

# Ibex Peak SPI Programming Guide

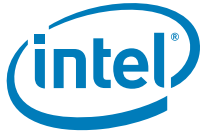
Application Note

---

*January 2009*

*Revision 1.3*

**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. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

The Intel® <product name> 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.

<sup>Δ</sup>Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See [http://www.intel.com/products/processor\\_number](http://www.intel.com/products/processor_number) for details. Over time processor numbers will increment based on changes in clock, speed, cache, FSB, or other features, and increments are not intended to represent proportional or quantitative increases in any particular feature. Current roadmap processor number progression is not necessarily representative of future roadmaps. See [www.intel.com/products/processor\\_number](http://www.intel.com/products/processor_number) for details.

<sup>†</sup> Hyper-Threading Technology requires a computer system with an Intel® Pentium® processor Extreme Edition supporting Hyper-Threading Technology and an HT Technology enabled chipset, BIOS, and an operating system. Performance will vary depending on the specific hardware and software you use. See <<http://www.intel.com/info/hyperthreading>> for information including details on which processors support HT Technology.

Enabling Execute Disable Bit functionality requires a PC with a processor with Execute Disable Bit capability and a supporting operating system. Check with your PC manufacturer on whether your system delivers Execute Disable Bit functionality.

<sup>±</sup>Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and, for some uses, certain platform software enabled for it. Functionality, performance or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all operating systems. Please check with your application vendor.

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 © 2008-2009, Intel Corporation. All Rights Reserved.

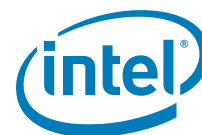


# Contents

<b>1</b>	<b>Introduction</b>	9
1.1	Overview	9
1.2	Terminology	10
1.3	Reference Documents	10
<b>2</b>	<b>PCH SPI Flash Architecture</b>	12
2.1	Non-Descriptor vs. Descriptor Mode	12
2.2	Boot Destination Options	12
2.2.1	Boot Flow for Ibex Peak	12
2.3	Flash Regions	13
2.3.1	Flash Region Sizes	13
2.4	Hardware vs. Software Sequencing	14
<b>3</b>	<b>PCH SPI Flash Compatibility Requirements</b>	15
3.1	Ibex Peak Family SPI Flash Requirements	15
3.1.1	SPI-based BIOS Requirements	15
3.1.2	Integrated LAN Firmware SPI Flash Requirements	15
3.1.2.1	SPI Flash Unlocking Requirements for Integrated LAN	16
3.1.3	Intel® Management Engine (Intel® ME) Firmware SPI Flash Requirements	16
3.1.3.1	SPI Flash Unlocking Requirements for Management Engine	16
3.1.4	JEDEC ID (Opcode 9Fh)	16
3.1.5	Multiple Page Write Usage Model	17
3.1.6	Hardware Sequencing Requirements	17
3.2	Ibex Peak SPI AC Electrical Compatibility Guidelines	18
3.3	SPI Flash DC Electrical compatibility guidelines	20
<b>4</b>	<b>Descriptor Overview</b>	21
4.1	Flash Descriptor Content	25
4.1.1	Descriptor Signature and Map	25
4.1.1.1	FLVALSIG - Flash Valid Signature (Flash Descriptor Records)	25
4.1.1.2	FLMAP0 - Flash Map 0 Register (Flash Descriptor Records)	25
4.1.1.3	FLMAP1—Flash Map 1 Register (Flash Descriptor Records)	26
4.1.1.4	FLMAP2—Flash Map 2 Register (Flash Descriptor Records)	26
4.1.2	Flash Descriptor Component Section	27
4.1.2.1	FLCOMP—Flash Components Record (Flash Descriptor Records)	27
4.1.2.2	FLILL—Flash Invalid Instructions Record (Flash Descriptor Records)	29
4.1.2.3	FLPB—Flash Partition Boundary Record (Flash Descriptor Records)	29
4.1.3	Flash Descriptor Region Section	29
4.1.3.1	FLREG0—Flash Region 0 (Flash Descriptor) Register (Flash Descriptor Records)	30
4.1.3.2	FLREG1—Flash Region 1 (BIOS) Register (Flash Descriptor Records)	30
4.1.3.3	FLREG2—Flash Region 2 (Intel ME) Register (Flash Descriptor Records)	30
4.1.3.4	FLREG3—Flash Region 3 (GbE) Register (Flash Descriptor Records)	31



4.1.3.5	FLREG4—Flash Region 4 (Platform Data) Register (Flash Descriptor Records).....	31
4.1.4	Flash Descriptor Master Section .....	31
4.1.4.1	FLMSTR1—Flash Master 1 (Host CPU/ BIOS) (Flash Descriptor Records).....	32
4.1.4.2	FLMSTR2—Flash Master 2 (Intel® ME) (Flash Descriptor Records).....	32
4.1.4.3	FLMSTR3—Flash Master 3 (GbE) (Flash Descriptor Records).....	33
4.1.5	PCH Softstraps .....	34
4.1.6	Descriptor Upper Map Section.....	34
4.1.6.1	FLUMAP1—Flash Upper Map 1 (Flash Descriptor Records).....	34
4.1.7	Intel® ME Vendor Specific Component Capabilities Table .....	36
4.1.7.1	JID0—JEDEC-ID 0 Register (Flash Descriptor Records).....	36
4.1.7.2	VSCC0—Vendor Specific Component Capabilities 0 (Flash Descriptor Records).....	36
4.1.7.3	JIDn—JEDEC-ID Register n (Flash Descriptor Records).....	39
4.1.7.4	VSCCn—Vendor Specific Component Capabilities n (Flash Descriptor Records).....	40
4.2	OEM Section .....	43
4.3	Region Access Control .....	43
4.3.1	Intel Recommended Permissions for Region Access .....	44
4.3.2	Overriding Region Access .....	45
4.4	Intel® Management Engine (Intel® ME) Vendor-Specific Component Capabilities Table .....	46
4.4.1	How to Set a JEDEC ID Portion of Intel® ME VSCC Table Entry .....	46
4.4.2	How to Set a VSCC Entry in Intel® ME VSCC Table for Ibex Peak Family Platforms .....	46
4.4.3	Example Intel® ME VSCC Table Settings for Intel Ibex Peak Family Systems ..	50
<b>5</b>	<b>Configuring BIOS/GbE for SPI Flash Access .....</b>	<b>52</b>
5.1	Unlocking SPI Flash Device Protection for Ibex Peak Family Platforms .....	52
5.2	Locking SPI Flash via Status Register .....	53
5.3	SPI Protected Range Register Recommendations.....	53
5.4	Software Sequencing Opcode Recommendations.....	53
5.5	Recommendations for Flash Configuration Lockdown and Vendor Component Lock Bits .....	55
5.5.1	Flash Configuration Lockdown .....	55
5.5.2	Vendor Component Lock .....	55
5.6	Host Vendor Specific Component Control Registers (LVSCC and UVSCC) for Ibex Peak Family Systems.....	55
5.7	Example Host VSCC Register Settings for Ibex Peak Family Systems .....	62
<b>6</b>	<b>Flash Image Tool .....</b>	<b>64</b>
6.1	Flash Image Details .....	64
6.1.1	Flash Space Allocation .....	65
6.2	Modifying the Flash Descriptor Region .....	65
6.2.1	Setting the Number and Size of the Flash Components .....	65
6.2.2	Region Access Control .....	68
6.3	PCH Soft Straps.....	69
6.4	Management Engine VSCC Table.....	69
6.4.1	Adding a New Table Entry .....	69
6.4.2	Removing an Existing Table Entry .....	70
<b>7</b>	<b>Flash Programming Tool .....</b>	<b>71</b>



