Description
31:24	<b>Master Region Write Access:</b> Each bit [31:24] corresponds to Regions [7:0]. If the bit is set, this master can erase and write that particular region through register accesses.  Bit 23 is a don't care as the primary master always has read/write permission to it's primary region
23:16	<b>Master Region Read Access:</b> Each bit [23:16] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses.  Bit 17 is a don't care as the primary master always read/write permission to it's primary region.
15:0	<b>Requester ID:</b> This is the Requester ID (Bus/Device/Function Number_ of the Host CPU  For the host CPU, this should be set to Bus/Device/Function: 0/0/0

#### 4.1.4.2 FLMSTR2—Flash Master 2 (Intel® ME) (Flash Descriptor Records)

Memory Address: FMBA + 004h

Size: 32 bits

Bits	Description
31:24	<b>Master Region Write Access:</b> Each bit [31:24] corresponds to Regions [7:0]. If the bit is set, this master can erase and write that particular region through register accesses.  Bit 26 is a don't care as the primary master always has read/write permission to it's primary region
23:16	<b>Master Region Read Access:</b> Each bit [23:16] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses.  Bit 18 is a don't care as the primary master always read/write permission to it's primary region.
15:0	<b>Requester ID:</b> This is the Requester ID (Bus/Device/Function Number_ of the ME



#### 4.1.4.3 FLMSTR3—Flash Master 3 (GbE) (Flash Descriptor Records)

Memory Address: FMBA + 008h

Size: 32 bits

Bits	Description
31:24	<b>Master Region Write Access:</b> Each bit [31:24] corresponds to Regions [7:0]. If the bit is set, this master can erase and write that particular region through register accesses.  Bit 27 is a don't care as the primary master always has read/write permission to it's primary region
23:16	<b>Master Region Read Access:</b> Each bit [23:16] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses.  Bit 19 is a don't care as the primary master always read/write permission to it's primary region.
15:0	<b>Requester ID:</b> This is the Requester ID (Bus/Device/Function Number_ of the GbE

#### 4.1.5 PCH Softstraps

See Appendix A for Record descriptions and listings.

#### 4.1.6 Descriptor Upper Map Section

##### 4.1.6.1 FLUMAP1—Flash Upper Map 1 (Flash Descriptor Records)

Memory Address: FDBAR + EFCh

Size: 32 bits

Bits	Default	Description
31:16	0	Reserved
15:8	1	<b>Intel ME VSCC Table Length (VTL).</b> Identifies the 1s based number of DWORDS contained in the VSCC Table. Each SPI component entry in the table is 2 DWORDS long.
7:0	1	<b>Intel ME VSCC Table Base Address (VTBA).</b> This identifies address bits [11:4] for the VSCC Table portion of the Flash Descriptor. Bits [26:12] and bits [3:0] are 0.

#### 4.1.7 Intel® ME Vendor Specific Component Capabilities Table

Entries in this table allow support for a SPI flash part for Intel Management Engine capabilities including Intel® Active Management Technology.

Since Flash Partition Boundary Address (FPBA) has been removed, UVSCC and LVSCC has been replaced with VSCC0 and VSCC1 in Lynx Point. VSCC0 is for SPI component 0 and VSCC1 is for SPI component 1.

If SFDP tables are not found by the SPI controller, then the VSCCn must be written before ME can issue a Write or Erase command using the ME Hardware Sequencing interface. BIOS will still need to set up the proper VSCC registers for BIOS and Integrated Gigabit Ethernet usage if there is no SFDP table found.

Each VSCC table entry is composed of two 32 bit fields: JEDEC IDn and the corresponding VSCCn value.

See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) for information on how to program individual entries.



#### 4.1.7.1 JID0—JEDEC-ID 0 Register (Flash Descriptor Records)

Memory Address: VTBA + 000h

Size: 32 bits

Bits	Description
31:24	Reserved
23:16	<b>SPI Component Device ID 1.</b> This field identifies the second byte of the Device ID of the SPI Flash Component. This is the third byte returned by the Read JEDEC-ID command (opcode 9Fh).
15:8	<b>SPI Component Device ID 0.</b> This field identifies the first byte of the Device ID of the SPI Flash Component. This is the second byte returned by the Read JEDEC-ID command (opcode 9Fh).
7:0	<b>SPI Component Vendor ID.</b> This field identifies the one byte Vendor ID of the SPI Flash Component. This is the first byte returned by the Read JEDEC-ID command (opcode 9Fh).

#### 4.1.7.2 VSCC0—Vendor Specific Component Capabilities 0 (Flash Descriptor Records)

Memory Address: VTBA + 004h

Size: 32 bits

**Note:**

VSCC0 applies to SPI flash that connected to CS0.

Bits	Description
31:16	Reserved
15:8	<b>Erase Opcode (EO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in BES.
7:5	<b>Quad Enable Requirements (QER)</b> 000 = Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability (e.g. Micron, Numonyx). 001 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes. If the status register is unlocked and SFDP WSR or VSCC WSR is 1 then SPI controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase. (e.g. Winbond, AMIC, Spansion). 010 = Part requires bit 6 of status register 1 to be set to enable quad IO. If the status register is unlocked and SFDP WSR bit or VSCC WSR is 1 then flash controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase (e.g. Macronix). 011 = Part requires bit 7 of the configuration register to be set to enable Quad (e.g. Atmel). 100 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte (SST/Microchip, Winbond). <b>Note:</b> Please refer to Table note#6 below for details.
4	<b>Write Enable on Write Status (WEWS)</b> 0 = 50h is the opcode used to unlock the status register on SPI flash if <b>WSR</b> (bit 3) is set to 1b. 1 = 06h is the opcode used to unlock the status register on SPI flash if <b>WSR</b> (bit 3) is set to 1b. <b>Note:</b> Please refer to Table Note #4 below for a description how this bit is used.
3	<b>Write Status Required (WSR)</b> 0 = No automatic write of 00h will be made to the SPI flash's status register) 1 = A write of 00h to the SPI flash's status register will be sent on EVERY write and erase performed by Intel ME to the SPI flash. <b>Note:</b> Please refer to Table Note #5 below for a description how this bit is used.
2	<b>Write Granularity (WG)</b> 0 = 1 Byte 1 = 64 Bytes



Bits	Description
1:0	<b>Block/Sector Erase Size (BES)</b> . This field identifies the erasable sector size for all Flash components. 00 = 256 Bytes 01 = 4 K Bytes 10 = 8 K Bytes 11 = 64K Bytes

**Notes:**

1. Bit 3 (**WEWS**) and/or bit 4 (**WSR**) should not be set to '1' if there are non volatile bits in the SPI flash's status register. This may lead to premature flash wear out.
2. This is not an atomic (uninterrupted) sequence. The PCH will not wait for the status write to complete before issuing the next command, potentially causing SPI flash instructions to be disregarded by the SPI flash part. If the SPI flash component's status register is non-volatile, then BIOS should issue an atomic software sequence cycle to unlock the flash part.
3. If both bits 3 (**WSR**) and 4 (**WEWS**) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Intel Management Engine firmware performs.
4. If bit 3 (**WSR**) is set to 1b and bit 4 (**WEWS**) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Intel Management Engine firmware performs.
5. If bit 3 (**WSR**) is set to 0b and bit 4 (**WEWS**) is set to 0b or 1b then sequence of 60h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.
6. The manufacturers information included in the QER list are for guidance purpose. Some manufacturer devices operate as shown in the table above. Check manufacturer's datasheet for exact requirements.

#### 4.1.7.3 JIDn—JEDEC-ID Register n (Flash Descriptor Records)

Memory Address: VTBA + (n\*8)h

Size: 32 bits

**Note:**

"n" is an integer denoting the index of the Intel ME VSCC table.

Bits	Description
31:24	Reserved
23:16	<b>SPI Component Device ID 1</b> . This field identifies the second byte of the Device ID of the SPI Flash Component. This is the third byte returned by the Read JEDEC-ID command (opcode 9Fh).
15:8	<b>SPI Component Device ID 0</b> . This field identifies the first byte of the Device ID of the SPI Flash Component. This is the second byte returned by the Read JEDEC-ID command (opcode 9Fh).
7:0	<b>SPI Component Vendor ID</b> . This field identifies the one byte Vendor ID of the SPI Flash Component. This is the first byte returned by the Read JEDEC-ID command (opcode 9Fh).





#### 4.1.7.4 VSCCn—Vendor Specific Component Capabilities n (Flash Descriptor Records)

Memory Address: VTBA + 004h + (n\*8)h      Size: 32 bits

**Note:** “n” is an integer denoting the index of the Intel ME VSCC table.

Bits	Description
31:16	Reserved
15:8	<b>Erase Opcode (EO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in BES.
7:5	<b>Quad Enable Requirements (QER)</b> 000 = Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability (e.g. Micron, Numonyx). 001 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes. If the status register is unlocked and SFDP bits WSR or VSCC WSR is 1 then SPI controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase. (e.g. Winbond, AMIC, Spansion). 010 = Part requires bit 6 of status register 1 to be set to enable quad IO. If the status register is unlocked and SFDP WSR bit or VSCC WSR is 1 then flash controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase (e.g. Macronix). 011 = Part requires bit 7 of the configuration register to be set to enable Quad (e.g. Atmel). 100 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte (SST/Microchip, Winbond).  <b>Note:</b> Please refer to Table note#6 below for details.
4	<b>Write Enable on Write Status (WEWS)</b> 0 = 50h is the opcode used to unlock the status register on SPI flash if <b>WSR</b> (bit 3) is set to 1b. 1 = 06h is the opcode used to unlock the status register on SPI flash if <b>WSR</b> (bit 3) is set to 1b. <b>Note:</b> Please refer to Table Note #4 below for a description how this bit is used.
3	<b>Write Status Required (WSR)</b> 0 = No automatic write of 00h will be made to the SPI flash's status register) 1 = A write of 00h to the SPI flash's status register will be sent on EVERY write and erase performed by Intel ME to the SPI flash. <b>Note:</b> Please refer to Table Note #5 below for a description how this bit is used.
2	<b>Write Granularity (WG).</b> 0 = 1 Byte 1 = 64 Bytes
1:0	<b>Block/Sector Erase Size (BES).</b> This field identifies the erasable sector size for all Flash components. 00 = 256 Bytes 01 = 4 K Bytes 10 = 8 K Bytes 11 = 64K Bytes

**Notes:**

- Bit 3 (**WEWS**) and/or bit 4 (**WSR**) should not be set to '1' if there are non volatile bits in the SPI flash's status register. This may lead to premature flash wear out.
- This is not an atomic (uninterrupted) sequence. The PCH will not wait for the status write to complete before issuing the next command, potentially causing SPI flash instructions to be disregarded by the SPI flash part. If the SPI flash component's status register is non-volatile, then BIOS should issue an atomic software sequence cycle to unlock the flash part.
- If both bits 3 (**WSR**) and 4 (**WEWS**) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Intel Management Engine firmware performs.
- If bit 3 (**WSR**) is set to 1b and bit 4 (**WEWS**) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Intel Management Engine firmware performs.
- If bit 3 (**WSR**) is set to 0b and bit 4 (**WEWS**) is set to 0b or 1b then sequence of 60h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.
- The manufacturers information included in the QER list are for guidance purpose. Some manufacturer devices operate as shown in the table above. Check manufacturer's datasheet for exact requirements.



## 4.2 OEM Section

Memory Address: F00h

Size: 256 Bytes

256 Bytes are reserved at the top of the Flash Descriptor for use by the OEM. The information stored by the OEM can only be written during the manufacturing process as the Flash Descriptor read/write permissions must be set to Read Only when the computer leaves the manufacturing floor. The PCH Flash controller does not read this information. FFh is suggested to reduce programming time.

## 4.3 Region Access Control

Regions of the flash can be defined from read or write access by setting a protection parameter in the Master section of the Descriptor. There are only three masters that have the ability to access other regions: CPU/BIOS, Intel ME Firmware, and GbE software/driver running on CPU.

Table 4-1. Region Access Control Table Options

Master Read/Write Access			
Region (#)	CPU and BIOS	ME/MCH	GbE Controller
Descriptor (0)	Read / Write	Read / Write	Read / Write
BIOS (1)	CPU and BIOS can always read from and write to BIOS region	Read / Write	Read / Write
ME (2)	Read / Write	ME can always read from and write to ME region	Read / Write
GbE (3)	Read / Write	Read / Write	GbE software can always read from and write to GbE region
PDR (4)	Read / Write	Read / Write	Read / Write

**Notes:**

1. Descriptor and PDR regions are not masters, so they will not have Master R/W access.
2. Descriptor should NOT have write access by any master in production systems.
3. PDR region should only have read and/or write access by CPU/Host. GbE and ME should NOT have access to PDR region.



### 4.3.1 Intel Recommended Permissions for Region Access

The following Intel recommended read/write permissions are necessary to secure Intel Management Engine and Intel ME Firmware.

**Table 4-2. Recommended Read/Write Settings for Platforms Using Intel® ME Firmware**

Master Access	Descriptor Region Bit 0	ME Region Bit2	GbE Region Bit3	BIOS Region Bit1	PDR Region Bit4
ME read access	Y	Y	Y	N	N
ME write access	N	Y	Y	N	N
GbE read access	N	N	Y	N	N
GbE write access	N	N	Y	N	N
BIOS read access	Y	N	Y	Y	‡
BIOS write access	N	N	Y	Y	‡

**Note:**

- ‡ = Host access to PDR is the discretion of the customer. Implementation of PDR is optional.

The table below shows the values to be inserted into the Flash image tool. The values below will provide the access levels described in the table above.

**Table 4-3. Recommended Read/Write Settings for Platforms Using Intel® ME Firmware (Cont'd)**

	ME	GbE	BIOS
Read	0b 0000 1101 = 0x0d	0b 0000 1000 = 0x08	0b 000‡ 1011 = 0x‡B
Write	0b 0000 1100 = 0x0c	0b 0000 1000 = 0x08	0b 000‡ 1010 = 0x‡A

**Note:**

- ‡ = Value dependent on if PDR is implemented and if Host access is desired.

### 4.3.2 Overriding Region Access

Once access Intel recommended Flash settings have been put into the flash descriptor, it may be necessary to update the ME region with a Host program or write a new Flash descriptor.

Assert HDA\_SDO HIGH during the rising edge of PWROK to set the Flash descriptor override strap.

This strap should only be visible and available in manufacturing or during product development.

After this strap has been set you can use a host based flash programming tool like FPT.exe to write/read any area of serial flash that is not protected by Protected Range Registers. Any area of flash protected by Protected range Registers will still NOT be writeable/readable.

See [6.3 SPI Protected Range Register Recommendations](#) for more details.



## 4.4 Intel® Management Engine (Intel® ME) Vendor-Specific Component Capabilities Table

The Intel ME VSCC Table defines how the Intel ME will communicate with the installed SPI flash if there is no SFDP table found. This table is defined in the descriptor and is the responsibility of who puts together the NVM image. VSCCn registers are defined in memory space and must be set by BIOS. This table must define every flash part that is intended to be used. The size (number of max entries) of the table is defined in [4.1.6.1 FLUMAP1—Flash Upper Map 1 \(Flash Descriptor Records\)](#). Each Table entry is made of two parts: the JEDEC ID and VSCC setting.

### 4.4.1 How to Set a JEDEC ID Portion of Intel® ME VSCC Table Entry

[8.3.2 Device ID](#) shows how to obtain the 3 byte JEDEC ID for the target SPI flash.

[7.4.1 Adding a New Table Entry](#) Shows how to set this value in FITC.