7.1	BIOS Support .....	71
7.2	Fparts.txt File .....	71
7.3	Configuring a Fparts.txt Entry .....	72
7.3.1	Display Name .....	72
7.3.2	Device ID .....	72
7.3.3	Device Size (in Bits) .....	73
7.3.4	Block Erase Size (in Bytes - 256B, 4K, 64K) .....	73
7.3.5	Block Erase Command .....	73
7.3.6	Write Granularity (1 or 64) .....	73
7.3.7	Enable Write Status /Unused .....	73
7.3.8	Chip Erase Command.....	74
<b>8</b>	<b>SPI Flash Programming Procedures .....</b>	<b>75</b>
8.1	Updating BIOS .....	75
8.1.1	Updating BIOS in Descriptor Mode .....	75
8.1.2	Updating BIOS in Non-Descriptor Mode .....	75
<b>9</b>	<b>Intel® Managment Engine Disable for debug/flash burning Purposes .....</b>	<b>77</b>
9.1	Intel® ME Ignition disable .....	77
9.1.1	Erasing/programming Intel® ME IFW region .....	77
9.1.2	Non-descriptor mode .....	77
9.2	Non-Intel ME Ignition disable .....	78
<b>10</b>	<b>Recommendations for SPI Flash Programming in Manufacturing Environments for Ibex Peak .....</b>	<b>79</b>
<b>11</b>	<b>FAQ and Troubleshooting .....</b>	<b>81</b>
11.1	FAQ.....	81
11.2	Troubleshooting .....	83

## Figures

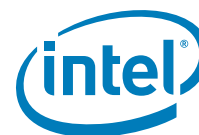
3-1	SPI Timings.....	19
4-1	Flash Descriptor (Ibex Peak A-stepping).....	22
4-2	Flash Descriptor (Ibex Peak B-Stepping and Beyond) .....	22
4-3	Flash Descriptor (Ibex Peak B-Stepping and Beyond) .....	23
6-1	Firmware Image Components .....	64
6-2	Editable Flash Image Region List .....	66
6-3	Descriptor Region – Descriptor Map Options .....	66
6-4	Descriptor Region – Fast Read Support Options.....	67
6-5	Descriptor Region – Component Section Options .....	67
6-6	Descriptor Region – Flash partition Boundary Address and Upper and Lower Flash Erase Size. ....	68
6-7	Descriptor Region – Master Access Section Options.....	68
6-8	Add New VSCC Table Entry .....	69
6-9	Add VSCC Table Entry .....	70
6-10	VSCC Table Entry.....	70
6-11	Remove VSCC Table Entry.....	70

## Tables

1-1	Terminology .....	10
1-2	Reference Documents.....	10
2-1	Region Size vs. Erase Granularity of Flash Components .....	14
3-1	SPI Timings (20 MHz) .....	18



3-2 SPI Timings (33 MHz) .....	18
3-3 SPI Timings (50 MHz) .....	19
4-1 Example Flash Master Register .....	43
4-2 Region Access Control Table Options .....	44
4-3 Recommended Read/Write Settings for Platforms Using Intel® ME Firmware .....	45
4-4 Recommended Read/Write Settings for Platforms Using Intel® ME Firmware (Cont'd) .....	45
4-5 Jidn - JEDEC ID Portion of Intel® ME VSCC Table .....	46
4-6 Vscn - Vendor-Specific Component Capabilities Portion of the Ibex Peak Family Platforms	47
5-1 Recommended opcodes for FPT operation .....	54
5-2 Recommended opcodes for FPT operation .....	54
5-3 LVSCC - Lower Vendor-Specific Component Capabilities Register .....	56
5-4 UVSCC - Upper Vendor-Specific Component Capabilities Register .....	59



## Revision History

---

Document Number	Revision Number	Description	Revision Date
	0.7	<ul style="list-style-type: none"><li>Initial release. Intel Ibex peak only</li></ul>	September 2008
CDI / IBL #: 403598	0.71	<ul style="list-style-type: none"><li>Clean up of SPI flash requirements section</li></ul>	September 2008
	0.72	<ul style="list-style-type: none"><li>Added section on Protected Range Registers. Added Descriptor change for Ibex Peak ES2.</li></ul>	December 2008
	1.0	<ul style="list-style-type: none"><li>Added SPI Softstrap Appendix</li></ul>	January 2009
	1.1	<ul style="list-style-type: none"><li>Added all of SPI Flash Descriptor</li><li>Updated ME VSCC and BIOS VSCC recommended values and descriptions.</li></ul>	February 2009
	1.2	<ul style="list-style-type: none"><li>Removed <b>Intel® ME SMBus General Purpose Address from Softstraps</b></li><li><a href="#">Added Intel® ME Disable sections</a></li></ul>	
	1.3	<ul style="list-style-type: none"><li><a href="#">Changed polarity of PCHSTRP9 bit 7</a></li><li><a href="#">Updated SPI flash Electrical Timing</a></li></ul>	<a href="#">May 2009</a>

§ §







# 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 Ibex Peak Family only.

### [Chapter 2. "PCH SPI Flash Architecture"](#)

Overview of SPI flash, Non-Descriptor vs. Descriptor, Flash Layout, Ibex Peak compatible SPI flash

### [Chapter 3. "PCH SPI Flash Compatibility Requirements"](#)

Overview of compatibility requirements for Ibex Peak products.

### [Chapter 4. "Descriptor Overview"](#)

Overview of the descriptor and Descriptor record definition

### [Chapter 5. "Configuring BIOS/GbE for SPI Flash Access"](#)

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

### [Chapter 6. "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 7. "Flash Programming Tool"](#)

This tool programs the SPI flash device on the Ibex Peak family platforms. This section will talk about requirements needed for FPT to work.

### [Chapter 8. "SPI Flash Programming Procedures"](#)

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

### [Chapter 9. "Intel® Management Engine Disable for debug/flash burning Purposes"](#)

Methods of disabling Intel Management Engine for debug purposes.

### [Chapter 10. "Recommendations for SPI Flash Programming in Manufacturing Environments for Ibex Peak"](#)

Recommendations for manufacturing environments.

### [Chapter 11. "FAQ and Troubleshooting"](#)

Frequently asked questions and Troubleshooting tips.



## 1.2 Terminology

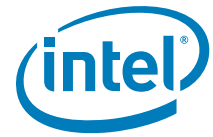
Table 1-1. Terminology

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® AMT	Intel® Active Management Technology
GbE	Intel Integrated 1000/100/10
HDCP	High bandwidth Digital Content Protection
Ibex Peak	Ibex Peak Chipset. Platform Controller Hub
Intel® ME Firmware	Intel firmware that adds Intel® Active Management Technology, Intel® QST, 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)
Intel® QST	Intel® Quiet System Technology - Embedded hardware and firmware solution that allows for algorithmic relationship between system cooling fans and temperature monitors so as to reduce noise without losing thermal efficiency
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

## 1.3 Reference Documents

Table 1-2. Reference Documents

Document	Document No./Location
<i>Intel Ibex Peak 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>I</u> ntel® <u>M</u> E kit from VIP/ARMS. The Kit MUST match the platform you intend to use the flash tools for.



Document	Document No./Location
<i>Intel® Flash Programming Tool (FPT)</i>	\\System Tools\\Flash Programming Tool of latest <u>Intel® 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) and up to two active low chip selects (CSX#) on Ibex Peak.

Ibex Peak can support SPI flash devices up to 16 Mbytes per chip select. Ibex Peak can support frequencies of both 20 MHz and 33 MHz, in future steppings it will support 50 MHz.

### 2.1 Non-Descriptor vs. Descriptor Mode

SPI Flash on Ibex Peak has two operational modes: descriptor and non-descriptor.  
**Ibex Peak supports descriptor mode only.**

Non-descriptor mode is not supported in due to all Ibex Peak platforms requiring Intel ME FW.

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. HDCP will be integrated into the chipset or add in card (not on flash) in all other instances. 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 I/O Controller Hub Family External Design Specification (EDS)* for Ibex Peak for more detailed information.

### 2.2 Boot Destination Options

#### 2.2.1 Boot Flow for Ibex Peak

When booting from Global Reset the PCH SPI controller will look for a descriptor signature on the SPI flash device on Chip Select 0 at address 0x10 (ES2 or later) or 0x0 (ES1). The descriptor fetch is triggered whichever comes first, the assertion of MEPWROK or deassertion of LAN\_RST#. 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. 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 Ibex Peak 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.

## 2.3 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 0x0 or 0x10). Descriptor and ME regions are required for all Ibex Peak 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.

### 2.3.1 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.



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

## 2.4 Hardware vs. Software Sequencing

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.

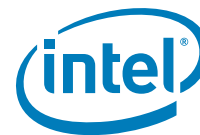
FPT as well as some BIOS implementation use software sequencing.

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 *Ibex Peak Family External Design Specification (EDS)* for more details.

§



## 3 PCH SPI Flash Compatibility Requirements

---

### 3.1 Ibex Peak Family SPI Flash Requirements

Ibex Peak 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 all Ibex Peak based platforms!**

#### 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 [3.1.6 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 JEDEC ID \(Opcode 9Fh\)](#)
- [3.1.5 Multiple Page Write Usage Model](#)
- [3.1.6 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 [3.1.3.1 SPI Flash Unlocking Requirements for Management Engine](#).

#### 3.1.3.1 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 [5.1 Unlocking SPI Flash Device Protection for Ibex Peak Family Platforms](#) and [5.2 Locking SPI Flash via Status Register](#) for more information about flash based write/erase protection.

#### 3.1.4 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).





### 3.1.5 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 AMT firmware multiple page write usage models apply to sequential and non-sequential data writes.

### 3.1.6 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	4 Kbyte erase
Chip Erase	C7h and/or 60	
JEDEC ID	9Fh	<a href="#">See Section 3.1.4 for more information</a>



## 3.2 Ibex Peak 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 Ibex Peak 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 Ibex Peak is 31.25 MHz
2. [Measurement point for low time and high time is taken at .5\(VccME3\\_3\)](#)



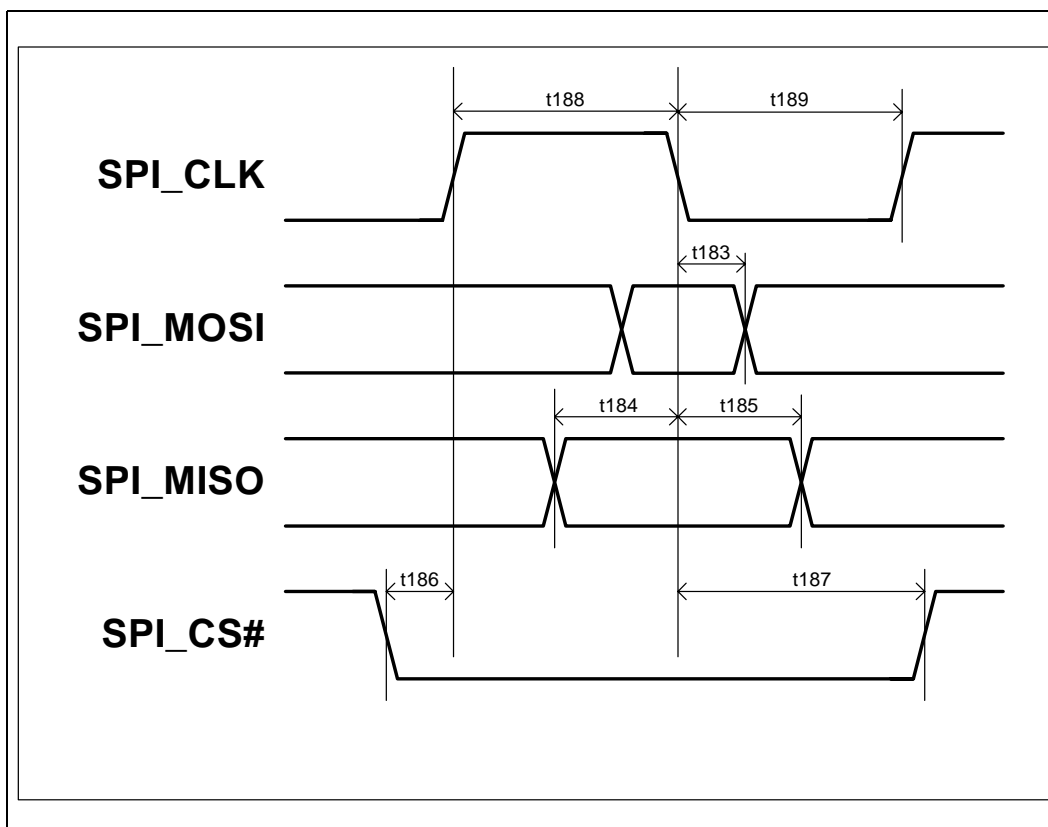
Table 3-3. SPI Timings (50 MHz)

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	7.1	-	ns	2,3
t189c	SPI_CLK Low time	11.17	-	ns	2,3

**Notes:**

1. Typical clock frequency driven by Ibex Peak is 50 MHz. This frequency is not available for ES1 samples.
2. When using 50 MHz mode ensure target flash component can meet t188c and t189c specifications.
3. Measurement point for low time and high time is taken at .5(VccME3 3)

Figure 3-1. **SPI Timings**



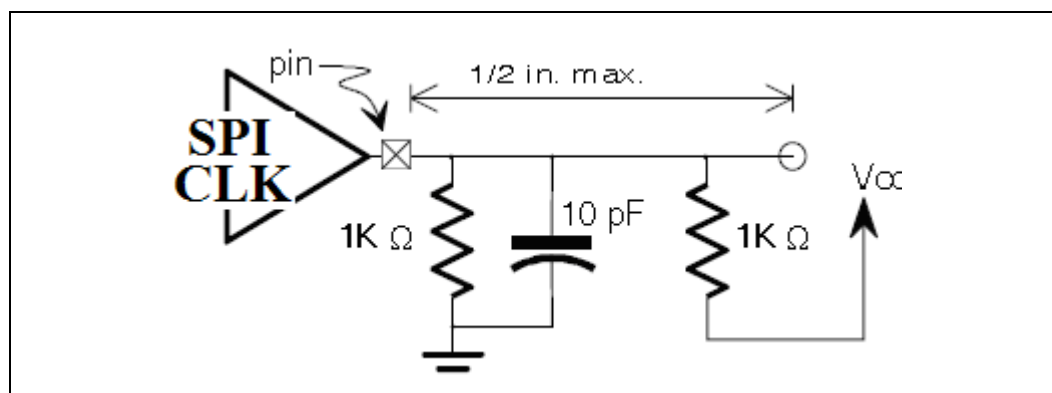
### 3.3 SPI Flash DC Electrical compatibility guidelines

Parameter	Min	Max	Units	Note
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