**Table 4-4. Jidn - JEDEC ID Portion of Intel® ME VSCC Table**

Bits	Description
31:24	Reserved.
23:16	<b>SPI Component Device ID 1:</b> This identifies the second byte of the Device ID of the SPI Flash Component. This is the third byte returned by the Read JEDEC-ID command (opcode 9Fh).
15:8	<b>SPI Component Device ID 0:</b> This identifies the first byte of the Device ID of the SPI Flash Component. This is the second byte returned by the Read JEDEC-ID command (opcode 9Fh).
7:0	<b>SPI Component Vendor ID:</b> This identifies the one byte Vendor ID of the SPI Flash Component. This is the first byte returned by the Read JEDEC-ID command (opcode 9Fh).

If using Flash Image Tool (FIT) refer to System Tools user guide in the Intel ME FW kit and the respective FW Bring up Guide on how to build the image. If not, refer to [4.1.6.1 FLUMAP1—Flash Upper Map 1 \(Flash Descriptor Records\)](#) thru [4.1.7.4 VSCCn—Vendor Specific Component Capabilities n \(Flash Descriptor Records\)](#).



## 4.4.2 How to Set a VSCC Entry in Intel® ME VSCC Table for Lynx Point Family Platforms

VSCC0 needs to be programmed in instances where there is only SPI component in the system. When using an asymmetric flash component (part with two different sets of attributes based on address) VSCC0 and VSCC1 will need to be used. This includes if the system is intended to support both symmetric AND asymmetric SPI flash parts.

Refer to [4.4.3 Intel® ME VSCC Table Settings for Lynx Point Family Systems](#).

See text below the table for explanation on how to determine Management Engine VSCC value.

**Table 4-5. Vscn – Vendor-Specific Component Capabilities Portion of the Lynx Point Family Platforms**

Bits	Description
31:16	Reserved
15:8	<b>Erase Opcode (EO)</b> . This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in BES.
7:5	<b>Quad Enable Requirements (QER)</b> 000 = Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability (e.g. Micron, Numonyx). 001 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes. If the status register is unlocked and SFDP WSR bit or VSCC WSR is 1 then SPI controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase. (e.g. Winbond, AMIC, Spansion). 010 = Part requires bit 6 of status register 1 to be set to enable quad IO. If the status register is unlocked and SFDP WSR bit or VSCC WSR is 1 then flash controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase (e.g. Macronix). 011 = Part requires bit 7 of the configuration register to be set to enable Quad (e.g. Atmel). 100 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte (SST/Microchip, Winbond).  <b>Note:</b> Please refer to Table note#6 below for details.
4	<b>Write Enable on Write Status (WEWS)</b> 0 = 50h is the opcode used to unlock the status register on SPI flash if <b>WSR</b> (bit 3) is set to 1b. 1 = 06h is the opcode used to unlock the status register on SPI flash if <b>WSR</b> (bit 3) is set to 1b. <b>Note:</b> Please refer to Table Note #4 below for a description how this bit is used.
3	<b>Write Status Required (WSR)</b> 0 = No automatic write of 00h will be made to the SPI flash's status register) 1 = A write of 00h to the SPI flash's status register will be sent on EVERY write and erase performed by Intel ME to the SPI flash. <b>Note:</b> Please refer to Table Note #5 below for a description how this bit is used.
2	<b>Write Granularity (WG)</b> . 0 = 1 Byte 1 = 64 Bytes
1:0	<b>Block/Sector Erase Size (BES)</b> . This field identifies the erasable sector size for all Flash components. 00 = 256 Bytes 01 = 4 K Bytes 10 = 8 K Bytes 11 = 64K Bytes

**Notes:**

1. Bit 3 (**WEWS**) and/or bit 4 (**WSR**) should not be set to '1' if there are non volatile bits in the SPI flash's status register. This may lead to premature flash wear out.
2. This is not an atomic (uninterrupted) sequence. The PCH will not wait for the status write to complete before issuing the next command, potentially causing SPI flash instructions to be disregarded by the SPI flash part. If the SPI flash component's status register is non-volatile, then BIOS should issue an atomic software sequence cycle to unlock the flash part.
3. If both bits 3 (**WSR**) and 4 (**WEWS**) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Intel Management Engine firmware performs.
4. If bit 3 (**WSR**) is set to 1b and bit 4 (**WEWS**) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Intel Management Engine firmware performs.
5. If bit 3 (**WSR**) is set to 0b and bit 4 (**WEWS**) is set to 0b or 1b then sequence of 60h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.
6. The manufacturers information included in the QER list are for guidance purpose. Some manufacturer devices operate as shown in the table above. Check manufacturer's datasheet for exact requirements.

**Erase Opcode (EO)** and **Block/Sector Erase Size (BSES)** should be set based on the flash part and the firmware on the platform. For Intel ME enabled platforms this should be 4 KB.

**Write Status Required (WSR)** or **Write Enable on Write Status (WEWS)** should be set on flash devices that require an opcode to enable a write to the status register. Intel ME Firmware will write a 00h to status register to unlock the flash part for every erase/write operation. If this bit is set on a flash part that has non-volatile bits in the status register then it may lead to pre-mature wear out of the flash.

- Set the **WSR** bit to 1b and **WEWS** to 0b if the Enable Write Status Register opcode (50h) is needed to unlock the status register. Opcodes sequence sent to SPI flash will bit 50h 01h 00h.
- Set the **WSR** bit to 1b AND **WEWS** bit to 1b if write enable (06h) will unlock the status register. Opcodes sequence sent to SPI flash will bit 06h 01h 00h.
- Set the **WSR** bit to 0b AND **WEWS** bit to 0b or 1b, if write enable (06h) will unlock the status register. Opcodes sequence sent to SPI flash will bit 06h
- **WSR or WEWS should be not be set on devices that use non volatile memory for their status register.** Setting this bit will cause operations to be ignored, which may cause undesired operation. Ask target flash vendor if this is the case for the target flash. See [6.1 Unlocking SPI Flash Device Protection for Intel® 8 Series Chipset Family Platforms](#) and [6.2 Locking SPI Flash via Status Register](#) for more information.

**Erase Opcode (EO)** and **Block/Sector Erase Size (BES)** should be set based on the flash part and the firmware on the platform.

**Write Granularity (WG)** bit should be set based on the capabilities of the flash device. If the flash part is capable of writing 1 to 64 bytes (or more) with the 02h command you can set this bit 0 or 1. Setting this bit high will result in faster write performance. If flash part only supports single byte write only, then set this bit to 0.

**Bit ranges 31:16 and 7:5** are reserved and should set to all zeros.

### 4.4.3 Intel® ME VSCC Table Settings for Lynx Point Family Systems

To understand general guidelines for BIOS VSCC settings on different SPI flash devices, please refer to **VSCCommn.bin Content application note** (VSCCommn\_bin Content.pdf under Flash Image Tool directory)





## 5 Serial Flash Discoverable Parameter (SFDP) Overview

### 5.1 Introduction

As the feature set of serial flash progresses, there is an increasing amount of divergence as individual vendors find different solution to adding new functionality such as speed and addressing.

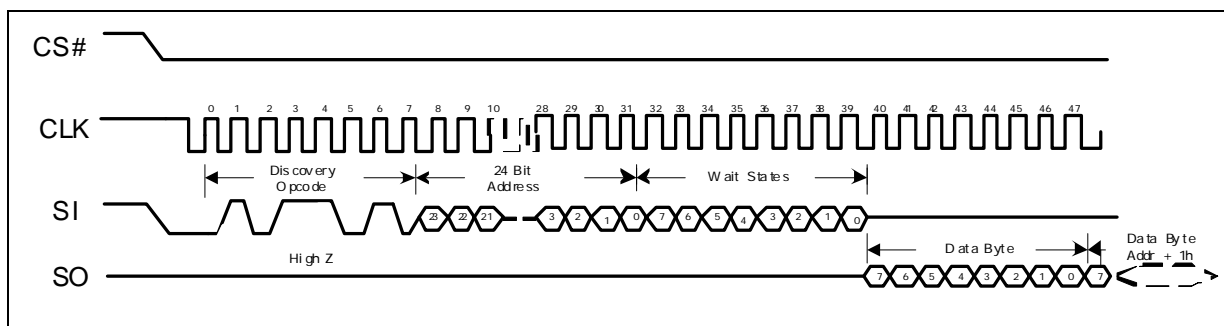
These guidelines are a standard that will allow for individual vendors to have their value add features, but will allow for a controller to discover the attributes needed to operate.

### 5.2 Discoverable Parameter Opcode and Flash Cycle

The discoverable parameter read opcode behaves like a fast read command. The opcode is 5Ah and the address cycle is 24 bit long. after the opcode 5Ah is clocked in, there are 24 bit of address clocked in. There will then be eight clock (8 wait states) before valid data is clocked out. There is flexibility in the number of wait states, but they must be byte aligned (multiple of 8 wait states).

SFDP read must update at a frequency between 17Mhz and 66Mhz with a single byte of wait state.

Figure 5-1. SFDP Read Instruction Sequence





## 5.3 Parameter Table Supported on PCH

The flash controller first checks for a valid SFDP header. The value of the major and minor revision fields in the SFDP header are don't care. If a valid SFDP header is found, the controller supports auto discovery of the Component Property Parameter Table (CPPT).

The following capabilities are only supported on PCH if CPPT is successfully discovered and parameter values indicate that they are supported. These capabilities are not supported as default.

- Quad I/O Read
- Quad Output Read
- Dual I/O read
- Block /Sector Erase size

**Note:** If SFDP is valid and advertises 4kByte erase capability, then BES is taken from the SFDP table, otherwise it is taken from the BIOS VCSS table.

PCH will also read the following opcode from parameter table and store to PCH if SFDP is valid and the following function is supported.

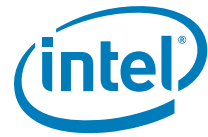
- Erase Opcode
- Dual Output Fast Read Opcode
- Dual I/O Fast Read Opcode
- Quad Output Fast Read Opcode
- Quad I/O Fast Read Opcode

## 5.4 Detail JEDEC Specification

Please refer to [www.jedec.com](http://www.jedec.com) JESD216 for detail SFDP specification on SPI.

§ §





## 6 Configuring BIOS/GbE for SPI Flash Access

---

### 6.1 Unlocking SPI Flash Device Protection for Intel® 8 Series Chipset Family Platforms

BIOS must account for any built in protection from the flash device itself. BIOS must ensure that any flash based protection will only apply to BIOS region only. It should not affect the ME or GbE regions.

All the SPI flash devices that meet the SPI flash requirements in the *Intel® 8 Series Chipset Family External Design Specification (EDS)* will be unlocked by writing a 00h to the SPI flash's status register. This command must be done via an atomic software sequencing to account for differences in flash architecture. Atomic cycles are uninterrupted in that it does not allow other commands to execute until a read status command returns a 'not busy' result from the flash.

Some flash vendors implement their status registers in NVM flash (non-volatile memory). This takes much more time than a write to volatile memory. During this write, the flash part will ignore all commands but a read to the status register (opcode 05h). The output of the read status register command will tell the PCH when the transaction is done.

Recommended flash unlocking sequence:

- Write enable (06h) command will have to be in the prefix opcode configuration register.
- The "write to status register" opcode (01h) will need to be an opcode menu configuration option.
- Opcode type for write to status register will be '01': a write cycle type with no address needed.
- The FDATA0 register should to be programmed to 0000 0000h.
- Data Byte Count (DBC) in Software Sequencing Flash Control register should be 000000b. Errors may occur if any non zero value is here.
- Set the Cycle Opcode Pointer (COP) to the "write to status register" opcode.
- Set to Sequence Prefix Opcode Pointer (SPOP) to Write Enable.
- Set the Data Cycle (DS) to 1.
- Set the Atomic Cycle Sequence (ACS) bit to 1.
- To execute sequence, set the SPI Cycle Go bit to 1.

Please see the [Serial Peripheral Interface Memory Mapped Configuration Registers](#) in the *Intel® 8 Series Chipset Family External Design Specification (EDS)* for more detailed information.



## 6.2 Locking SPI Flash via Status Register

Flash vendors that implement their status register with non-volatile memory can be updated a limited number of times. This means that this register may wear out before the desired endurance for the rest of the flash. It is highly recommended that BIOS vendors and customers do NOT use the SPI flash's status register to protect the flash in multiple master systems.

BIOS should try to minimize the number of times that the system is locked and unlocked.

Care should be taken when using status register based SPI flash protection in multiple master systems such as Management Engine firmware and/or integrated GbE. BIOS must ensure that any flash based protection will only apply to BIOS region only. It should affect not the ME or GbE regions.

Please contact your desired flash vendor to see if their status register protection bits are volatile or non-volatile. Flash parts implemented with volatile systems do not have this concern.

## 6.3 SPI Protected Range Register Recommendations

The PCH has a mechanism to set up to 5 address ranges from HOST access. These are defined in PR0, PR1, PR2, PR3 and PR4 in the PCH EDS. These address ranges are NOT unlocked by assertion of Flash descriptor Override.

It is strongly recommended to use a protected range register to lock down the factory default portion of Intel ME Ignition FW region. The runtime portion should be left unprotected as to allow BIOS to update it.

It is strongly recommended that if Flash Descriptor Override strap (which can be checked by reading **FDOPSS (0b Flash Descriptor override is set, 1b not set) in PCH memory space (SPIBAR+4h bit 13)**) is set, do not set a Protected range to cover the Intel ME Ignition FW factory defaults. This would allow a flashing of a complete image when the Flash descriptor Override strap is set.



## 6.4 Software Sequencing Opcode Recommendations

It is strongly recommended that the “9Fh” JEDEC ID be used instead of “90h” or “AB”. The JEDEC ID Council ensures that every SPI flash model is unique. There are flash vendors that have flash parts of different sizes that report out the same value using the “90h” opcode.

Intel utilities such as the Flash Programming Tool will incorrectly detect the flash part in the system and it may lead to undesired program operation.

The Flash Programming Tool requires the following software sequencing opcodes to be programmed in the OPMENU and corresponding OPTYPE register.

It is strongly recommended that you do not program opcodes write enable commands into the OPMENU definition. These should be programmed in the PREOP register.

Order of the opcodes is not important, but the OPMENU and OPTYPE do have to correspond. see [OPTYPE— Opcode Type Configuration Register OPMENU-Opcode Menu Configuration Register](#) in the *Intel® 8 Series Chipset Family External Design Specification (EDS)*.

**Table 6-1. Recommended Opcodes for FPT Operation**

Function	OPMENU	OPTYPE
Write to Status Register	0x01	'01'
Program Data	0x02	'11'
Read Data	0x03	'10'
Read Status Register	0x05	'00'
4 KB Erase	0x20	'11'
JEDEC ID	0x9F	'00'

**Table 6-2. Recommended Opcodes for FPT Operation**

Function	PREOP
Write Enable	0x06
Enable Status Register Write	0x50



## 6.5 Recommendations for Flash Configuration Lockdown and Vendor Component Lock Bits

### 6.5.1 Flash Configuration Lockdown

It is strongly recommended that BIOS sets the Host and GbE **Flash Configuration Lock-Down (FLOCKDN)** bits (located at SPIBAR + 04h and MBAR +04h respectively) to '1' on production platforms. If these bits are not set, it is possible to make register changes that can cause undesired host, integrated GbE and Intel ME functionality as well as lead to unauthorized flash region access.

Refer to [HSFS— Hardware Sequencing Flash Status Register in the Serial Peripheral Interface Memory Mapped Configuration Registers](#) section and [HSFS— Hardware Sequencing Flash Status Register in the GbE SPI Flash Programing Registers](#) section in the *Intel® 8 Series Chipset Family External Design Specification (EDS)*.