**Notes:**

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

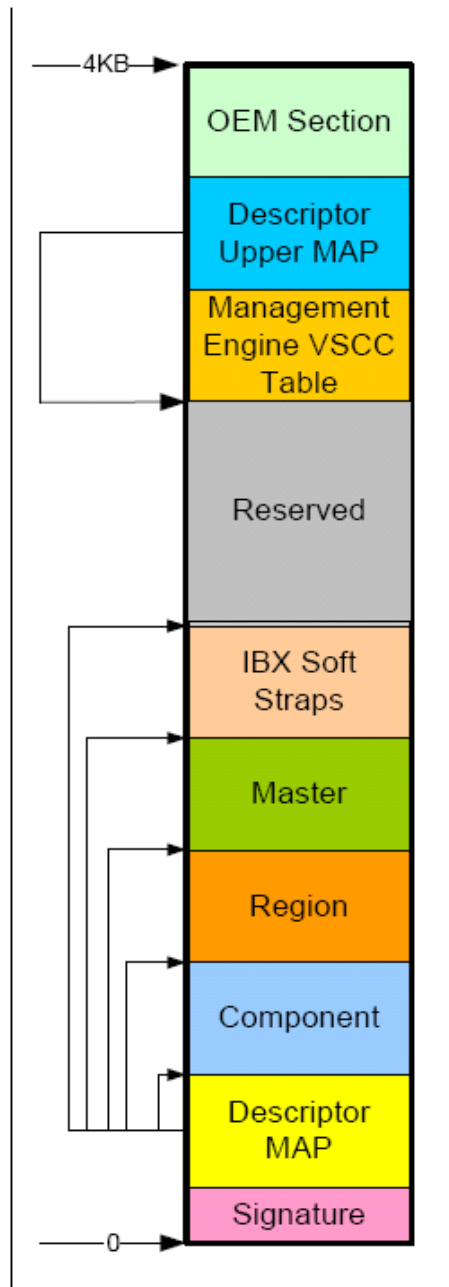
---

The Flash Descriptor is a data structure that is programmed on the SPI flash part on Ibex Peak 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.

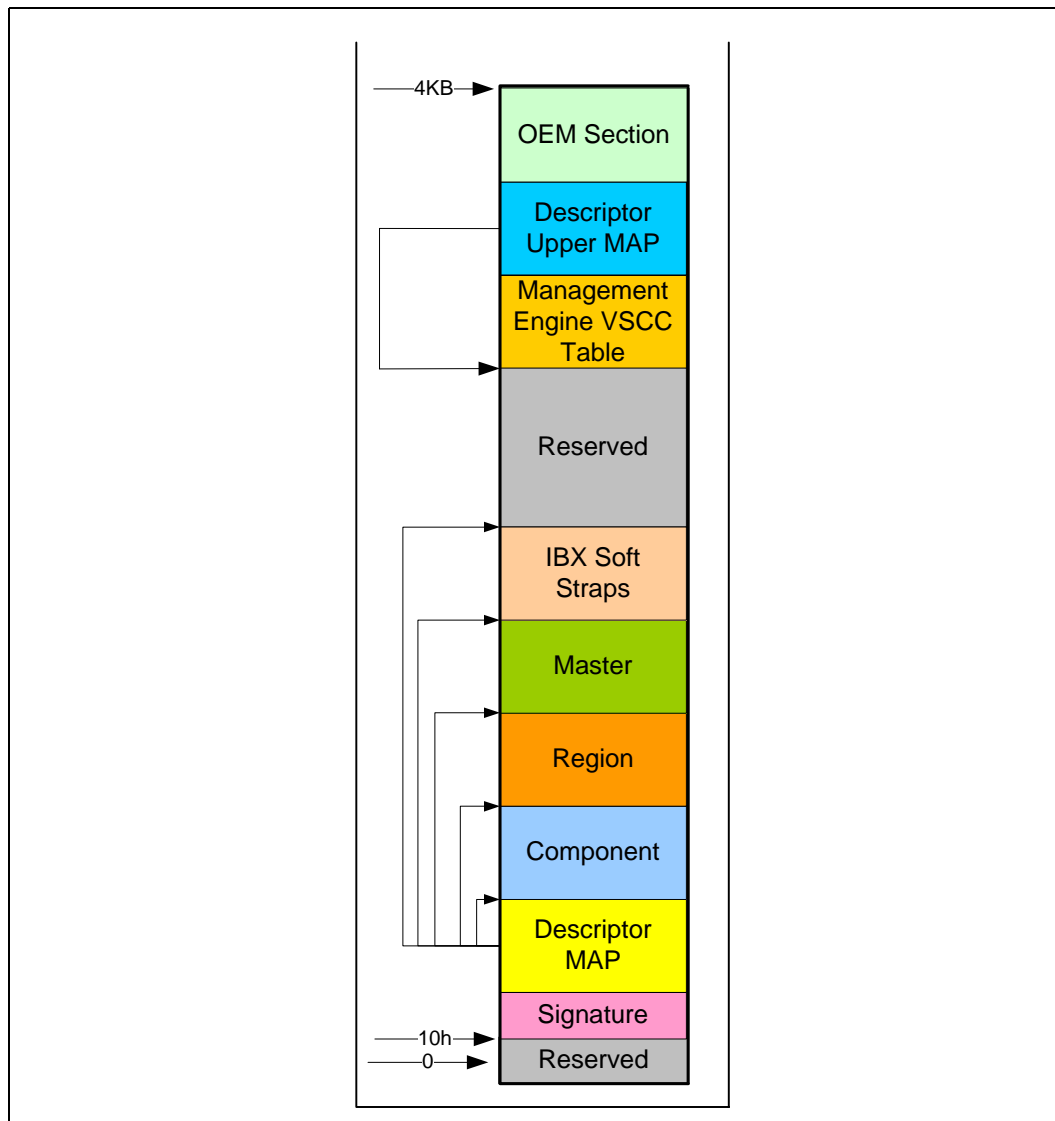
The descriptor has 9 basic parts:

**Figure 4-1. Flash Descriptor (Ibex Peak A-stepping)**



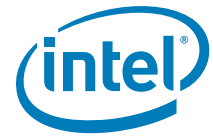
**Figure 4-2. Flash Descriptor (Ibex Peak B-Stepping and Beyond)**

Figure 4-3. Flash Descriptor (Ibex Peak B-Stepping and Beyond)



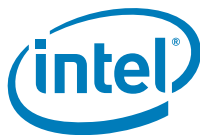
- The Flash signature at the bottom of the flash (offset 0) must be 0FF0A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the lower five descriptor sections as well as the size of each.
- The Component section has information about the SPI flash part(s) the system. It includes 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. This table is NOT used by Intel® ME Ignition FW only. BIOS and GbE write and erase capabilities depend on LVSCC and UVSCC 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 *Intel Ibex Peak 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 (B-step)

Memory Address: FDBAR + 000h (A-step)

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 FLMAP0 - Flash Map 0 Register (Flash Descriptor Records)

Memory Address: FDBAR + 014h (B-step)

Memory Address: FDBAR + 004h (A-step)

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. <b>Note:</b> 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. <b>Note:</b> For B Step set this field to 02h. This will define FCBA as 20h <b>Note:</b> For A Step set this field to 01h. This will define FCBA as 10h



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

Memory Address: FDBAR + 018h (B-step)  
Memory Address: FDBAR + 008h (A-step)

Size: 32 bits

Recommended Value: 10100206h

Bits	Description
31:24	<b>PCH Strap Length (ISL)</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.  <b>Note:</b> This field <b>MUST</b> be set to 10h
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.  <b>Note:</b> 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.  <b>Note:</b> 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.  <b>Note:</b> Set this field to 06h. This will define FMBA as 60h

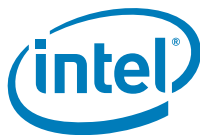
#### 4.1.1.4 FLMAP2—Flash Map 2 Register (Flash Descriptor Records)

Memory Address: FDBAR + 01Ch (B-step)  
Memory Address: FDBAR + 00Ch (A-step)

Size: 32 bits

Recommended Value: 00000020h

Bits	Description
31:16	Reserved
15:08	<b>PROC Strap Length (PSL)</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.  <b>Note:</b> Set this field to 0h
7:0	<b>Flash Processor Strap Base Address (FMSBA)</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.  <b>Note:</b> Set this field to 20h. This will define FMSBA as 200h



## 4.1.2 Flash Descriptor Component Section

The following section of the Flash Descriptor is used to identify the different SPI Flash Components and their capabilities.