### 6.5.2 Vendor Component Lock

It is strongly recommended that BIOS sets the **Vendor Component Lock (VCL)** bits. These bits are located in the BIOS/GbE VSCC0 registers. VCL applies the lock to both VSCC0 and VSCC1 even if VSCC1 is not used. Without the VCL bits set, it is possible to make Host/GbE VSCC register(s) changes in that can cause undesired host and integrated GbE SPI flash functionality.

Refer to [VSCC— Vendor Specific Component Capabilities Register](#) in the *Intel® 8 Series Chipset Family External Design Specification (EDS)* for more information.

## 6.6 Host Vendor Specific Component Control Registers (VSCC) for Intel® 8 Series Chipset Family Systems

VSCC are memory mapped registers are used by the PCH when BIOS or Integrate LAN reads, programs or erases the SPI flash via Hardware sequencing.

Flash Partition Boundary Address (FBPBA) has been removed and UVSCC and LVSCC has been replaced with VSCC0 and VSCC1 in Intel® 8 Series Chipset. VSCC0 is for SPI component 0 and VSCC1 is for SPI component 1. SPI controller will determine which VSCC (VSCC0 or VSCC1) to be used by comparing Flash Linear Address (FLA) with size of SPI component 0 (CODEN). When FLA <= CODEN then VSCC0 will be used; whereas FLA > CODEN then VSCC1 will be used if one SPI flash component used in the system, VSCC0 needs to be set.

Refer to [VSCC— Lower Vendor Specific Component Capabilities Register](#) and in the *Intel® 8 Series Chipset Family External Design Specification (EDS)*.

See text below the tables for explanation on how to determine VSCC register values.



**Table 6-3. VSCC0 - Vendor-Specific Component Capabilities Register for SPI Component 0 (Sheet 1 of 2)**

Bit	Description
31	<p><b>Component Property Parameter Table Valid (CPPTV) - RO:</b>            This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in SPI Component 0            If CPPTV bit is '0', software must configure the VSCC register appropriately. If CPPTV bit is '1', the corresponding parameter values discovered via SFDP will be used. In most cases, software is not required to configure the VSCC register. However, if the SFDP table indicates an erase size other than 4k byte, then the software is required to program the VSCC.EO register with the correct erase opcode.</p>
30:24	Reserved
23	<p><b>Vendor Component Lock (VCL): — RW/L:</b>            '0': The lock bit is not set            '1': The Vendor Component Lock bit is set.</p> <p>This register locks itself when set.</p> <p>This bit applies to both VSCC0 and VSCC1            All bits locked by (VCL) will remained locked until a global reset.</p>
22:16	Reserved
15:8	<p><b>Erase Opcode (EO)— RW:</b>            This register is programmed with the Flash erase instruction opcode required by the vendor's Flash component. Software must program this register if the SFDP table for this component does not show 4 kByte erase capability</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p><b>Note:</b> If CPPTV is 1 and the SPDP0 table shows 4k erase capability, the SFDP0 erase code is used instead of this register</p>
7:5	<p><b>Quad Enable Requirements (QER)</b>            000 = Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability (e.g. Micron, Numonyx).            001 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes. If the status register is unlocked and SFDP bits WSR or VSCC WSR is 1 then SPI controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase. (e.g. Winbond, AMIC , Spansion).            010 = Part requires bit 6 of status register 1 to be set to enable quad IO. If the status register is unlocked and SFDP WSR bit or VSCC WSR is 1 then flash controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase (e.g. Macronix).            011 = Part requires bit 7 of the configuration register to be set to enable Quad (e.g. Atmel).            100 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte (SST/Microchip, Winbond).</p> <p><b>Note:</b> This register is locked by the Vendor Component Lock (VCL) bit.</p>
4	<p><b>Write Enable on Write Status (WEWS) — RW:</b>            '0' = 50h will be the opcode used to unlock the status register on the SPI flash if <b>WSR</b> (bit 3) is set to 1b.            '1' = 06h will be the opcode used to unlock the status register on the SPI flash if <b>WSR</b> (bit 3) is set to 1b.</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p><b>Note:</b> Please refer to <a href="#">Table 6-5</a> for a description of how these bits is used.</p>

**Table 6-3. VSCC0 - Vendor-Specific Component Capabilities Register for SPI Component 0 (Sheet 2 of 2)**

Bit	Description
3	<p><b>Write Status Required (WSR)</b> — RW:</p> <p>'0' = No automatic write of 00h will be made to the SPI flash's status register.</p> <p>'1' = A write of 00h to the SPI flash's status register will be sent on EVERY write and erase to the SPI flash performed by Host and GbE.</p> <p>This register is locked by the Vendor Component Lock (<b>VCL</b>) bit.</p> <p><b>Note:</b> Please refer to <a href="#">Table 6-5</a> for a description of how these bits is used.</p>
2	<p><b>Write Granularity (WG)</b> — RW:</p> <p>0: 1 Byte</p> <p>1: 64 Byte</p> <p>This register is locked by the Vendor Component Lock (<b>VCL</b>) bit.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"><li>1. If more than one Flash component exists, this field must be set to the lowest common write granularity of the different Flash components</li><li>2. If using 64 B write, BIOS must ensure that multiple byte writes do not occur over 256 B boundaries. This will lead to corruption as the write will wrap around the page boundary on the SPI flash part. This is a feature in page writable SPI flash.</li></ol>
1:0	<p><b>Block/Sector Erase Size (BES)</b>— RW:</p> <p>This field identifies the erasable sector size for Flash components.</p> <p>Valid Bit Settings:</p> <p>00: 256 Byte</p> <p>01: 4 KByte</p> <p>10: 8 KByte</p> <p>11: 64 K</p> <p>This register is locked by the Vendor Component Lock (<b>VCL</b>) bit.</p> <p>Hardware takes no action based on the value of this register. The contents of this register are to be used only by software and can be read in the HSFSTS.BERASE register in both the BIOS and the GbE program registers if FLA is less than FPBA.</p>



**Table 6-4. VSCC1 - Vendor Specific Component Capabilities Register for SPI Component 1 (Sheet 1 of 2)**

Bit	Description
31	<p><b>Component Property Parameter Table Valid (CPPTV) - RO:</b></p> <p>This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in SPI Component 1</p> <p>If CPPTV bit is '0', software must configure the VSCC register appropriately. If CPPTV bit is '1', the corresponding parameter values discovered via SFDP will be used. In most cases, software is not required to configure the VSCC register. However, if the SFDP table indicates an erase size other than 4k byte, then the software is required to program the VSCC.EO register with the correct erase opcode.</p>
31:16	Reserved
15:8	<p><b>Erase Opcode (EO)— RW:</b></p> <p>This register is programmed with the Flash erase instruction opcode required by the vendor's Flash component.</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p>
7:5	<p><b>Quad Enable Requirements (QER)</b></p> <p>000 = Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability (e.g. Micron, Numonyx).</p> <p>001 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes. If the status register is unlocked and SFDP bits WSR or VSCC WSR is 1 then SPI controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase. (e.g. Winbond, AMIC , Spansion).</p> <p>010 = Part requires bit 6 of status register 1 to be set to enable quad IO. If the status register is unlocked and SFDP WSR bit or VSCC WSR is 1 then flash controller cannot use the quad output, quad IO features of this part because the hardware will automatically write one byte of zeros to status register with every write/erase (e.g. Macronix).</p> <p>011 = Part requires bit 7 of the configuration register to be set to enable Quad (e.g. Atmel).</p> <p>100 = Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte (SST/Microchip, Winbond).</p> <p><b>Note:</b> This register is locked by the Vendor Component Lock (VCL) bit.</p>
4	<p><b>Write Enable on Write to Status (WEWS) — RW:</b></p> <p>'0' = 50h will be the opcode used to unlock the status register if <b>WSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register if <b>WSR</b> (bit 3) is set to 1b.</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p>Please refer to <a href="#">Table 6-5</a> for a description of how these bits is used.</p>

**Table 6-4. VSCC1 - Vendor Specific Component Capabilities Register for SPI Component 1 (Sheet 2 of 2)**

Bit	Description
3	<p><b>Write Status Required (WSR)</b> — RW:</p> <p>'0' = No automatic write of 00h will be made to the SPI flash's status register</p> <p>'1' = A write of 00h to the SPI flash's status register will be sent on EVERY write and erase to the SPI flash performed by Host and GbE.</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p><b>Note:</b> Please refer to <a href="#">Table 6-5</a> for a description of how these bits is used.</p>
2	<p><b>Write Granularity (WG)</b> — RW:</p> <p>0: 1 Byte</p> <p>1: 64 Byte</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p>If more than one Flash component exists, this field must be set to the lowest common write granularity of the different Flash components.</p> <p>If using 64 B write, BIOS must ensure that multiple byte writes do not occur over 256 B boundaries. This will lead to corruption as the write will wrap around the page boundary on the SPI flash part. This is a feature in page writeable SPI flash.</p>
1:0	<p><b>Block/Sector Erase Size (BES)</b>— RW: This field identifies the erasable sector size for all Flash components.</p> <p>Valid Bit Settings:</p> <p>00: 256 Byte</p> <p>01: 4 KByte</p> <p>10: 8 KByte</p> <p>11: 64 K</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p>Hardware takes no action based on the value of this register. The contents of this register are to be used only by software and can be read in the HSFSTS.BERASE register in both the BIOS and the GbE program registers if FLA is less than FPBA.</p>

**Erase Opcode (EO)** and **Block/Sector Erase Size (BSES)** should be set based on the flash part and the firmware on the platform.

- Either **Write Status Required (WSR)** or **Write Enable on Write Status (WEWS)** should be set on flash devices that require an opcode to enable a write to the status register. BIOS and GbE will write a 00h to the SPI flash's status register to unlock the flash part for every erase/write operation. If this bit is set on a flash part that has non-volatile bits in the status register then it may lead to pre-mature wear out of the flash and may result in undesired flash operation. Please refer to [Table 6-5](#) for a description of how these bits is set and what is the expected operation from the controller during erase/write operation.





Table 6-5. Description of How WSR and WEWS is Used

WSR	WEWS	Flash Operation
1b	0b	If the Enable Write Status Register opcode (50h) is needed to unlock the status register. Opcodes sequence sent to SPI flash will bit 50h 01h 00h.
1b	1b	If write enable (06h) will unlock the status register. Opcodes sequence sent to SPI flash will bit 06h 01h 00h.
0b	0 or 1b	Sequence of 60h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.

**Note:** **WSR or WEWS should be not be set on devices that use non volatile memory for their status register.** Setting this bit will cause operations to be ignored, which may cause undesired operation. Ask target flash vendor if this is the case for the target flash. See [6.1 Unlocking SPI Flash Device Protection for Intel® 8 Series Chipset Family Platforms](#) and [6.2 Locking SPI Flash via Status Register](#) for more information.

**Write Granularity (WG)** bit should be set based on the capabilities of the flash device. If the flash part is capable of writing 1 to 64 bytes (or more) with the 02h command you can set this bit 0 or 1. Setting this bit high will result in faster write performance. If flash part only supports single byte write only, then set this bit to 0. Setting this bit high requires that BIOS ensure that no multiple byte write operation does not cross a 256 Byte page boundary, as it will have unintended results. This is a feature of page programming capable flash parts.

**Vendor Component Lock (VCL)** should remain unlocked during development, but locked in shipping platforms. When **VCL** and **FLOCKDN** are set, it is possible that you may not be able to use in system programming methodologies including Intel Flash Programming Tool if programmed improperly. It will require a system reset to unlock this register and BIOS not to set this bits. See [6.5 Recommendations for Flash Configuration Lockdown and Vendor Component Lock Bits](#) for more details.

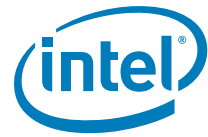
**All reserved bits** should set to zeros.

## 6.7 Host VSCC Register Settings for Intel® 8 Series Chipset Family Systems

To understand general guidelines for VSCC settings with different SPI flash devices, please refer to **VSCCCommn.bin content application note** (VSCCCommn\_bin Content.pdf under Flash Image Tool directory). VSCCCommn.bin contains SPI devices vendor ID, device ID and recommended VSCC values.

§ §





## 7 Flash Image Tool

This is a general overview to the Flash Image Tool (FIT). Please refer to the documentation that comes with the flash tools executables for the correct feature set for the version of the flash tool being used.

The purpose of the Flash Image Tool is to simplify the creation and configuration of the Flash image for the Intel® 8 Series Chipset family platforms. The Flash Image Tool makes a flash image by creating a descriptor and combining the following image files:

- BIOS
- Intel Integrated Gigabit LAN
- Intel ME Firmware
- Platform Data Region

The user is able to manipulate the image layout through a graphical user interface (GUI) and change the various chipset parameters to match the target hardware. Different configurations can be saved to a file so image layouts do not need to be recreated each time.

The user does not need to interact with the GUI each time they need to create an image. The tool supports a set of command line parameters that can be used to build an image from the command prompt or from a makefile. A previously stored configuration can be used to define the image layout, making interacting with the GUI unnecessary.

The Flash Image Tool does not program the flash. The Flash Image tool only generates a binary image file. This image must be burned onto the flash by other means.

### 7.1 Flash Image Details

A flash image is composed of five regions. The locations of these regions are referred to in terms of where it can be found within the total memory of the flash.

**Figure 7-1. Firmware Image Components**

Descriptor	PDR	GbE	ME	BIOS
------------	-----	-----	----	------

- **Descriptor:** Takes up a fixed amount of space at the beginning of flash memory. The descriptor contains information such as space allocated for each region of the flash image, read-write permissions for each region, and a space which can be used for vendor-specific data.
- **ME:** Required region that contains code and configuration data for ME functions such as ME Clock control, Intel AMT, etc.
- **GbE:** Optional region that contains code and configuration data for Intel integrated Gigabit Ethernet and 10/100 Ethernet.
- **Platform Data Region:** Optional region that contains data reserved for BIOS/Host usage.
- **BIOS:** Optional region that contains code and configuration for the entire platform. Region is only optional if BIOS is on Firmware Hub.

### 7.1.1 Flash Space Allocation

FITC allocates SPI flash space allocation for each region as follows:

1. Each region can be assigned a fixed amount of space. If no fixed space is assigned, then the region occupies only as much space as it requires.
2. If after allocation for all regions there is still space left in flash, then the ME region expands to fill the remaining space.
3. If there is leftover space and the ME region is not implemented, then the BIOS region expands to use the remaining space.

If there is leftover space and the BIOS region is not implemented, then the GbE region expands to contain the remaining space.

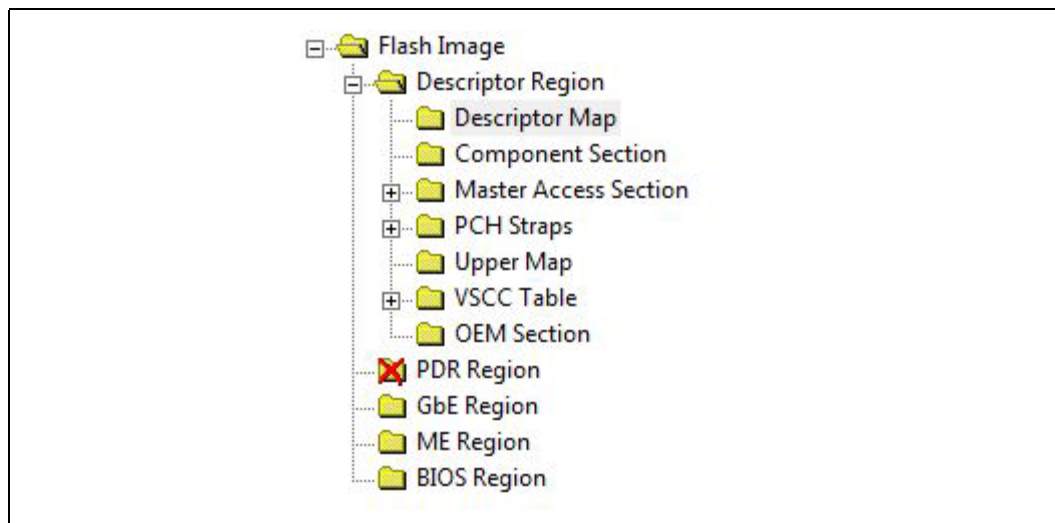
## 7.2 Modifying the Flash Descriptor Region

The flash descriptor region contains information about the flash image and the target hardware. It is important for this region to be configured correctly or else the target system may not function as desired.

### 7.2.1 Setting the Number and Size of the Flash Components

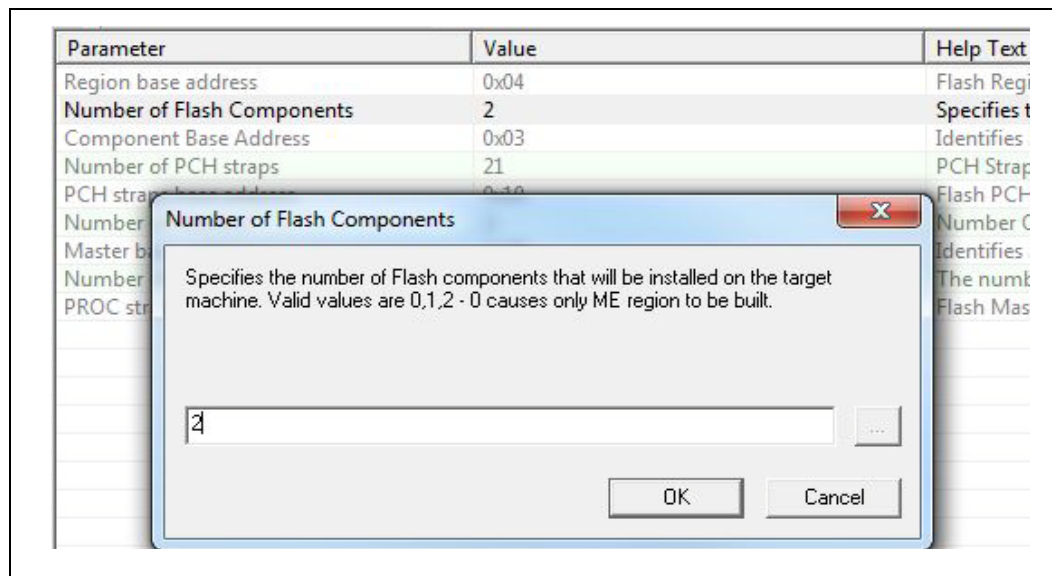
To set the number of flash components, expand the “Descriptor Region” node in the tree on the left side of the main window. Then, select the “Descriptor Map” node (See 3). All of the parameters for the descriptor map section will appear in the list on the right side of the main window.

Figure 7-2. Editable Flash Image Region List



Double-click the list item named “Number of Flash Components” (See [Section 7.3](#)). A dialog will appear allowing the user to enter the number of flash components (valid values are 1 or 2). Click “Ok” to update the parameter.

Figure 7-3. Descriptor Region – Descriptor Map Options



Some SPI flash devices support both standard and fast read opcodes. Fast reads are able to operate at faster frequencies than the regular reads. For PCH to support these faster read commands, fast read support must be set to true. For Intel® 8 Series Chipset, this should be set to 50 MHz for Intel AMT enabled platforms.

Figure 7-4. Descriptor Region – Fast Read Support Options

Parameter	Value	Help Te
Read ID and Read Status clock frequency	50MHz	If more
Write and erase clock frequency	50MHz	If more
Fast read clock frequency	50MHz	This fiel
Fast read support	true	Enables
Read clock frequency	20MHz	Sets the
Flash component 2 density	8MB	This fiel
Flash component 1 density	8MB	This fiel
Dual Output Fast Read Support	true	false: Ni
Invalid Instruction 0	0	Op-cod
Invalid Instruction 1	0	Op-cod

To set the size of each flash component, expand the "Descriptor Region" tree node and select the "Component Section" node. The parameters "Flash component 1 density" and "Flash component 2 density" specify the size of each flash component. Double-click on each parameter and select the correct component size from the drop-down list. Click "OK" to update the parameters.

**Note:** The size of the second flash component will only be editable if the number of flash components is set to 2.

Figure 7-5. Descriptor Region - Component Section Options

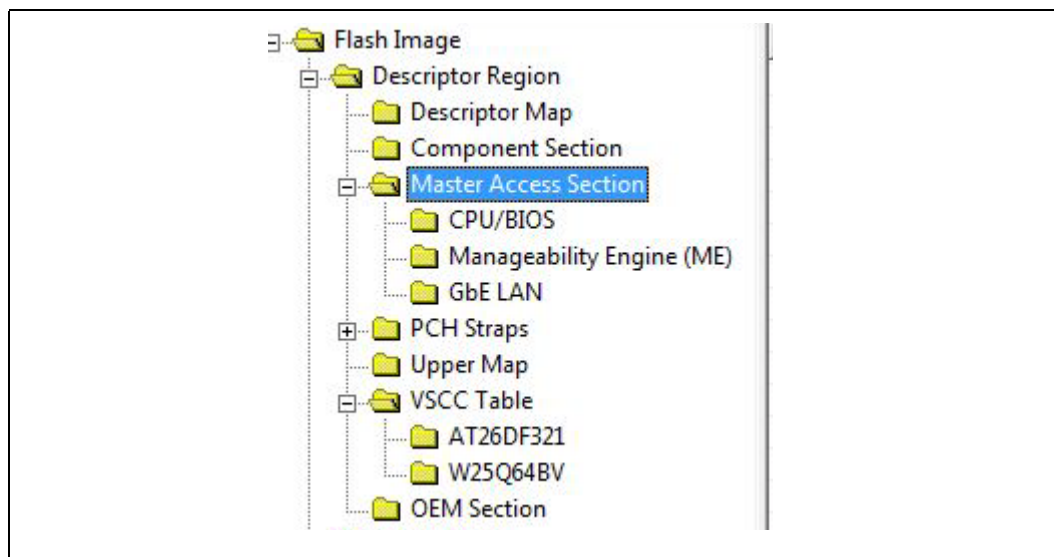
Fast read support	true	Enable
Read clock frequency	20MHz	Set the
Flash component 2 density	8MB	This fi
Flash component 1 density	8MB	This fi
Dual Output Fast Read Support	true	raise:1
Invalid Instruction 0	0	Op-co

The Upper and Lower Flash Erase sizes and Flash Partition Boundary address is not editable from this view. In order to modify these entries you must enter the Build Settings dialog box. Note that Assymmetric flash parts are no longer supported.

### 7.2.1.1 Region Access Control

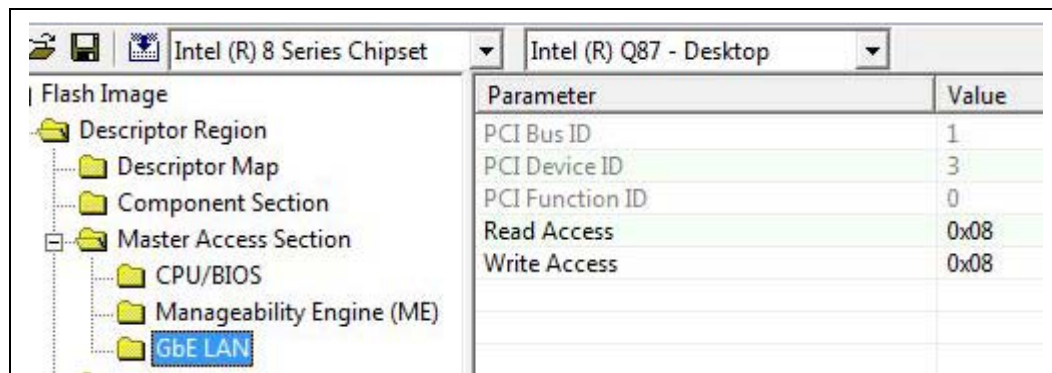
In the Flash Image Tool these access values can be set by selecting the "Descriptor Region" tree node and selecting "CPU/BIOS" under "Master Access Section".

Figure 7-6. Region Access Control



The read and write access hexadecimal values can be specified in the appropriate parameters.

**Figure 7-7. Descriptor Region – Master Access Section Options**



See [Section 4.3](#) for more information.

The setting shown above is the minimum set of the read/write parameters for GbE LAN master access recommended for production phase. It will lock down descriptor region with a necessary level of security for Management Engine enabled systems. There are recommended settings (intended for debug/manufacturing or production phase) provided by FITC for different Master Access. By locking the descriptor region late in the manufacturing flow, the manufacturer has more flexibility in the programming of the flash device. As stated above, once the region is locked, changes to the flash device will be limited.

## 7.3 PCH Soft Straps

These sections contain configuration options for the PCH. The number of Soft Strap sections and their functionality differ based on the target PCH. Please refer to Appendix A and the respective FW Bringup Guide.

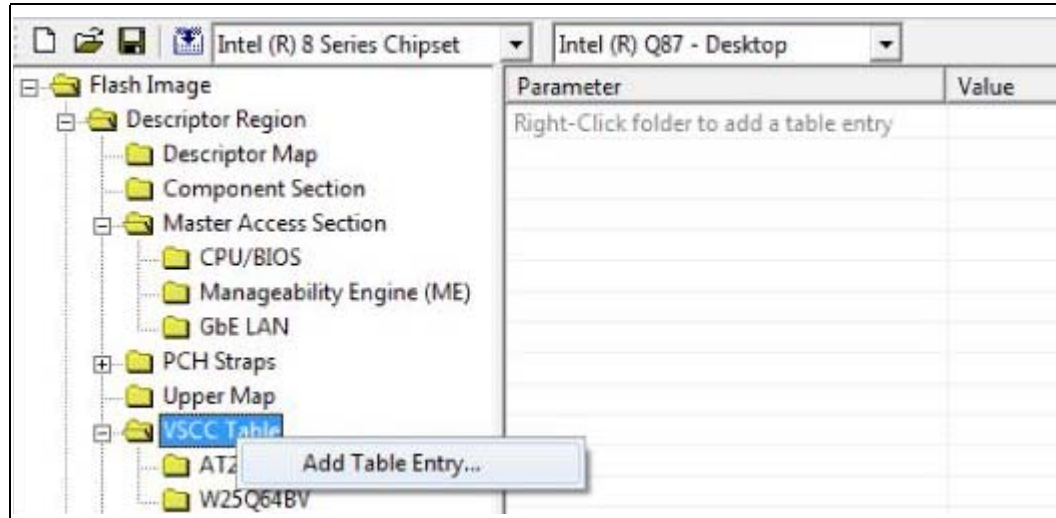
## 7.4 Management Engine VSCC Table

This section is used to store information to setup flash access for ME. This does not have any effect on the usage of the Flash programming Tool (FPT) if the information in this section is incorrect, the Intel ME Firmware may not communicate with the flash device. See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#). This information provided is dependent on the flash device used on the system. for more information. Please contact your flash vendor for information on the specific SPI flash device.

### 7.4.1 Adding a New Table Entry

To add a new table, right click on VSCC table and select add a new table entry.

Figure 7-8. Add New VSCC Table Entry



The program will then prompt the user for a table entry name. To avoid confusion it is recommended that each table entry be unique. FITc will not create an error message for table entries that have the same name.

Figure 7-9. Add VSCC Table Entry



After a table entry has been added, the user will be able to fill in values for the flash device. The values in the VSCC table are provided by your flash vendor. The information in the VSCC table entry is similar to information that is displayed in the [fparts.txt](#) file from the Flash Programming tool. See [8.2 Fparts.txt File](#) for information on how to set the Vendor ID, Device ID 0 and Device ID 1 (three components of JEDEC ID) See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) for more detailed information on how to set the VSCC register value.

Figure 7-10. VSCC Table Entry

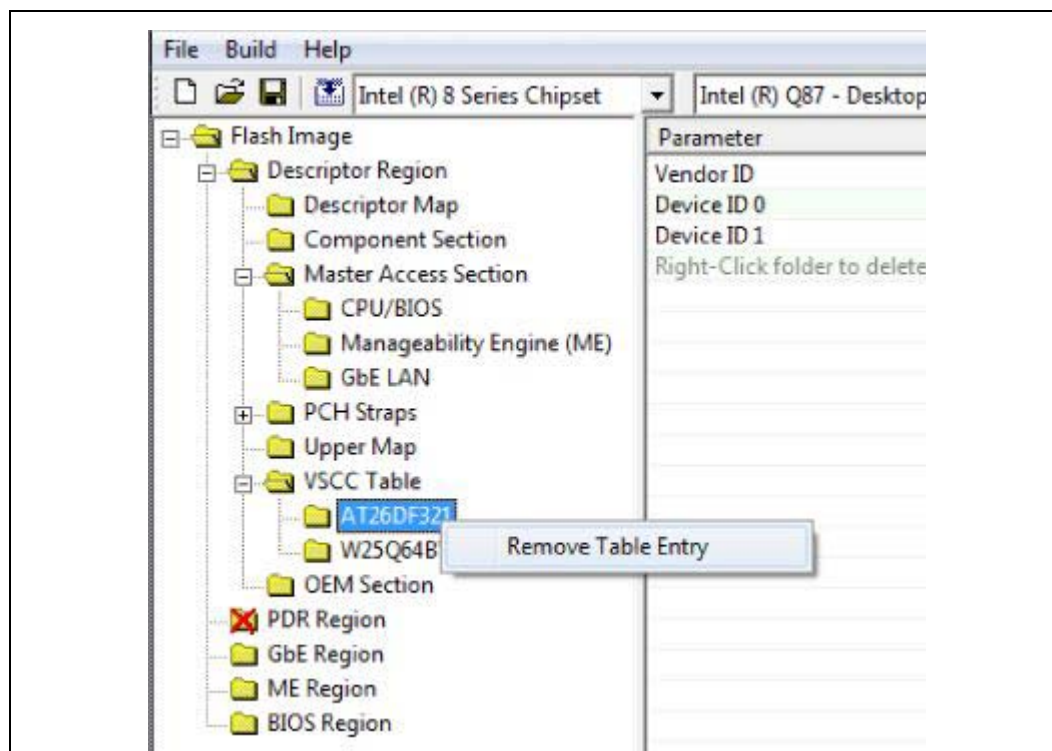
Parameter	Value	Help T
Vendor ID	0x1F	The ve
Device ID 0	0x47	The fir
Device ID 1	0x00	The sec
Right-Click folder to delete this table entry		To dele



## 7.4.2 Removing an Existing Table Entry

To remove an existing table, right click the table that needs to be removed and select remove table. All information in the table along with the table entry will be removed.

Figure 7-11. Remove VSCC Table Entry



§ §





## 8 Flash Programming Tool

---

This is a general overview to the Flash Programming Tool (FPT). Please refer to the documentation that comes with the flash tools executables for the correct feature set for the version of the flash tool being used.

The purpose of the Flash Programming Tool is to program an image file to the flash. The Flash Programming Tool can program the following “regions”, in the form of binary files, into flash. This tool can program an individual region, or the entire flash device.

- Descriptor
- BIOS
- Gigabit Ethernet
- Intel Management Engine
- Platform Data Region

### 8.1 BIOS Support

FPT requires proper opcodes programmed if the FLOCKDN bit is set. Please refer to [6.4 Software Sequencing Opcode Recommendations](#) and [6.3 SPI Protected Range Register Recommendations](#) for more details.