### 4.1.2.1 FLCOMP—Flash Components Record (Flash Descriptor Records)

Memory Address: FCBA + 000h

Size:

32 bits

Bits	Description
31:30	Reserved
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> 1. If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components. 2. If setting to 50 MHz, ensure flash meets timing requirements defined in <a href="#">Table 3-3</a>
26:24	<b>Write and Erase Clock Frequency.</b> 000 = 20 MHz 001 = 33 MHz 100 = 50 MHz <b>Notes:</b> All other Settings = Reserved  <b>Notes:</b> 1. If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components. 2. If setting to 50 MHz, ensure flash meets timing requirements defined in <a href="#">Table 3-3</a>
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> 1. If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components. 2. If setting to 50 MHz, ensure flash meets timing requirements defined in <a href="#">Table 3-3</a>



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> If more than one Flash component exists, this field must be set to the lowest common frequency of the different Flash components.</p>
16:6	Reserved
5:3	<p><b>Component 2 Density.</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 are unused.</p> <p>000 = 512 KB            001 = 1 MB            010 = 2 MB            011 = 4 MB            100 = 8 MB            101 = 16 MB            111 = Reserved</p>
2:0	<p><b>Component 1 Density.</b> This field identifies the size of the 1st or only Flash component connected directly to the PCH.</p> <p>000 = 512 KB            001 = 1 MB            010 = 2 MB            011 = 4 MB            100 = 8 MB            101 = 16 MB            111 = Reserved</p> <p><b>Note:</b> If using a flash part smaller than 512 KB, use the 512 KB setting.</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.

#### 4.1.2.3 FLPB—Flash Partition Boundary Record (Flash Descriptor Records)

Memory Address: FCBA + 008h

Size: 32 bits

Bits	Description
31:13	Reserved
12:0	<b>Flash Partition Boundary Address (FPBA).</b> This register specifies Flash Boundary Address bits[24:12] that logically divides the flash space into two partitions, a lower and an upper partition. The lower and upper partitions can support SPI flash parts with different attributes between partitions that are defined in the LVSCC and UVSCC.  <b>Notes:</b> <ol style="list-style-type: none"><li>1. All flash space in each partition must have the same in the VSCC attributes, even if it spans between different flash parts.</li><li>2. If this field is set to all 0s, then there is only one partition, the upper partition, and the entire address space has uniform erasable sector sizes, write granularity, and write state required settings. The FPBA must reside on an erasable sector boundary. If set to all zeros, then only UVSCC register value is used (with the exception of the VCL bit).</li></ol>

### 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 FFFh and the Region Limit to 000h 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
31:29	Reserved
28:16	<b>Region Limit.</b> This specifies bits 24:12 of the ending address for this Region. <b>Note:</b> Set this field to 0b. This defines the ending address of descriptor as being FFFh. <b>Note:</b> <u>Region limit address Bits[11:0] are assumed to be FFFh</u>
15:13	Reserved
12:0	<b>Region Base.</b> This specifies address bits 24:12 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
31:29	Reserved
28:16	<b>Region Limit.</b> This specifies bits 24:12 of the ending address for this Region. <b>Notes:</b> <ol style="list-style-type: none"> <li>1. Must be set to 0000h if BIOS region is unused (on Firmware hub)</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. <u>Region limit address Bits[11:0] are assumed to be FFFh</u></li> </ol>
15:13	Reserved
12:0	<b>Region Base.</b> This specifies address bits 24:12 for the Region Base. <b>Note:</b> If the BIOS region is not used, the Region Base must be programmed to 1FFFh

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

Memory Address: FRBA + 008h

Size: 32 bits

Bits	Description
31:29	Reserved
28:16	<b>Region Limit.</b> This specifies bits 24:12 of the ending address for this Region. <b>Note:</b> Ensure size is a correct reflection of actual Intel ME firmware size that will be used in the platform <b>Note:</b> <u>Region limit address Bits[11:0] are assumed to be FFFh</u>
15:13	Reserved
12:0	<b>Region Base.</b> This specifies address bits 24:12 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
31:29	Reserved
28:16	<b>Region Limit.</b> This specifies bits 24:12 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. <a href="#">Region limit address Bits[11:0] are assumed to be FFFh</a>
15:13	Reserved
12:0	<b>Region Base.</b> This specifies address bits 24:12 for the Region Base. <b>Note:</b> If the GbE region is not used, the Region Base must be programmed to 1FFFh

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

Memory Address: FRBA + 010h

Size: 32 bits

Bits	Description
31:29	Reserved
28:16	<b>Region Limit.</b> This specifies bits 24:12 of the ending address for this Region. <b>Notes:</b> 1. If PDR Region is not used, the Region Limit must be programmed to 0000h 2. Ensure BIOS region size is a correct reflection of actual BIOS image that will be used in the platform 3. <a href="#">Region limit address Bits[11:0] are assumed to be FFFh</a>
15:13	Reserved
12:0	<b>Region Base.</b> This specifies address bits 24:12 for the Region Base. <b>Note:</b> If the Platform Data region is not used, the Region Base must be programmed to 1FFFh

### 4.1.4 Flash Descriptor Master Section

See 4.3 [Region Access Control](#) for more detail on how to properly set this section.

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

Memory Address: FMBA + 000h

Size: 32 bits





Bits	Description
31:29	Reserved  <b>Note:</b> This field should be set to 111b if all regions of flash are open to all masters in pre-production environments. See <a href="#">4.3.1 Intel Recommended Permissions for Region Access</a> for more details.
28	<b>Platform Data Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
27	<b>GbE Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
26	<b>Intel ME Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
25	<b>Host CPU/BIOS Master Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses. Bit 25 is a don't care as the primary master always has read/write permissions to it's primary region
24	<b>Flash Descriptor Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
23:21	Reserved  <b>Note:</b> This field should be set to 111b if all regions of flash are open to all masters in pre-production environments. See <a href="#">4.3.1 Intel Recommended Permissions for Region Access</a> for more details.
20	<b>Platform Data Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
19	<b>GbE Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
18	<b>Intel ME Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
17	<b>Host CPU/BIOS Master Region Read Access.</b> 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 has read/write permissions to it's primary region
16	<b>Flash Descriptor Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
15:0	<b>Requester ID.</b> This is the Requester ID of the Host processor. This must be set to 0000h.

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

Memory Address: FMBA + 004h  
Size: 32 bits

Bits	Description
31:29	Reserved  <b>Note:</b> This field should be set to 111b if all regions of flash are open to all masters in pre-production environments. See 4.3.1 <a href="#">Intel Recommended Permissions for Region Access</a> for more details.
28	<b>Platform Data Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
27	<b>GbE Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
26	<b>Intel ME Master Region Write Access.</b> 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 permissions to it's primary region
25	<b>Host CPU/BIOS Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
24	<b>Flash Descriptor Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
23:21	Reserved  <b>Note:</b> This field should be set to 111b if all regions of flash are open to all masters in pre-production environments. See 4.3.1 <a href="#">Intel Recommended Permissions for Region Access</a> for more details.
20	<b>Platform Data Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
19	<b>GbE Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
18	<b>Intel ME Master Region Read Access.</b> 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 has read/write permissions to it's primary region
17	<b>Host CPU/BIOS Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
16	<b>Flash Descriptor Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
15:0	<b>Requester ID.</b> This is the Requester ID of the Intel Management Engine. This must be set to 0000h.