### 8.2 Fparts.txt File

This text file contains a list of all flash devices that this tool supports. If the flash device is not listed below the user will receive the following error:

Flash Programming Tool. Version X.X.X

--- Flash Devices Found ---

>>> Error: There is no supported SPI flash device installed!

If the device is not located in the fparts.txt file, the user is expected to provide information about their device and insert the values into the file using the same format as the rest of the devices. The description and order of the fields is listed below:

1. Display name
2. Device ID (2 or 3 bytes)
3. Device Size (in bits)
4. Block Erase Size (in bytes - 256, 4K, 64K)
5. Block Erase Command
6. Write Granularity (1 or 64)
7. Enable Write Status Register Command (1- True, 0- False)
8. Chip Erase Command
9. Chip Erase Timeout (in milliseconds)



## 8.3 Configuring a Fparts.txt Entry

This section shows how to add support for a flash device for the Flash Programming Tool (fpt.exe/fptw.exe).

Each valid entry in the fparts.txt is comma delineated and has the following fields:

1. Display name
2. Device ID (2 or 3 bytes)
3. Device Size (in bits)
4. Block Erase Size (in bytes - 256, 4K, 64K)
5. Block Erase Command
6. Write Granularity (1 or 64)
7. Enable Write Status Register (50h opcode required to unlock status register)
8. Chip Erase Command
9. Chip Erase Timeout (in milliseconds)

### 8.3.1 Display Name

This is a user defined field that FPT will display on the screen to describe that flash part. It is recommended to use the part number to ensure unique and identifiable entry.

### 8.3.2 Device ID

This is how the flash programming tool identifies a flash part. FPT cycles through three opcodes in order to find a matching entry: JEDEC ID (9Fh), Read ID (90h or ABh) JEDEC ID is a three byte sequence which the industry standard opcode and is guaranteed to be unique to each part number.

When looking in the SPI flash's datasheet for the JEDEC device ID, look for the 9Fh opcode and look for the 3 byte output of that opcode. If there is more than 3 bytes described, just use the first 3 bytes. JEDEC ID, manufacturer ID and Read ID are other keywords to search for.

In parts where JEDEC ID is not available, look for the 2 byte output of 90h or ABh. Read ID is the most common description for this attribute. Read ID is not guaranteed to be unique between different part numbers from the same manufacturer.

### 8.3.3 Device Size (in Bits)

This defines the size of flash space for the flash programming tool. This value is the size of the flash in bits in hexadecimal (0x) notation.

For example 8 Mb part =  $(8 * 1024 * 1024) = (8,388,608)$  convert to hex  $\Rightarrow 0x800000$ .



### 8.3.4 Block Erase Size (in Bytes - 256B, 4K, 64K)

This tells FPT how to properly configure PCH family parts to set the correct erase granularity, or in other words how big of a block gets erased at a time. This value is limited by the flash part and the PCH SPI controller: 256 B, 4 KB or 64 KB.

The SPI flash's data sheet will tell what erase granularity is supported.

For Intel® 8 Series Chipset Platforms, the only granularity supported will be 4 KB.

This field is notated in hexadecimal notation. The choices for this field are: 0x100, 0x1000 (default), or 0x10000.

### 8.3.5 Block Erase Command

This field is the erase command opcode that FPT will use. After the Block Erase size is chosen, use the corresponding opcode in this field. This is a one byte opcode in hexadecimal notation.

For example: 0x20 if the opcode is 20h.

### 8.3.6 Write Granularity (1 or 64)

This field dictates how many bytes will be written for each write command.

Intel® 8 Series Chipset only supports 1 or 64 B writes. Flash devices that allow writes more than a single byte at a time usually support up to 256 bytes at a time. Look to see how many bytes the 02h opcode can support.

64 B has much better write performance, but if any issues are noted, set this field to 1 B write.

This field is in decimal notation. The choices for this field are: 1 or 64.

### 8.3.7 Enable Write Status /Unused

Legacy flash parts may only be able to use 50h opcode in order to unlock the status register. Unlocking the status register is described in detail in section [6.1 Unlocking SPI Flash Device Protection for Intel® 8 Series Chipset Family Platforms](#). This bit should not be set for most flash parts, only those that do not support 06h opcode for unlocking the status register.

### 8.3.8 Chip Erase Command

This command is the one that is used to erase the entire flash part when FPT is used with the /c option. This field is in hexadecimal notation.

Example: 0xC7

§ §





## 9 SPI Flash Programming Procedures

---

This chapter assumes the use of Intel flash tools: Flash Programming Tool and Flash Image Tool (FPT and FITC).

### 9.1 Updating BIOS

If the target system does not have a working BIOS and no alternate method of booting (for example: FWH) then you must use a 3<sup>rd</sup> party out of system programmer. If updating BIOS in a system where the BIOS region is defined in the descriptor, you can use the following command.

```
C:\> fpt /f <file> /bios
```

If unsure that descriptor or the BIOS region is not defined, use fpt /i. Make sure that the descriptor is valid and that BIOS region is large enough to accommodate the intended image. 1-MByte BIOS image (1MB.bin), 2 MByte SPI flash on platform.

#### 9.1.1 Example of SPI flash programming

In system programming

- a. If BIOS image size is an even factor of the total size of flash, it is possible to manipulate image from the DOS prompt to match the size of the flash to ensure the image will be at the top of flash.

```
C:\> copy /b <input file> + <input file> <result file>
```

**Input file** is the name of the BIOS binary that you want to double in size.

**Result file** is the name of resultant binary file.

This DOS command will double the size of the image. Repeat if quadrupling the size is necessary. When the image matches the size of the flash, program the result to flash.

```
C:\> fpt /f <result file>
```

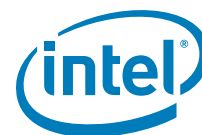
- b. It is possible to program at specific offset. E.g. use fpt to program the one MByte binary image at offset 0x100000.

```
C:\> fpt /f <input file> /address 0x100000
```

§ §







# 10 Intel® Management Engine Disable for Debug/Flash Burning Purposes

---

This section is purely for debug purposes. Intel ME firmware is the only supported configuration for Intel® 8 Series Chipset based system.

## 10.1 Intel® ME Disable

Here are the ways one can disable the Intel ME for purposes of in system programming the flash.

1. Temporarily disable the Intel ME through the MEBX. Power off or cold reset.
2. HDA\_SDO (Manufacturing mode jumper or Flash descriptor override jumper) asserted HIGH on the rising edge of PWROK. Power off or cold reset. Note: this is only valid as long as you do not specifically set the variable Flash Descriptor Override Pin-Strap Ignore in the Flash Image Tool to false.
3. Use -disableME command line option through FPT tool.

HECI ME region unlock - There is a HECI command that allows Intel ME firmware to boot up in a temporarily disabled state and allows for a host program to overwrite the ME region.

**Note:** Removing the DIMM from channel 0 no longer has any effect on Intel Management Engine functionality.

### 10.1.1 Erasing/Programming Intel® ME Region

If CPU/Host has access to ME region, then one could either erase/program the ME region to all FFh. If there is no access, then one must assert HDA\_SDO (Flash descriptor override strap) HIGH during the rising edge of PWROK. If there are Protected Range registers set, then you will not be able to program this w/o a BIOS option to turn off this protected range. (See [6.3 SPI Protected Range Register Recommendations](#)) for more detail.

This depends on the board booting HW defaults for clock configuration. If any clock configuration is required for booting the platform that is not in the HW defaults, then this option may not work for you.

FPT will automatically disable ME when erasing any address in ME region.

§ §



*Intel® Management Engine Disable for Debug/Flash Burning Purposes*



# 11 Recommendations for SPI Flash Programming in Manufacturing Environments for Intel® 8 Series Chipset Family

---

It is recommended that the Intel ME be disabled when you are programming the ME region. Intel Management Engine firmware performs regular writes/erases to the ME region. Therefore some bits may be changed after programming. Please note that not all of these options will be optimal for your manufacturing process.

**Any method of programming SPI flash where the system is not powered will not result in any interference from Management Engine FW. The following methods are for Intel ME FW.**

- Program via In Circuit Test – System is not fully powered here.
- Program via external flash burn-in solution.
- Disable the ME through the BIOS/MEBX before programming fixed offset variables (FOV) into the non-volatile memory area, or before any operation that depends on the base address for fixed variable offsets remaining constant.
- Assert HDA\_SDO HIGH (Flash Descriptor Override Jumper) on the rising edge of PWROK. Note: this is only valid as long as you do not specifically disable this functionality in fixed offset variable.

§ §





# 12 FAQ and Troubleshooting

## 12.1 FAQ

### **Q: What is VSCC and why do I need to set this value?**

**A:** VSCC stands for Vendor Specific Component Capabilities. This defines how BIOS and Intel ME communicate with the SPI flash. Improper BIOS and Intel ME settings can result in improper flash functionality and lead to premature flash wear out. VSCC information is defined in two places. Two host-based VSCC registers (Host VSCC0 Register and Host VSCC1 Register) is in memory mapped space and one table of VSCC entries (Management Engine VSCC Table) is available in the Descriptor Table on the SPI flash. These are separate so Intel ME Firmware does not depend on BIOS for identifying the SPI flash part. This adds some robustness as well as accommodates different BIOS flows where SPI flash is not identified until after the Management Engine needs to access the flash.

The host based VSCC registers must be programmed for any host based application, or integrated GbE software to access the SPI flash. This will have to be done by your BIOS and NOT by FITC. See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) and/or [6.5 Recommendations for Flash Configuration Lockdown and Vendor Component Lock Bits](#) for more information.

The Management Engine VSCC table has no flash parts put in by default. All flash parts that are intended to be used by the platform must have an entry in Management Engine VSCC table. This allows the ability for OEM/ODM to add Intel ME support to any flash parts that meet the requirements defined in the *Intel® 8 Series Chipset Family External Design Specification (EDS)* See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) and [7.4 Management Engine VSCC Table](#) for more information.

### **Q: How do I find the Flash Programming Tool (FPT) and Flash Image Tool (FITC) for my platform?**

**A:** The aforementioned flash tools are included in the system tools director in Intel ME firmware kit (Intel Active Management Technology, Intel ASF, etc.) Please ensure that you download the appropriate kit for the target platform.

Target	Platform Name In VIP/ARMS	Kit Name
ICH8	Averill	Intel® Active Management Technology 2.X (use latest version)
ICH8M	Santa Rosa	Intel® Active Management Technology 2.X (use latest version)
ICH9	Weybridge	Intel® Active Management Technology 3.X (use latest version)
ICH9M	Montevina	Intel® Active Management Technology 4.X (use latest version)
ICH10	McCreary	Intel® Active Management Technology 5.X (use latest version)



Target	Platform Name In VIP/ ARMS	Kit Name
Ibex Peak	Piketon/Kings Creek	Intel® Active Management Technology 6.X (use latest version)
Ibex Peak-M	Calpella	Intel® Active Management Technology 6.X (use latest version)
Cougar Point	Sugar Bay and Bromolow - WS	Intel® Active Management Technology 7.X (use latest version)
Panther Point	Maho Bay	Intel® Active Management Technology 8.X (use latest version)
Lynx Point	Shark Bay	Intel® Active Management Technology 9.X (use latest version)

**Q: How do I build an Image for my Intel® PCH based platform?**

**A:** Intel® 8 Series Chipset family or Lynx Point based platforms, you can follow the appropriate instructions in the FW Bringup Guide which is located in the root directory of the appropriate Intel ME KIT.

**Q: Is my flash part supported by the Flash Programming Tool (FPT)? How can I add support for a new flash to FPT?**

**A:** Look at fparts.txt to see if the intended flash part is present. If the intended flash part meets the guidelines defined in the *Intel® 8 Series Chipset Family External Design Specification (EDS)*, Intel® Management Engine (Intel® ME) Firmware SPI Flash Requirements and support may be added to FPT by referring to [8.3 Configuring a fparts.txt Entry](#)

**Q: Is my flash part supported by Intel ME Firmware? How can I add support for a new flash to Intel ME Firmware?**

**A:** As long as the SPI flash devices meets the requirements defined in the *Intel® 8 Series Chipset Family External Design Specification (EDS)*, support may be added for the device. BIOS will have to set up the Host VSCC registers. The Management Engine VSCC table in the descriptor will also have to be set up in order to get Intel ME firmware to work. See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) and [7.4 Management Engine VSCC Table](#) for more information.

Adding support does not imply validation or guarantee a flash part will work. Platform designers/integrators will have to validate all flash parts with their platforms to ensure full functionality and reliability.

**Q: Why does FPT/verify fail for my system even when I wrote nothing to flash?**

**A:** Intel ME Firmware performs periodic writes to SPI flash when it is active. Due to this the ME region may not match the source file. Please see [11 Recommendations for SPI Flash Programming in Manufacturing Environments for Intel® 8 Series Chipset Family](#) for more information. There are also other system activities beside the Intel ME that can change the data on the flash vs the original image. E.g. the GbE checksum is updated on flash part whenever the value is incorrect.



***Q: How can I overwrite the descriptor when FPT does not have write access?  
How can I overwrite a region that is locked down by descriptor protections?  
How do I write to flash space that is not defined by the descriptor?***

**A:** By asserting HDA\_SDO (flash descriptor override strap) low on the rising edge of PWROK, you can read, write and erase all of SPI flash space regardless of descriptor protections. Any protections imposed by BIOS or directly to the SPI flash part still apply. This should only be used in debug or manufacturing environments. End customers should **NOT** receive systems with this strap engaged.

***Q: I have two flash parts installed on the board. Why does fpt /i only show one flash part?***

**A:** Intel® 8 Series Chipset will not recognize the second SPI flash part unless it is in descriptor mode and the Component section of the descriptor properly describes the flash. Another possibility is that you have two different flash parts and the second flash part is not defined in fparts.txt.

## 12.2 Troubleshooting

***Q: I'm seeing the following error:***

```

Flash Programming Tool. Version 0.8.12
Reading file "fparts.txt" into memory...
Initializing SPI utilities
Reading HSFSTS register... Flash Descriptor: INVALID

--- Flash Devices Found ---

>>> Error: Timedout waiting for hardware to complete read operation!
      SSFSTS register: 0x00

>>> Error: Timedout waiting for hardware to complete read operation!
      SSFSTS register: 0x00

>>> Error: Timedout waiting for hardware to complete read operation!
      SSFSTS register: 0x00

>>> Error: Failed to read the device ID from the flash part!

A:\SR>

```

**A:** You may be using the wrong version of FPT. Please ensure that you are using the flash tools that were provided in the kit for the target systems.

***Q: What does following FPT error message mean?***

**Error: The host does not have write access to the target flash memory!**

**A:** In order for FPT to read or write to a given region, BIOS/Host must have read/write permissions to that target region. This access is set in the descriptor. Look closely at all the addresses defined in the output of FPT /i. If there are any gaps in flash space defined you cannot perform a full flash write. You have to update region by region. Refer to [4.3 Region Access Control](#) for more information. You may have to reflash the descriptor to get the proper access.



**Q: What does following FPT error message mean?**

**Error: Flash program registers are locked! HSFSTS[15] (FLOCKDN).**

**A:** The Flash Configuration Lock-Down (FLCOKDN) bit was set HSFS (hardware sequencing flash status register). This locks down all the program registers in the ICH. If your BIOS and descriptor do not set up Hardware Sequencing, you will have to leave this bit unset in order to use FPT. You may have to upgrade the latest version of FPT as older versions do not support Hardware Sequencing. Please refer to [Hardware Sequencing Flash Status Register](#) in the *Intel® 8 Series Chipset Family External Design Specification (EDS)* for the location for the HSFS. Try reflashing the SPI device with a 3<sup>rd</sup> Party programmer. If you still see this error message, please contact your BIOS vendor to ensure that they are not setting this bit.

**Q: What does following FPT error message mean?**

**Error: There is no supported SPI flash device installed.**

**A:** See the answer to the question above: ***Is my flash part supported by the Flash Programming Tool (FPT)? How can I add support for a new flash to FPT?***

If the tool correctly identifies the flash part installed and still gives an error message like:

--- Flash Devices Found ---

SPI 1234 ID:0x123456 Size: 4096KB (32768Kb)  
Device ID: 0xFFFF not supported.

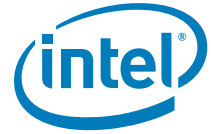
**Error 405: There is no supported SPI flash device installed**

This error will result when the descriptor has two flash parts defined. Edit the image via FIT/FITC and set the number of flash components to 1. See [7.2 Modifying the Flash Descriptor Region](#) for more information.

This error can also result if BIOS has not correctly set up software sequencing. See [6.4 Software Sequencing Opcode Recommendations](#) for Opcodes required for FPT operation.

§ §





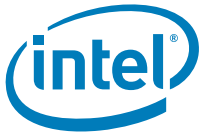
# A APPENDIX A - Descriptor Configuration

---

## A.1 Flash Descriptor PCH Soft Strap Section

The following section describes functionality and how to set soft strapping for a target platform. Improper setting of soft straps can lead to undesired operation and may lead to returns/recalls.

Only default values that will be provided are for softstraps that are reserved.



## A.2 PCHSTRP0—Strap 0 Record (Flash Descriptor Records)

Flash Address: FPSBA + 000h

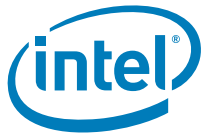
Size: 32 bits

Default Flash Address: 100h

Bits	Description	Usage
31:29	<p><b>Top Swap BIOS Boot-Block size (TSBBBS):</b> Sets Top Swap BIOS boot-block size</p> <p>000: 64 KB. Invert A16 if Top Swap is enabled (Default) 001: 128 KB. Invert A17 if Top Swap is enabled 010: 256 KB. Invert A18 if Top Swap is enabled 011: 512 KB. Invert A19 if Top Swap is enabled (applicable to server sku) 100: 1 MB. Invert A20 if Top Swap is enabled (applicable to server sku) 101 - 111: Reserved</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>This setting is dependent on BIOS architecture and can be different per design. The BIOS developer for the target platform has to determine this value.</li> <li>If FWH is set as Boot BIOS destination then PCH only supports 64 KB Top Swap Boot block size. This value has to be determined by how BIOS implements Boot-Block.</li> <li>Client supports boot block size of up to 256 KB. Boot block size of greater than 256KB is available on <b>server sku</b> only.</li> </ol>	<p>Top Swap BIOS Boot-Block size deals with a BIOS recovery mechanism. It allows for the system to use alternate code in order to boot a platform based upon the <b>Top Swap</b> (GPIO66 pulled low during the rising edge of <b>PWROK</b>) strap being asserted.</p> <p><b>Top Swap</b> inverts an address on access to SPI and firmware hub, so the processor believes its fetches the alternate boot block instead of the original boot-block. The size of the boot-block and setting of this field must be determined by the BIOS developer. If this is not set correctly, then BIOS boot-block recovery mechanism will not work.</p> <p>If BIOS is located on firmware hub, then this value must be set to '00'.</p> <p>Refer to <b>Boot-Block Update Scheme</b> in the latest revision of Intel® 8 Series Express Chipset EDS.</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the architecture of BIOS. The setting of this field must be determined by the BIOS developer.</p>
28:25	Reserved, set to '0'	
24	<p><b>DMI RequesterID Check Disable (DMI_REQID_DIS):</b> The primary purpose of this strap is to support environments with multiple processors that each have a different RequesterID that can each access to Serial Flash.</p> <p>0 = DMI RequesterID Checks are enabled 1 = DMI RequesterID Checks are disabled. No Requester ID checking is done on accesses from DMI.</p>	<p>This bit is only applicable for platforms that contain multiple processor sockets. If multiple processors need to access Serial Flash then this bit would need to set to '1'.</p> <p>Platforms that have a single processor socket set to '0'</p>
23:22	Reserved, set to '0'	
21	<p><b>MACsec Disable (MACSEC_DIS)</b> 0 = MACsec is Enabled 1 = MACsec is Disabled</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>If not using Intel integrated wired LAN or if disabling it, then set to '1'</li> <li>If using Intel integrated wired LAN solution <b>AND</b> the use of MACsec is desired set to '0'.</li> </ol>	<p>MACsec is a hop-by-hop network security solution. It provides Layer 2 encryption and authenticity/integrity protection for packets traveling between MACsec-enabled nodes of the network. The key components that need to support this functionality are the server, client and switch network interface devices.</p> <p>If not using Intel's integrated wired solution, then this field must be set to '1'.</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The platform hardware designer can determine the setting for this</p>



Bits	Description	Usage
20	<p><b>LAN PHY Power Control GPIO12 Select (LANPHYPC_GP12_SEL):</b>            0 = GPIO12 default is General Purpose (GP) output            1 = GPIO12 is used in native mode as LAN_PHY_PWR_CTRL</p> <p><b>Notes:</b>            1. If not using Intel integrated wired LAN or if disabling it, then set to '0'            2. If using Intel integrated wired LAN solution <b>AND</b> if GPIO12 is routed to LAN_DISABLE_N on the Intel PHY, this bit should be set to '1'.</p>	<p>If using Intel integrated wired LAN solution <b>AND</b> if GPIO12 is routed to LAN_DISABLE_N on the Intel PHY, this bit must be set to '1'.</p> <p>If GPIO12 is routed not routed to LAN_DISABLE_N on the Intel PHY, this bit must be set to '0'.</p> <p>If not using Intel integrated wired LAN or if disabling it, this bit must be set to '0'</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The platform hardware designer can determine the setting for this.</p>
19:16	Reserved, set to '0'	
15:14	<p><b>SMLink0 Frequency (SMLOFRQ):</b> These bits determine the physical bus speed supported by the HW.            00: Reserved            01: Standard Mode - up to 100kHz            10: Fast Mode - up to 400kHz            11: Fast Mode Plus - up to 1MHz (default)</p>	Speed is dependent on board topology and layout.
13:12	<p><b>Intel ME SMBus Frequency (SMB0FRQ):</b> The value of these bits determine the physical bus speed supported by the HW.</p> <p>Must be programmed to 01b (100 kHz)</p>	Intel ME SMBus
11:10	<p><b>SMLink1 Frequency (SML1FRQ) Frequency</b>            00: Reserved            01: Standard Mode - up to 100kHz            10: Reserved            11: Reserved</p>	Only supports 100kHz mode.
9	<p><b>SMLink1 Enable (SML1_EN):</b> Configures if SMLink1 segment is enabled            0 = Disabled            1 = Enabled</p> <p><b>Note:</b> This must be set to '1' platforms that use PCH SMBus based thermal reporting.</p>	<p>This bit must be set to '1' if using the PCH's Thermal reporting. If setting this bit to '0', there must be an external solution that gathers temperature information from PCH and processor.</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The setting of this field must be determined by the BIOS developer and the platform hardware designer.</p>



Bits	Description	Usage
8	<p><b>SMLink0 Enable (SMLO_EN):</b> Configures if SMLink0 segment is enabled</p> <p>0 = Disabled 1 = Enabled</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. This bit MUST be set to '1' when Intel NFC enabled on the platform.</li> <li>2. The Intel PHY SMBus controller must be routed to this SMLink 0 Segment.</li> <li>3. This segment should be set to 0 in one of the following cases: <ol style="list-style-type: none"> <li>a. Not using Intel NFC solution</li> <li>b. Disabled by the user.</li> </ol> </li> </ol>	<p>This bit MUST be set to '1' when utilizing Intel NFC solution on the platform.</p> <p>The Intel PHY SMBus controller must be routed to this SMLink 0 Segment.</p> <p>If not using Intel NFC solution or if disabling it, then this segment must be disabled (set to '0').</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The setting of this field must be determined by the BIOS developer and the platform hardware designer.</p>
7	<p><b>Intel ME SMBus Select (SMB_EN):</b> Configures if the ME SMBus segment is enabled</p> <p>0 = Disabled 1 = Enabled</p> <p><b>Note:</b> This bit MUST be set to '1'.</p>	<p>This bit must always be set to '1'.</p>
6:3	Reserved, set to '0100b'	
2:1	<b>Chipset Configuration Softstrap 1:</b> Must be set to 01b.	
0	Reserved, set to '0'	



## A.3 PCHSTRP1—Strap 1 Record (Flash Descriptor Records)

Flash Address: FPSBA + 004h

Default Value: 0000000Fh

Size: 32 bits

Default Flash Address: 104h

Bits	Description	Usage
31:28	Reserved, set to '0'	
27:26	<b>SPI TPM Clock Frequency (STCF)</b> 00: 20Mhz 01: 33Mhz (Default) All other settings is reserved.	This field identifies the frequency that should be used with the TPM on SPI. This field is undefined if the TPM on SPI is disabled by softstrap
25	<b>TPM on SPI (TOS)</b> 0 = TPM is not on SPI (Default) 1 = TPM is on SPI	
24:9	Reserved, set to '0'.	
8	Chipset configuration, set to '1'	
7	<b>Dual Output Read Enable (DORE):</b> '0': Dual Output Read is disabled '1': Dual Output Read is enabled (Default)	This soft strap only has effect if Dual Output read is discovered as supported via the SFDP If parameter table is not detected via SFDP, this bit has no effect and Dual Output Read is controlled via the Flash Descriptor Component Section. Dual Output Fast Read Support Bit
6	<b>Dual I/O Read Enable (DIORE)</b> '0': Dual I/O Read is disabled '1': Dual I/O Read is enabled (Default)	this soft strap only has effect if Dual I/O Read is discovered as supported via the SFDP
5	<b>Quad Output Read Enable (QORE):</b> '0': Quad Output Read is disabled '1': Quad Output Read is enabled (Default)	This soft strap only has effect if Quad Output Read is discovered as supported via the SFDP
4	<b>Quad I/O Read Enable (QIORE):</b> '0': Quad I/O Read is disabled '1': Quad I/O Read is enabled (Default)	This soft strap only has effect if Quad Output Read is discovered as supported via the SFDP
3:0	<b>Chipset Configuration Softstrap 2:</b> Must be set to Fh.	



## A.4 PCHSTRP2—Strap 2 Record (Flash Descriptor Records)

Flash Address: FPSBA + 008h      Size: 32 bits

Default Flash Address: 108h

Bits	Description	Usage
31:25	<b>Intel® ME SMBus I<sup>2</sup>C Address (MESMI2CA):</b> Defines 7 bit Intel ME SMBus I <sup>2</sup> C target address  <b>Note:</b> This field is only used for testing purposes	This address is only used by Intel ME FW for testing purposes. If <b>MESMI2CEN (PCHSTRP2 bit 24)</b> is set to 1 then the address used in this field must be non-zero and not conflict with any other devices on the segment.
24	<b>Intel ME SMBus I<sup>2</sup>C Address Enable (MESMI2CEN):</b> 0 = Intel ME SMBus I <sup>2</sup> C Address is disabled 1 = Intel ME SMBus I <sup>2</sup> C Address is enabled  <b>Note:</b> This field is only used for testing purposes on Intel ME FW	This field should only be set to '1' for testing purposes
23:17	<b>Intel ME SMBus MCTP Address (MESMMCTPA):</b> Defines 7 bit Intel ME SMBus MCTP target address  <b>Note:</b> This field is only used for testing purposes on Intel ME FW.	This address is used by Intel ME Anti-Theft Technology .  If <b>MESMI2CEN (PCHSTRP2 bit 24)</b> is set to 1 then the address used in this field must be non-zero and not conflict with any other devices on the segment.
16	<b>Intel ME SMBus MCTP Address Enable (MESMMCTPAEN):</b> 0 = Intel ME SMBus MCTP Address is disabled 1 = Intel ME SMBus MCTP Address is enabled  <b>Note:</b> This field is only used for testing purposes on Intel ME FW	This field should only be set to '1' for testing purposes on platforms that use Intel ME FW.



Bits	Description	Usage
15:9	<p><b>Intel ME SMBus Alert Sending Device (ASD) Address (MESMASDA):</b> Intel ME SMBus Controller ASD Target Address.</p> <p><b>Note:</b> This field is only applicable if there is an ASD attached to SMBus and using Intel AMT</p>	<p>If <b>MESMASDEN(PCHSTRP2 bit 8)</b> is set to '1' there must be a valid address for ASD. The address must be determined by the BIOS developer based on the requirements below.</p> <p>A valid address must be:</p> <ul style="list-style-type: none"> <li>• Non-zero value</li> <li>• Must be a unique address on the Host SMBus segment</li> <li>• Be compatible with the master on SMBus - For example, if the ASD address the master that needs write thermal information to an address "xy"h. Then this field must be set to "xy"h.</li> </ul>
8	<p><b>Intel ME SMBus Alert Sending Device (ASD) Address Enable (MESMASDEN):</b> 0 = Intel ME SMBus ASD Address is disabled 1 = Intel ME SMBus ASD Address is enabled</p> <p><b>Note:</b> This field is only applicable if there is an ASD attached to SMBus and using Intel AMT</p>	<p>This bit must only be set to '1' when there is an ASD (Alert Sending Device) attached to Host SMBus. This is only applicable in platforms using Intel AMT.</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The setting of this field must be determined by the BIOS developer and the platform hardware designer.</p>
7:0	Reserved, set to '0'	



## A.5 PCHSTRP3—Strap 3 Record (Flash Descriptor Records)

Flash Address: FPSBA + 00Ch      Default Value: 00000000h      Size: 32 bits

Default Flash Address: 10Ch

Bits	Description	Usage
31:0	Reserved, set to '0'	

## A.6 PCHSTRP4—Strap 4 Record (Flash Descriptor Records)

Flash Address: FPSBA + 010h      Size: 32 bits

Default Flash Address: 110h

Bits	Description	Usage
31:24	Reserved, set to '0'	
23:17	<b>GbE PHY SMBus Address:</b> This is the 7 bit SMBus address the PHY uses to accept SMBus cycles from the MAC.  <b>Note:</b> This field must be programmed to 64h.	This is the Intel PHY's SMBus address. This field must be programmed to 64h.  GbE PHY SMBus Address and GbE MAC address have to be programmed to 64h and 70h in order to ensure proper arbitration of SMBus communication between the Intel integrated MAC and PHY.
16	Reserved, set to '0'	
15:9	<b>GbE MAC SMBus Address:</b> This is the 7 bit SMBus address uses to accept SMBus cycles from the PHY.  <b>Note:</b> This field must be programmed to 70h.	This is the Intel integrated wired MAC's SMBus address. This field must be programmed to 70h.  GbE PHY SMBus Address and GbE MAC address have to be programmed to 64h and 70h in order to ensure proper arbitration of SMBus communication between the Intel integrated MAC and PHY.
8	<b>Gbe MAC SMBus Address Enable (GBEMAC_SMBUS_ADDR_EN):</b> 0 = Disable 1 = Enable  <b>Notes:</b> 1. This bit MUST be set to '1' when utilizing Intel integrated wired LAN. 2. If not using Intel integrated wired LAN solution or if disabling it, then this segment must be set to '0'.	This bit must be set to '1' if Intel integrated wired LAN solution is used.  If not using, or if disabling Intel integrated wired LAN solution, then this field must be set to '0'.