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

Memory Address: FMBA + 008h  
Size: 32 bits



Bits	Description
31:29	Reserved <b>Note:</b> This field should be set to 111b if all regions of flash are open to all masters in pre-production environments. See <a href="#">4.3.1 Intel Recommended Permissions for Region Access</a> for more details.
28	<b>Platform Data Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
27	<b>GbE Master Region Write Access.</b> 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 permissions to its primary region
26	<b>Intel ME Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
25	<b>Host CPU/BIOS Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
24	<b>Flash Descriptor Region Write Access.</b> If the bit is set, this master can erase and write that particular region through register accesses.
23:21	Reserved <b>Note:</b> This field should be set to 111b if all regions of flash are open to all masters in pre-production environments. See <a href="#">4.3.1 Intel Recommended Permissions for Region Access</a> for more details.
20	<b>Platform Data Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
19	<b>GbE Master Region Read Access.</b> 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 has read/write permissions to its primary region
18	<b>Intel ME Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
17	<b>Host CPU/BIOS Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
16	<b>Flash Descriptor Region Read Access.</b> If the bit is set, this master can read that particular region through register accesses.
15:0	<b>Requester ID.</b> This is the Requester ID of the GbE. This must be set to 0118h.

## 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 [24:12] and bits [3:0] are 0. <b>NOTE:</b> VTBA should be above the offset for PROCSTRP0 and below FLUMAP1. It is recommended that this address is set based on the anticipated maximum number of different flash parts entries.



## 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, Intel® Quiet System Technology. BIOS will still need to set up the proper VSCC registers for BIOS and Integrated Gigabit Ethernet usage.

Each VSCC table entry is composed of two 32 bit fields: JEDEC ID and the corresponding VSCC 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:**

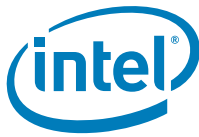
In this table “Lower” applies to characteristics of all flash space below the Flash Partition Boundary Address (FPBA). “Upper” applies to characteristics of all flash space above the FPBA.

Bits	Description
31:24	<b>Lower Erase Opcode (LEO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in LBES.
23:21	Reserved

Bits	Description
20	<p><b>Lower Write Enable on Write Status (LWEWS).</b></p> <p>'0' = 50h will be the opcode used to unlock the status register on SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register on SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 19 (<b>LWEWS</b>) and/or bit 20 (<b>LWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 19 (<b>LWSR</b>) and 20 (<b>LWEWS</b>) 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.</li> <li>4.If bit 19 (<b>LWSR</b>) is set to 1b and bit 20 (<b>LWEWS</b>) 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.</li> </ol>
19	<p><b>Lower Write Status Required (LWSR).</b></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 performed by Intel ME to the SPI flash.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 19 (<b>LWEWS</b>) and/or bit 20 (<b>LWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 19 (<b>LWSR</b>) and 20 (<b>LWEWS</b>) 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.</li> <li>4.If bit 19 (<b>LWSR</b>) is set to 1b and bit 20 (<b>LWEWS</b>) 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.</li> </ol>
18	<p><b>Lower Write Granularity (LWG).</b></p> <p>0 = 1 Byte</p> <p>1 = 64 Byte</p>



Bits	Description
17:16	<b>Lower Block/Sector Erase Size (LBES)</b> . This field identifies the erasable sector size for all Flash space below the flash partition boundary address. Valid Bit Settings: 00 = 256 Byte 01 = 4 KB 10 = 8 KB 11 = 64 KB
15:8	<b>Upper Erase Opcode (UEO)</b> . This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in LBES.
7:5	Reserved
4	<b>Upper Write Enable on Write Status (UWEWS)</b> . '0' = 50h will be the opcode used to unlock the status register on SPI flash if <b>UWSR</b> (bit 3) is set to 1b. '1' = 06h will be the opcode used to unlock the status register on SPI flash if <b>UWSR</b> (bit 3) is set to 1b. <b>NOTES:</b> <ol style="list-style-type: none"> <li>1.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) 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.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) 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.</li> </ol>



Bits	Description
3	<b>Upper Write Status Required (UWSR).</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>NOTES:</b>  1.Bit 3 ( <b>UWEWS</b> ) and/or bit 4 ( <b>UWSR</b> ) 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 ( <b>UWSR</b> ) and 4 ( <b>UWEWS</b> ) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Intel Management Engine firmware performs.  4.If bit 3 ( <b>UWSR</b> ) is set to 1b and bit 4 ( <b>UWEWS</b> ) 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
2	<b>Upper Write Granularity (UWG).</b> 0 = 1 Byte 1 = 64 Bytes
1:0	<b>Upper Block/Sector Erase Size (UBES).</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

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

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

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

**Note:** In this table “Lower” applies to characteristics of all flash space below the Flash Partition Boundary Address (FPBA). “Upper” applies to characteristics of all flash space above the FPBA.

Bits	Description
31:24	<b>Lower Erase Opcode (LEO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in LBES.
23:21	Reserved
20	<p><b>Lower Write Enable on Write Status (LWEWS).</b></p> <p>‘0’ = 50h will be the opcode used to unlock the status register on SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p>‘1’ = 06h will be the opcode used to unlock the status register on SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 19 (<b>LWEWS</b>) and/or bit 20 (<b>LWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 19 (<b>LWSR</b>) and 20 (<b>LWEWS</b>) 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.</li> <li>4.If bit 19 (<b>LWSR</b>) is set to 1b and bit 20 (<b>LWEWS</b>) 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.</li> </ol>

Bits	Description
19	<p><b>Lower Write Status Required (LWSR).</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.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 19 (<b>LWEWS</b>) and/or bit 20 (<b>LWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 19 (<b>LWSR</b>) and 20 (<b>LWEWS</b>) 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.</li> <li>4.If bit 19 (<b>LWSR</b>) is set to 1b and bit 20 (<b>LWEWS</b>) 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.</li> </ol>
18	<p><b>Lower Write Granularity (LWG).</b>            0 = 1 Byte            1 = 64 Byte</p>
17:16	<p><b>Lower Block/Sector Erase Size (LBES).</b> This field identifies the erasable sector size for all Flash space below the flash partition boundary address.            Valid Bit Settings:            00 = 256 Byte            01 = 4 KB            10 = 8 KB            11 = 64 KB</p>
15:8	<p><b>Upper Erase Opcode (UEO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in LBES.</p>
7:5	Reserved



Bits	Description
4	<p><b>Upper Write Enable on Write Status (UWEWS).</b></p> <p>'0' = 50h will be the opcode used to unlock the status register on SPI flash if <b>UWSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register on SPI flash if <b>UWSR</b> (bit 3) is set to 1b.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) 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.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) 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.</li> </ol>
3	<p><b>Upper Write Status Required (UWSR).</b></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 performed by Intel ME to the SPI flash.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Intel Management Engine firmware performs.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) 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</li> </ol>
2	<p><b>Upper Write Granularity (UWG).</b></p> <p>0 = 1 Byte</p> <p>1 = 64 Bytes</p>



Bits	Description
1:0	<b>Upper Block/Sector Erase Size (UBES)</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

## 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.

Refer to the [FLMSTR1](#), [FLMSTR2](#) and [FLMSTR3](#) sections of *Intel Ibex Peak Family External Design Specification (EDS)* for register information for each master.

**Table 4-1. Example Flash Master Register**

Bits	Description
31:29	Reserved, must be zero.
28	<b>Platform Data Region Write Access:</b> If the bit is set, this master can erase and write that particular region through register accesses.
27	<b>GbE Region Write Access:</b> If the bit is set, this master can erase and write that particular region through register accesses.
26	<b>ME Region Write Access:</b> If the bit is set, this master can erase and write that particular region through register accesses.
25	<b>Host CPU/BIOS Master Region Write Access:</b> If the bit is set, this master can erase and write that particular region through register accesses.
24	<b>Flash Descriptor Region Write Access:</b> If the bit is set, this master can erase and write that particular region through register accesses.
23:21	Reserved, must be zero.



Bits	Description
20	<b>Platform Data Region Read Access:</b> If the bit is set, this master can read that particular region through register accesses.
19	<b>GbE Region Read Access:</b> If the bit is set, this master can read that particular region through register accesses.
18	<b>ME Region Read Access:</b> If the bit is set, this master can read that particular region through register accesses.
17	<b>Host CPU/BIOS Master Region Read Access:</b> If the bit is set, this master can read that particular region through register accesses.
16	<b>Flash Descriptor Region Read Access:</b> If the bit is set, this master can read that particular region through register accesses.
15:0	<b>Requester ID:</b> This field is different for each master: Host CPU/BIOS = 0000h, ME= 0000h, GbE = 0118h .

Table 4-2. 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
<b>BIOS (1)</b>	CPU and BIOS can always read from and write to BIOS region	Read / Write	Read / Write
<b>ME (2)</b>	Read / Write	ME can always read from and write to ME region	Read / Write
<b>GbE (3)</b>	Read / Write	Read / Write	GbE software can always read from and write to GbE region
<b>PDR (4)</b>	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 4-3. Recommended Read/Write Settings for Platforms Using Intel® ME Firmware**

Master Access	Descriptor Region Bit 0	ME Region Bit 2	GbE Region Bit 3	BIOS Region Bit 1	PDR Region Bit 4
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	‡

NOTES:

- ‡ = 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 4-4. 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

NOTES:

- ‡ = 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 GPIO33 low 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 writable/readable.

See [5.3 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. This table is defined in the descriptor and is the responsibility of who puts together the NVM image. LVSCC and/or UVSCC 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

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

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

**Table 4-5. 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 Ixex Peak Family Platforms

Lower VSCC (bits 31:16) needs to be programmed in instances where the Flash Partition Boundary is not 0x0. When using an asymmetric flash component (part with two different sets of attributes based on address) a Flash Partition Boundary will need to be used. This includes if the system is intended to support both symmetric AND

asymmetric SPI flash parts. If all flash parts that will be used on this system are not asymmetric, and if all flash space has all the same attributes (not the same vendor or family), then only UVSCC (bits 15:0) needs to be populated.

It is advised that you program both LVSCC and UVSCC in order to support the widest range of flash components.

Refer to [4.4.3 Example Intel® ME VSCC Table Settings for Ibex Peak Family Systems](#).

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

**Table 4-6. Vsccn – Vendor-Specific Component Capabilities Portion of the Ibex Peak Family Platforms**

Bits	Description
31:24	<b>Lower Erase Opcode (LEO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in LBES.
23:21	Reserved
20	<p><b>Lower Write Enable on Write Status (LWEWS).</b></p> <p>'0' = 50h will be the opcode used to unlock the status register on SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register on SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 19 (<b>LWEWS</b>) and/or bit 20 (<b>LWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 19 (<b>LWSR</b>) and 20 (<b>LWEWS</b>) 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.</li> <li>4.If bit 19 (<b>LWSR</b>) is set to 1b and bit 20 (<b>LWEWS</b>) 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.</li> </ol>





Bits	Description
19	<p><b>Lower Write Status Required (LWSR).</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.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 19 (<b>LWEWS</b>) and/or bit 20 (<b>LWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 19 (<b>LWSR</b>) and 20 (<b>LWEWS</b>) 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.</li> <li>4.If bit 19 (<b>LWSR</b>) is set to 1b and bit 20 (<b>LWEWS</b>) 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.</li> </ol>
18	<p><b>Lower Write Granularity (LWG).</b>            0 = 1 Byte            1 = 64 Byte</p>
17:16	<p><b>Lower Block/Sector Erase Size (LBES).</b> This field identifies the erasable sector size for all Flash space below the flash partition boundary address.            Valid Bit Settings:            00 = 256 Byte            01 = 4 KB            10 = 8 KB            11 = 64 KB</p>
15:8	<p><b>Upper Erase Opcode (UEO).</b> This field must be programmed with the Flash erase instruction opcode that corresponds to the erase size that is in LBES.</p>
7:5	Reserved

Bits	Description
4	<p><b>Upper Write Enable on Write Status (UWEWS).</b></p> <p>'0' = 50h will be the opcode used to unlock the status register on SPI flash if <b>UWSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register on SPI flash if <b>UWSR</b> (bit 3) is set to 1b.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) 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.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) 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.</li> </ol>
3	<p><b>Upper Write Status Required (UWSR).</b></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 performed by Intel ME to the SPI flash.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Intel Management Engine firmware performs.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) 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</li> </ol>
2	<p><b>Upper Write Granularity (UWG).</b></p> <p>0 = 1 Byte</p> <p>1 = 64 Bytes</p>



Bits	Description
1:0	<b>Upper Block/Sector Erase Size (UBES)</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

**Upper and Lower Erase Opcode (LEO/UEO)** and **Upper and Lower Block/Sector Erase Size (LBSES/UBSES)** should be set based on the flash part and the firmware on the platform. For Intel® ME enabled platforms this should be 4 KB.

Either **Upper and Lower Write Status Required (LWSR and UWSR)** or **Upper Write Enable on Write Status (LWEWS and UWEWS)** 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 **LWSR/UWSR** bit to 1b and **LWEWS/UWEWS** to 0b if the Enable Write Status Register opcode (50h) is needed to unlock the status register. Opcodes sequence sent to SPI flash will be 50h 01h 00h.
- Set the **LWEWS/UWEWS** bit AND **LWSR/UWSR** bit to 1b if write enable (06h) will unlock the status register. Opcodes sequence sent to SPI flash will be 06h 01h 00h.
- LWSR/UWSR or LWEWS/UWEWS 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 [5.1 Unlocking SPI Flash Device Protection for Ibex Peak Family Platforms](#) and [5.2 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 23:21 and 7:5** are reserved and should be set to all zeros.

#### 4.4.3 Example Intel® ME VSCC Table Settings for Ibex Peak Family Systems

Below is a table that provides general guidelines for BIOS VSCC settings for different SPI flash devices. These settings are not part recommendations, nor are they an



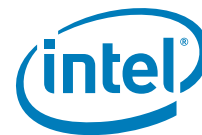
indication these parts are supported on Intel platforms. Flash parts may change opcodes and architectures so please refer to the respective flash datasheet and flash vendor to confirm.

Please refer to [4.4.2 How to Set a VSCC Entry in Intel® ME VSCC Table for Ibex Peak Family Platforms](#) for requirements and how the below values were derived.

Vendor/ Family	Jedec Vendor ID	ME VSCC Table Entry	Upper Flash Erase	Lower Flash Erase	Notes
Atmel* AT25DFxxx or AT26DFxxx1	0x1F	0x20152015, or 0x201D201D	4 KB	4 KB	1, 4, 5
Macronix* MX25L	0xC2	0x20052005	4 KB	4 KB	1, 4
SST* 25VF	0xBF	0x20092009	4 KB	4 KB	1,2,4
Numonyx* / ST Micro* M25PE/PF/PX	0x20	0x20052005	4 KB	4 KB	1,3,4
Winbond* W25X	0xEF	0x20052005	4 KB	4 KB	1,4

NOTES:

1. Upper 2 bytes of ME VSCC Table Entry is not necessary to program if Flash Partition Boundary is zero and flash is not asymmetric. For example: 0x00002005 instead of 0x20052005.
2. SST\* is a registered trademark of Silicon Storage Technology, Inc.
3. Verify the Erase granularity as it may change with revision of flash part. 256 B erase is not supported in any Intel® ME Firmware.
4. Using 0x20012001, 0x20192019 or 0x20112011 will result in slower Intel® ME Firmware performance.
5. Both values are valid.



## 5 Configuring BIOS/GbE for SPI Flash Access

---

### 5.1 Unlocking SPI Flash Device Protection for Ibex Peak 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 Ibex Peak 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 Ibex Peak Family External Design Specification (EDS)* more detailed information.