Bits	Description	Usage
7:4	Reserved, set to '0'	
3:2	<b>SATA Port 5 PCIe Port 2 Mode (SATAP5_PCIEP2_MODE):</b> 00 : Statically assigned to SATA Port 5 01 : Statically assigned to PCIe Port 2 10 : Reserved 11 : Assigned based on the native mode of GPIO49 pin. If the native GPIO49 pin is a '1', then it is assigned to SATA Port 5, else it is assigned to PCIe Port 2.	If this soft strap is set to "11" then GPIO49 native mode is SATA5_PCIE2#, else the native mode is SATA5GP.
1:0	<b>Intel PHY Connectivity (PHYCON[1:0]):</b> This field determines if Intel wired PHY is connected to SMLink0  00: No Intel wired PHY connected 10: Intel wired PHY on SMLink0 All other values Reserved  <b>Notes:</b> 1. This bit MUST be set to '10' when utilizing Intel integrated wired LAN. 2. If not using, or if disabling Intel integrated wired LAN solution, then this segment must be set to 00b.	This field must be set to "10" if Intel integrated wired LAN solution is used.  If not using, or if disabling Intel integrated wired LAN solution, then field must be set to "00".

## A.7 PCHSTRP5—Strap 5 Record (Flash Descriptor Records)

Flash Address: FPSBA + 014h      Default Value: 00000000h      Size: 32 bits

Default Flash Address: 114h

Bits	Description	Usage
31:0	Reserved, set to '0'	

## A.8 PCHSTRP6—Strap 6 Record (Flash Descriptor Records)

Flash Address: FPSBA + 018h      Default Value: 00000000h      Size: 32 bits

Bits	Description	Usage
31:0	Reserved, set to '0'	



## A.9 PCHSTRP7—Strap 7 Record (Flash Descriptor Records)

Flash Address: FPSBA + 01Ch

Default Value: 00000000h

Size: 32 bits

Bits	Description	Usage
31:0	<b>Intel ME SMBus Subsystem Vendor and Device ID</b> (MESMA2UDID): MESMAUDID[15:0] - Subsystem Vendor ID MESMAUDID[31:16] - Subsystem Device ID  The values contained in MESMAUDID[15:0] and MESMAUDID[31:16] are provided as bytes 8-9 and 10-11 of the data payload to an external master when it initiates a Directed GET UDID Block Read Command to the Alert Sending Device ASD's address.	This bit must only be set to '1' when there is an ASD (Alert Sending Device) attached to SMBus and when <b>MESMASDEN(PCHSTRP2 bit 8)</b> is set to '1'. This is only applicable in platforms using Intel® AMT. Set this if you want to add a 4 byte payload to an external master when a GET UDID Block read command is made to Intel ME SMBus ASD's address.

## A.10 PCHSTRP8—Strap 8 Record (Flash Descriptor Records)

Flash Address: FPSBA + 020h

Size: 32 bits

Default Flash Address: 120h

Bits	Description	Usage
31:0	Reserved, set to '0'	



## A.11 PCHSTRP9—Strap 9 Record (Flash Descriptor Records)

Flash Address: FPSBA + 024h

Size: 32 bits

Default Flash Address: 124h

Bits	Description	Usage
31:30	<b>SATA Port 4 PCIe Port 1 Mode (SATAP4_PCIEP1_MODE)</b> 00: Statically assigned to SATA Port 4 01: Statically assigned to PCIe Port 1 10: Reserved 11: Assigned based on the native mode of GPIO16 pin. If the native GPIO16 pin is a '1', then it is assigned to SATA Port 4, else it is assigned to PCIe Port 1.	If this soft strap is set to "11" then GPIO16 native mode is SATA4_PCIE1#, else the native mode is SATA4GP.
29:28	Must be set to 11b.	
27:23	Reserved, set to '0'.	
22	<b>PCHHOT# or SML1ALERT# Select (PCHHOT#_SML1ALERT#_SEL)</b> This strap determines the native mode operation of GPIO74 0 = SML1ALERT# is the native functionality of GPIO74 1 = PCHHOT# is the native functionality of GPIO74	<b>PCHHOT#</b> is used to indicate the PCH temperature out of bounds condition to an external agent such as BMC or EC, when PCH temperature is greater than value programmed by BIOS.
21:20	<b>USB3 Port 4 PCIe Port 2 Mode (USB3P4_PCIEP2_MODE)</b> 00: PCIe Lane 2 is statically assigned to PCIe Express (or GbE) 01: PCIe Lane 2 is statically assigned to USB3 Port 4 10: Reserved 11: Reserved	
19:18	<b>USB3 Port 3 PCIe Port 1 Mode (USB3P3_PCIEP1_MODE)</b> 00: PCIe Lane 1 is statically assigned to PCIe Express (or GbE) 01: PCIe Lane 1 is statically assigned to USB3 Port 3 10: Reserved 11: Reserved	
17:15	Reserved, set to '0'	
14	<b>Subtractive Decode Agent Enable (SUB_DECODE_EN)</b> 0 = Disables PCH PCIe ports from Subtractive Decode Agent 1 = Enables PCH's PCIe ports to behave as a subtractive decode agent  <b>Note:</b> If connecting a PCI bridge chip to the PCH that requires the PCH to behave as a subtractive decode agent, then set this bit to '1'.	Set this bit to '1' if there is a PCI bridge chip connected to the PCH, that requires subtractive decode agent. Set to '0' if the platform has no PCI bridge chip.  <b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The setting of this field must be determined by the platform hardware designer.
13:12	Reserved, set to '0'	



Bits	Description	Usage
11	<b>Intel PHY Over PCI Express* Enable (PHY_PCIE_EN):</b> 0 = Intel integrated wired MAC/PHY communication is not enabled over PCI Express*. 1 = The PCI Express* port selected by the <b>PHY_PCIEPORT_SEL</b> soft strap to be used by Intel PHY  <b>Note:</b> This bit must be "1" if using Intel integrated wired LAN solution.	This bit MUST be set to '1' if using Intel integrated wired LAN solution.  If not using, or if disabling Intel integrated wired LAN solution then set this to '0'.
10:8	<b>Intel PHY PCIe* Port Select (PHY_PCIEPORTSEL):</b> Sets the default PCIe* port to use for Intel integrated wired PHY.  000: Port 1 001: Port 2 010: Port 3 011: Port 4 100: Port 5 101: Port 6 110: Port 7 111: Port 8  <b>Note:</b> This field only applies when <b>PHY_PCIE_EN</b> = '1'. Set to 000b when <b>PHY_PCIE_EN</b> is set to '0'	This field tells the PCH which PCI Express* port an Intel PHY is connected.  If <b>PHY_PCIE_EN</b> is = '0', then this field is ignored.  <b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The platform hardware designer or schematic review can determine what PCIe* Port the Intel wired PHY is routed.
7	<b>Chipset Configuration Softstrap 3</b> Set to '1'b	
6	<b>DMI Lane Reversal (DMILR).</b>  0 = DMI Lanes 0 - 3 are not reversed. 1 = DMI Lanes 0 - 3 are reversed.	This field is used only when DMI Lanes are reversed on the layout. This usually only is done on layout constrained boards where reversing lanes help routing.  <b>Note:</b> This setting is dependent on the board design. The platform hardware designer must determine if DMI needs lane reversal.
5	<b>PCIe Lane Reversal 2 (PCIELR2).</b> This bit lane reversal behavior for PCIe Port 5 if configured as a x4 PCIe* port.  0 = PCIe Lanes 5-8 are not reversed. 1 = PCIe Lanes 5-8 are reversed when Port 5 is configured as a 1x4.  <b>Note:</b> This field only is in effect if PCIEPCS2 is set to '11'b.	If configuring PCIe* port 5 as a x4 PCIe* bus, reversing the lanes of this port is done via this strap.  PCI Express* port lane reversal can be done to aid in the laying out of the board.  <b>Note:</b> This setting is dependent on the board design. The platform hardware designer must determine if this port needs lane reversal.



Bits	Description	Usage
4	<b>PCIe Lane Reversal 1 (PCIELR1).</b> This bit lane reversal behavior for PCIe Port 1 if configured as a x4 PCIe port.  0 = PCIe Lanes 1-4 are not reversed. 1 = PCIe Lanes 1-4 are reversed when Port 1 is configured as a 1x4.  <b>Note:</b> This field only is in effect if PCIEPCS1 is set to '11'b.	If configuring PCIe port 1 as a x4 PCIe bus, reversing the lanes of this port is done via this strap.  PCI Express* port lane reversal can be done to aid in the laying out of the board.  <b>Note:</b> This setting is dependent on the board design. The platform hardware designer can determine if this port needs lane reversal
3:2	<b>PCI Express Port Configuration Strap 2 (PCIEPCS2).</b> Straps to set the default value of the PCI Express port Configuration 2 register covering PCIe ports 5-8.  11: 1x4 Port 5 (x4), Ports 6-8 (disabled) 10: 2x2 Port 5-6 (x2) and Port 7-8 (x2) 01: 1x2, 2x1 Port 5 (x2), Port 6 (disabled), Ports 7 (x1) and Port 8 (x1) 00: 4x1 Port 5 (x1), Ports 6 (x1), Ports 7 (x1) and Port 8 (x1)	Setting of this field depend on what PCIe ports 5-8 configurations are desired by the board manufacturer. Only the x4 configuration ("11") has the option of lane reversal if PCIELR2 is set to '1'.  <b>Note:</b> This field must be determined by the PCI Express* port requirements of the design. The platform hardware designer must determine this setting. <b>Note:</b> Refer to Lynx Point Flexible IO configuration tool user guide (doc#501026) for possible configuration settings.
1:0	<b>PCI Express Port Configuration Strap 1 (PCIEPCS1).</b> Straps to set the default value of the PCI Express Port Configuration 1 register covering PCIe ports 1-4.  11: 1x4 Port 1 (x4), Ports 2-4 (disabled) 10: 2x2 Port 1-2 (x2) and Port 3-4 (x2) 01: 1x2, 2x1 Port 1 (x2), Port 2 (disabled), Ports 3 and Port 4 (x1) 00: 4x1 Port 1 (x1), Ports 2 (x1), Ports 3 (x1) and Port 8 (x1)	Setting of this field depend on what PCIe ports 1-4 configurations are desired by the board manufacturer. Only the x4 configuration ("11") has the option of lane reversal if PCIELR1 is set to '1'.  <b>Note:</b> This field must be determined by the PCI Express* port requirements of the design. The platform hardware designer must determine this setting. <b>Note:</b> Refer to Lynx Point Flexible IO configuration tool user guide (doc#501026) for possible configuration settings.



## A.12 PCHSTRP10—Strap 10 Record (Flash Descriptor Records)

Flash Address: FPSBA + 028h      Size: 32 bits

Default Flash Address: 128h

Bits	Description	Usage
31:25	Reserved set to '0'	
24	<b>Intel® ME Debug LAN Emergency Mode</b> 0 = Intel ME LAN Debug Disable 1 = Enables Emergency mode of Intel ME LAN Debug mode	<b>Note:</b> Default for production platforms should be '0'
23	<b>Deep SX Support (Deep_SX_EN)</b> 0 = Deep SX NOT supported on the platform 1 = Deep SX supported on the platform	This requires the target platform to support Deep SX state
22	<b>Integrated Clocking Controller (ICC) Profile Selection (ICC_PRO_SEL)</b> 0 = ICC Profile will be provided by BIOS 1 = ICC Profile selected by Softstraps (ICC_SEL)	
21	<b>Intel® ME Reset Capture on CL_RST1#: (MER_CL1)</b> 0 = PCH Signal CL_RST1# does NOT assert when Intel ME performs a reset. 1 = PCH Signal CL_RST1# asserts when Intel ME resets.	<b>Notes:</b> Signal CL_RST1# is only present on mobile PCH
20:17	Reserved set to '0'	
16	<b>ME Debug Extended Data Enable</b> 0 = Disable 1 = Enable	
15:9	<b>ME Debug SMBus Emergency Mode Address (MDSMBE_ADD):</b> SMBUS address used for ME Debug status writes. If this field is 00h, the default address, 38h, is used.	This field is only used for testing purposes.
8	<b>ME Debug SMBus Emergency Mode Enable (MDSMBE_EN):</b> 0 = Disable Intel ME Debug status writes 1 = Enable Intel ME Debug status writes over SMBUS using the address set by MMADDR.	This field is only used for testing purposes. When this bit is enabled, you will see writes on SMBus to address 38h bits address (70h bit shifted), or value is specified in <b>MDSMBE_ADD</b> . <b>MDSMBE_ADD</b> specifies address bits [7:1] of the target address.
7	Reserved.	
6:2	Reserved, set to '0'	
1	<b>ME Boot Flash (ME_Boot_Flash).</b> 0 = Intel Management Engine will boot from ROM, then flash 1 = Intel Management Engine will boot from flash  <b>Note:</b> This field should only be set to '1b' if the Intel ME binary loaded in the platform has a ME ROM Bypass image	This bit must be set to 0 for production PCH based platforms.  This bit will only be set to '1' in order to work around issues in pre-production hardware and Intel ME FW.
0	Reserved, Default set to '0'	



## A.13 PCHSTRP11—Strap 11 Record (Flash Descriptor Records)

Flash Address: FPSBA + 02Ch Size: 32 bits

Default Flash Address: 12Ch

Bits	Description	Usage
31:25	<p><b>SMLink1 I<sup>2</sup>C* Target Address (SML1I2CA)</b> Defines the 7 bit I<sup>2</sup>C target address for PCH Thermal Reporting on SMLink1.</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>This field is not active unless SML1I2CAEN is set to '1'.</li> <li>This address MUST be set if there is a device on the SMLink1 segment that will use thermal reporting supplied by PCH.</li> <li>If SML1I2CAEN = '1' then this field must be a valid 7 bit, non-zero address that does not conflict with any other devices on SMLink1 segment.</li> <li>This address can be different for every design, ensure BIOS developer supplies the address.</li> </ol>	<p>When <b>SML1I2CAEN(PCHSTRP11 bit 24)</b> = '1', there needs to be a valid I<sup>2</sup>C address in this field. This address used here is design specific. The BIOS developer and/or platform hardware designer must supply an address with the criteria below.</p> <p>A valid address must be:</p> <ul style="list-style-type: none"> <li>Non-zero value</li> <li>Must be a unique address on the SMLink1 segment</li> <li>Be compatible with the master on SMLink1 - For example, if the I<sup>2</sup>C address the master that needs write thermal information to a address "xy"h. Then this field must be to "xy"h.</li> </ul>
24	<p><b>SMLink1 I<sup>2</sup>C Target Address Enable (SML1I2CAEN)</b> 0 = SMLink1 I<sup>2</sup>C Address is disabled 1 = SMLink1 I<sup>2</sup>C Address is enabled</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>This bit MUST set to '1' if there is a device on the SMLink1 segment that will use PCH thermal reporting.</li> <li>This bit MUST be set to '0' if PCH thermal reporting is not used.</li> </ol>	<p>This bit must be set in cases where SMLink1 has a master that requires SMBus based Thermal Reporting that is supplied by the PCH. Some examples of this master could be an Embedded Controller, a BMC, or any other SMBus Capable device that needs Processor and/or PCH temperature information. If no master on the SMLink1 segment is capable of utilizing thermal reporting, then this field must be set to '0'.</p> <p><b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The setting of this field must be determined by the BIOS developer and the platform hardware designer.</p>
23:8	Reserved, set to '0'	
7:1	<p><b>SMLink1 GP Address (SML1GPA):</b> SMLink1 controller General Purpose Target Address (7:1)</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>This field is not active unless <b>SML1GPAEN</b> is set to '1'.</li> <li>This address MUST be set if there is a device on the SMLink1 segment that will use SMBus based PCH thermal reporting.</li> <li>If <b>SML1GPAEN</b> = '1' then this field must be a valid 7 bit, non-zero address that does not conflict with any other devices on SMLink1 segment.</li> </ol>	<p>When <b>SML1GPAEN</b> = '1', there needs to be a valid GP address in this field. This address used here is design specific. The BIOS developer and/or platform hardware designer must supply an address with the criteria below.</p> <p>A valid address must be:</p> <ul style="list-style-type: none"> <li>Non-zero value</li> <li>Must be a unique address on the SMLink1 segment</li> <li>Be compatible with the master on SMLink1 - For example if the GP address the master that needs read thermal information from a certain address, then this field must be set accordingly.</li> </ul>



Bits	Description	Usage
0	<b>SMLink1 GP Address Enable(SML1GPAEN):</b> SMLink1 controller General Purpose Target Address Enable 0 = SMLink1 GP Address is disabled 1 = SMLink1 GP Address is enabled  <b>Notes:</b> 1. This bit MUST set to '1' if there is a device on the SMLink1 segment that will use SMBus based PCH thermal reporting. 2. This bit MUST be set to '0' if PCH thermal reporting is not used.	This bit must be set in cases where SMLink1 has a master that requires SMBus based Thermal Reporting that is supplied by the PCH. Some examples of this master could be an Embedded Controller, a BMC, or any other SMBus Capable device that needs Processor or PCH temperature information. If no master on the SMLink1 segment is capable of utilizing thermal reporting, then this field must be set to '0'.  <b>Note:</b> This setting is not the same for all designs, is dependent on the board design. The setting of this field must be determined by the BIOS developer and the platform hardware designer.

## A.14 PCHSTRP12—Strap 12 Record (Flash Descriptor Records)

Flash Address: FPSBA + 030h      Default Value: 00000000h      Size: 32 bits

Default Flash Address: 130h

Bits	Description	Usage
31:0	Reserved, set to '0'	

## A.15 PCHSTRP13—Strap 13 Record (Flash Descriptor Records)

Flash Address: FPSBA + 034h      Default Value: 00000000h      Size: 32 bits

Default Flash Address: 134h

Bits	Description	Usage
31:0	Reserved, set to '0'	

## A.16 PCHSTRP14—Strap 14 Record (Flash Descriptor Records)

Flash Address: FPSBA + 038h      Default Value: 00000000h      Size: 32 bits

Default Flash Address: 138h

Bits	Description	Usage
31:0	Reserved, set to '0'	





## A.17 PCHSTRP15—Strap 15 Record (Flash Descriptor Records)

Flash Address: FPSBA + 03Ch

Size:

32 bits

Default Flash Address: 13Ch

Recommended Value:

Bits	Description	Usage
31:25	Reserved, set to '0'	
24	<b>PCIe Power Stable Timer (t205b timer)</b> 0 = t205b timer is disabled (default) 1 = PCH will count 99ms from PWROK assertion before PLTRST# is de-asserted.	Board dependent. Default is disabled, Platform is required to ensure timing of PWROK and SYS_PWROK in such a way that it satisfies the PCIe timing requirement of power stable to reset de-assertion.
23:21	Reserved, set to '0'	
20	<b>DeepSx Platform Configuration (DEEPSX_PLT_CFG_SS)</b> 0 = The platform does not support DeepSx. 1 = The platform supports DeepSx.	
19	Reserved, set to '1b'	
18:16	Reserved, set to '0'	
15	<b>SLP_WLAN# or GPIO29/MGPIO3 Select (SLP_WLAN#_GP29MGPIO3_SEL)</b> 0 = SLP_WLAN# 1 = GPIO29/ MGPIO3	
14	Reserved, set to '1b'	
13:12	Reserved, set to '0'	
11:10	<b>t210 Timing</b> 00: 1 ms (default) 01: 30 us 10: 5 ms 11: 2 ms	t210: PROCPWRGD and SYS_PWROK high to SUS_STAT# deassertion. Refer to EDS for details.
9:8	<b>t209 Timing</b> 00: 100 ms 01: 50 ms 10: 5 ms 11: 1 ms (default)	t209: PCH clock output stable to PROCPWRGD high. Refer to EDS for details.



Bits	Description	Usage
7	Reserved, set to '0'	
6	<b>Intel integrated wired LAN Enable (IWL_EN)</b> 0 = Disable Intel integrated wired LAN Solution 1 = Enable Intel integrated wired LAN Solution  <b>Notes:</b> This must be set to '1' if the platform is using Intel's integrated wired LAN solution. Set to '0' if not using Intel integrated wired LAN solution or if disabling it.	This must be set to '1' if the platform is using Intel's integrated wired LAN solution.  This must be set to '0' if not using Intel's integrated wired LAN solution or if disabling it.
5:0	<b>Chipset Configuration Softstrap 4</b> Set to '111110'b	

## A.18 PCHSTRP16—Strap 16 Record (Flash Descriptor Records)

Flash Address: FPSBA + 040h

Size:

32 bits

Default Flash Address: 140h

Recommended Value:

Bits	Description	Usage
31:0	Reserved, set to '0'	

## A.19 PCHSTRP17—Strap 17 Record (Flash Descriptor Records)

Flash Address: FPSBA + 044h

Size:

32 bits

Default Flash Address: 144h

Recommended Value:

Bits	Description	Usage
31:2	Reserved, set to '0'	
1	<b>Chipset Configuration Softstrap 5:</b> Set to '1'b	
0	<b>Clock Mode Selection (FCIM/BTM)</b> 0 = Full Clock Integrated Mode (FCIM) 1 = Buffered Through Mode (BTM)	



## A.20 PCHSTRP18—Strap 18 Record (Flash Descriptor Records)

Flash Address: FPSBA + 048h  
Default Flash Address: 148h

Size: 32 bits

Recommended Value:

Bits	Description	Usage
31:0	Reserved, set to '0'	

## A.21 PCHSTRP19—Strap 19 Record (Flash Descriptor Records)

Flash Address: FPSBA + 04Ch  
Default Flash Address: 14Ch

Size: 32 bits

Recommended Value:

Bits	Description	Usage
31:3	Reserved, set to '0'	
2	Reserved, set to '1'	
1:0	Reserved, set to '0'	

## A.22 PCHSTRP20—Strap 20Record (Flash Descriptor Records)

Flash Address: FPSBA + 050h  
Default Flash Address: 150h

Size: 32 bits

Recommended Value:

Bits	Description	Usage
31:0	Reserved, set to '0'	



## A.23 CPUSTRP0—Strap 0 Record (Flash Descriptor Records)

Flash Address: FCPUSBA + 000h  
Default Flash Address: 200h

Size: 32 bits

Recommended Value:

Bits	Description	Usage
31:0	Reserved, set to '0'	



## A.24 Softstrap Step Through

General questions help in setting softstraps and certain other descriptor values.

For All configurations the following must be set.

Name	Location	Value
SMB_EN	PCHSTRP0[7]	1b

### A.24.1 Does the target platform use the Intel integrated wired LAN solution?

1. If Yes,

Name	Location	Value
SML0_EN	PCHSTRP0[8]	1b
GBEPHY_SMBUS_ADDR	PCHSTRP4[23:17]	C4h
GBEMAC_SMBUS_ADDR	PCHSTRP4[15:9]	E0h
GBE_SMBUS_ADDR_EN	PCHSTRP4[8]	1b
PHYCON[1:0]	PCHSTRP4[1:0]	10b
PHY_PCIE_EN	PCHSTRP9[11]	1b
IWL_EN	PCHSTRP15[6]	1b

a. What PCIe\* port is the Intel PHY attached? Note: Intel CRBs use port 6.

Name	Location	Value
PHY_PCIEPORTSEL	PCHSTRP9[10:8]	000b: Port 1, 001b: Port 2, 010b: Port 3, 011b: Port 4, 100b: Port 5, 101b: Port 6, 110b: Port 7, 111b: Port 8

b. Is the signal GPIO12 from the PCH routed to the signal LAN\_DISABLE\_N on the Intel wired PHY?

i. if YES (default):

Name	Location	Value
LANPHYPC_GP12_SEL	PCHSTRP0[20]	1b

ii. if NO:

Name	Location	Value
LANPHYPC_GP12_SEL	PCHSTRP0[20]	0b



- c. is MACsec Disabled  
i. if YES (default):

Name	Location	Value
MACSEC_DIS	PCHSTRP0[21]	1b

- ii. if NO:

Name	Location	Value
MACSEC_DIS	PCHSTRP0[21]	0b

2. If the target platform IS NOT using Intel integrated wired LAN solution.

Name	Location	Value
MACSEC_DIS	PCHSTRP0[21]	1b
LANPHYPC_GP12_SEL	PCHSTRP0[20]	0b
SML0_EN	PCHSTRP0[8]	0b
GBE_SMBUS_ADDR_EN	PCHSTRP4[8]	0b
PHYCON[1:0]	PCHSTRP4[1:0]	00b
PHY_PCIE_EN	PCHSTRP9[11]	0b
IWL_EN	PCHSTRP15[6]	0b

## A.24.2 Are DMI Lanes reversed on target design?

1. if YES:

Name	Location	Value
DMILR	PCHSTRP9[6]	1b

2. if NO:

Name	Location	Value
DMILR	PCHSTRP9[6]	0b



### A.24.3 How should PCIe\* Lanes 1-4 on the target platform be configured?

1. 1x4: Port 1 (x4), Ports 2-4 (disabled)

Name	Location	Value
PCIEPCS1	PCHSTRP9[1:0]	11b

- a. If 1X4 PCIe lane 1 **is** reversed

Name	Location	Value
PCIELR1	PCHSTRP9[4]	1b

- b. If 1X4 PCIe lane 1 **is not** reversed

Name	Location	Value
PCIELR1	PCHSTRP9[4]	0b

2. 2x2: 2x2 Port 1 (x2), Port 3 (x2), Ports 2, 4 (disabled) (Not for Desktop)

Name	Location	Value
PCIEPCS1	PCHSTRP9[1:0]	10b

3. 1x2, 2x1 Port 1 (x2), Port 2 (disabled), Ports 3, 4 (x1) (Not for Desktop)

Name	Location	Value
PCIEPCS1	PCHSTRP9[1:0]	01b

4. 4x1: Ports 1-4 (x1)

Name	Location	Value
PCIEPCS1	PCHSTRP9[1:0]	00b



#### A.24.4 How should PCIe\* Lanes 5-8 on the target platform be configured?

1. 1x4 – one 4 lane PCIe port

Name	Location	Value
PCIEPCS2	PCHSTRP9[3:2]	11b

- a. If 1X4 PCIe lane 5 **is** reversed

Name	Location	Value
PCIELR2	PCHSTRP9[5]	1b

- b. If 1X4 PCIe lane 5 **is not** reversed

Name	Location	Value
PCIELR2	PCHSTRP9[5]	0b

2. 2x2: Port 5 (x2), Port 7 (x2), Ports 6, 8 (disabled) (Not for Desktop)

Name	Location	Value
PCIEPCS2	PCHSTRP9[3:2]	10b

3. 1x2, 2x1: Port 5 (x2), Port 6 (disabled), Ports 7, 8 (x1) (Not for Desktop)

Name	Location	Value
PCIEPCS2	PCHSTRP9[3:2]	01b

4. 4x1: Ports 5-8 (x1)

Name	Location	Value
PCIEPCS2	PCHSTRP9[3:2]	00b





### A.24.5 Is there a third party device connected to SMLink1 that will gather Thermal Reporting Data on the target platform?

1. If YES,

Name	Location	Value
SM1_EN	PCHSTRP0[9]	1b
SML1I2CA	PCHSTRP11[31:25]	See "PCHSTRP11—Strap 11 Record (Flash Descriptor Records)" [31:25] usage
SML1I2CAEN	PCHSTRP11[24]	1b
SML1GPA	PCHSTRP11[7:1]	See "PCHSTRP11—Strap 11 Record (Flash Descriptor Records)" [7:1] usage
SML1GPEN	PCHSTRP11[0]	1b

a. If thermal data to be collected is PCH only

Name	Location	Value
SMLINK1_THERM_SEL	PCHSTRP15[14]	1b

b. If thermal data is to Processor, and PCH

Name	Location	Value
SMLINK1_THERM_SEL	PCHSTRP15[14]	0b

2. If NO,

Name	Location	Value
SM1_EN	PCHSTRP0[9]	0b
SML1I2CA	PCHSTRP11[31:25]	00h
SML1I2CAEN	PCHSTRP11[24]	0b
SML1GPA	PCHSTRP11[7:1]	00h
SML1GPEN	PCHSTRP11[0]	0b
SMLINK1_THERM_SEL	PCHSTRP15[14]	0b



### A.24.6 What is the size of the boot BIOS block on the target platform? Note: Value must be determined by BIOS developer.

1. If 64 KB,

Name	Location	Value
BBBS	PCHSTRP0[31:29]	000b

2. If 128 KB,

Name	Location	Value
BBBS	PCHSTRP0[31:29]	001b

3. If 256 KB,

Name	Location	Value
BBBS	PCHSTRP0[31:29]	010b

4. If 512KB,

Name	Location	Value
BBBS	PCHSTRP0[31:29]	011b

5. If 1MB,

Name	Location	Value
BBBS	PCHSTRP0[31:29]	100b

### A.24.7 Is there an alert sending device (ASD) on Host SMBus on the target platform? NOTE: this is only valid for Intel® AMT enabled platforms

1. If Yes,

Name	Location	Value
MESMASDA	PCHSTRP2[15:9]	See PCHSTRP2[15:9] usage
MESMASDEN	PCHSTRP2[8]	1b
MESMA2UDID	PCHSTRP7[31:0]	See PCHSTRP7 usage

2. If No,

Name	Location	Value
MESMASDA	PCHSTRP2[15:9]	00h
MESMASDEN	PCHSTRP2[8]	0b
MESMA2UDID	PCHSTRP7[31:0]	00000000h



### A.24.8 Are there multiple processors in the target system?

1. If no,

Name	Location	Value
DMI_REQID_DIS	PCHSTRP0[24]	0b

2. If yes,

Name	Location	Value
DMI_REQID_DIS	PCHSTRP0[24]	1b

### A.24.9 Enable Intel ME Debug Options. Including Logging for Intel MDDD (Intel ME Memory-attached Debug Display Device), Intel MESSDC (ME SMBus Debug Console)? Note: All production systems must have logging disabled.

1. If Yes

Name	Location	Value
ME_DEBUG_EN	PCHSTRP10[24]	1b
MDSMBE_ADD	PCHSTRP10[15:9]	38h
MDSMBE_EN	PCHSTRP10[8]	1b

2. If No, **NOTE:** All production platforms **MUST** disable Options.

Name	Location	Value
ME_DEBUG_EN	PCHSTRP10[24]	0b
MDSMBE_ADD	PCHSTRP10[15:9]	00h
MDSMBE_EN	PCHSTRP10[8]	0b

### A.24.10 What is the desired native functionality of GPIO74?

1. If **SML1Alert#**

Name	Location	Value
PCHHOT#_SML1ALERT#_SEL	PCHSTRP9[22]	0b

2. If **PCHHOT#**,

Name	Location	Value
PCHHOT#_SML1ALERT#_SEL	PCHSTRP9[22]	1b



**A.24.11 Does the platform have a PCI bridge chip that requires a subtractive decode agent? Note: If your platform doesn't support PCI set this to no. If using a Desktop/Server PCH that supports PCI interface and do NOT require an external PCI bridge chip then set this to no.**

1. If Yes

Name	Location	Value
SUB_DECODE_EN	PCHSTRP09[14]	1b

2. If No

Name	Location	Value
SUB_DECODE_EN	PCHSTRP09[14]	0b

§ §