## 5.2 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.

## 5.3 SPI Protected Range Register Recommendations

The PCH has a mechanism to set up to 5 address ranges from HOST access. These are defined in PRO, 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.

## 5.4 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.

Intel 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 Ibex Peak Family External Design Specification (EDS)*.

**Table 5-1. 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 5-2. Recommended opcodes for FPT operation**

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



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

### 5.5.1 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 Programming Registers](#) section in the *Intel Ibex Peak Family External Design Specification (EDS)*.

### 5.5.2 Vendor Component Lock

It is strongly recommended that BIOS sets the **Vendor Component Lock (VCL)** bits. These bits are located in the BIOS/GbE LVSCC registers. VCL applies the lock to both LVSCC and UVSCC even if LVSCC 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 [LVSCC— Lower Vendor Specific Component Capabilities Register](#) in the *Intel Ibex Peak Family External Design Specification (EDS)* for more information.

## 5.6 Host Vendor Specific Component Control Registers (LVSCC and UVSCC) for Ibex Peak Family Systems

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

All SPI flash address space above or equal to the Flash Partition Boundary Address (FPBA) that is in the Flash Partition Boundary Register (FLPB) utilizes the UVSCC register for flash access. All SPI flash address space below what is defined as the Flash Partition Boundary Address (FPBA) uses the LVSCC register for flash access.

If SPI flash space has only one set of attributes, UVSCC needs to be set. In addition, the Flash Partition Boundary Address in the FLPB in the descriptor must be set to all 0's. The bit definitions for UVSCC and LVSCC are identical, they just apply to different areas of SPI flash space.

Refer to [LVSCC— Lower Vendor Specific Component Capabilities Register](#) and [UVSCC— Upper Vendor Specific Component Capabilities Register](#) in the *Intel Ibex Peak Family External Design Specification (EDS)*.

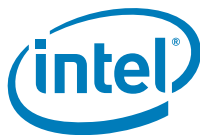




See text below the tables for explanation on how to determine LVSCC and UVSCC register values.

**Table 5-3. LVSCC - Lower Vendor-Specific Component Capabilities Register**

Bit	Description
31:24	Reserved
23	<p><b>Vendor Component Lock (VCL):</b> — RW/L:</p> <p>'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><b>Notes:</b></p> <ol style="list-style-type: none"> <li>This bit applies to both UVSCC and LVSCC registers.</li> <li>All bits locked by <b>(VCL)</b> will remained locked until a global reset.</li> </ol>
22:16	Reserved
15:8	<p><b>Lower Erase Opcode (LEO)</b>— RW:</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 <b>(VCL)</b> bit.</p>
7:5	Reserved



Bit	Description
4	<p><b>Lower Write Enable on Write Status (LWEWS)</b> — RW:</p> <p>'0' = 50h will be the opcode used to unlock the status register on the SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register on the SPI flash if <b>LWSR</b> (bit 3) is set to 1b.</p> <p>This register is locked by the Vendor Component Lock (VCL) bit.</p> <p><b>NOTES:</b></p> <ol style="list-style-type: none"><li>1.Bit 3 (<b>LWEWS</b>) and/or bit 4 (<b>LWSR</b>) should not be set to 1b if there are non volatile bits in the SPI flash device's status register. This may lead to premature flash wear out.</li><li>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 flash part. If the SPI component's status register is non-volatile, then BIOS should issue an atomic software sequence cycle to unlock the flash part.</li><li>3.If both bits 3 (<b>LWSR</b>) and 4 (<b>LWEWS</b>) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Processor or Intel GbE FW performs.</li><li>4.If bit 3 (<b>LWSR</b>) is set to 1b and bit 4 (<b>LWEWS</b>) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.</li></ol>
3	<p><b>Lower Write Status Required (LWSR)</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>NOTES:</b></p> <ol style="list-style-type: none"><li>1.Bit 3 (<b>LWEWS</b>) and/or bit 4 (<b>LWSR</b>) 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.</li><li>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.</li><li>3.If both bits 3 (<b>LWSR</b>) and 4 (<b>LWEWS</b>) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Processor or Intel GbE FW performs.</li><li>4.If bit 3 (<b>LWSR</b>) is set to 1b and bit 4 (<b>LWEWS</b>) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.</li></ol>



Bit	Description
2	<p><b>Lower Write Granularity (LWG) — RW:</b></p> <p>0: 1 Byte 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>Lower Block/Sector Erase Size (LBSES)— RW:</b> This field identifies the erasable sector size for all Flash components.</p> <p>Valid Bit Settings:</p> <p>00: 256 Byte 01: 4 KByte 10: 8 KByte 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>

**Lower Erase Opcode (LEO)** and **Lower Block/Sector Erase Size (LBSES)** should be set based on the flash part and the firmware image on the platform.

Either **Lower Write Status Required (LWSR) OR Lower Write Enable on Write Status (LWEWS)** 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 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.

- Set the **LWSR** bit to 1b and **LWEWS** 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 **LWEWS** bit AND **LWSR** bit to 1b if write enable (06h) will unlock the status register. Opcodes sequence sent to SPI flash will bit 06h 01h 00h.
- LWSR or LWEWS 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 [5.1 Unlocking SPI Flash Device](#)



Protection for Ibex Peak Family Platforms and [5.2 Locking SPI Flash via Status Register](#) for more information.

**Lower Write Granularity (LWG)** 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 [5.5 Recommendations for Flash Configuration Lockdown and Vendor Component Lock Bits](#) for more details.

**Bit ranges 31:24 and 22:16 and 7:5** are reserved and should set to all zeros.

See below table for explanation on how to set bits.

**Table 5-4. UVSCC - Upper Vendor-Specific Component Capabilities Register**

Bit	Description
31:16	Reserved
15:8	<b>Upper Erase Opcode (UEO)</b> — RW: This register is programmed with the Flash erase instruction opcode required by the vendor's Flash component. This register is locked by the Vendor Component Lock ( <b>VCL</b> ) bit.
7:5	Reserved



Bit	Description
4	<p><b>Upper Write Enable on Write to Status (UWEWS)</b> — RW:</p> <p>'0' = 50h will be the opcode used to unlock the status register if <b>UWSR</b> (bit 3) is set to 1b.</p> <p>'1' = 06h will be the opcode used to unlock the status register if <b>UWSR</b> (bit 3) is set to 1b.</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.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) should not be set to 1b if there are non volatile bits in the SPI flash device's status register. This may lead to premature flash wear out.</li> <li>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 flash part. If the SPI component's status register is non-volatile, then BIOS should issue an atomic software sequence cycle to unlock the flash part.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Processor or Intel GbE FW performs.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs.</li> </ol>
3	<p><b>Upper Write Status Required (UWSR)</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>NOTES:</b></p> <ol style="list-style-type: none"> <li>1.Bit 3 (<b>UWEWS</b>) and/or bit 4 (<b>UWSR</b>) 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.</li> <li>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.</li> <li>3.If both bits 3 (<b>UWSR</b>) and 4 (<b>UWEWS</b>) are set to 1b, then sequence of 06h 01h 00h is sent to unlock the flash on EVERY write and erase that Processor or Intel GbE FW performs.</li> <li>4.If bit 3 (<b>UWSR</b>) is set to 1b and bit 4 (<b>UWEWS</b>) is set to 0b then sequence of 50h 01h 00h is sent to unlock the SPI flash on EVERY write and erase that Processor or Intel GbE FW performs</li> </ol>



Bit	Description
2	<p><b>Upper Write Granularity (UWG) — RW:</b> 0: 1 Byte 1: 64 Byte</p> <p>This register is locked by the Vendor Component Lock (<b>VCL</b>) 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>Upper Block/Sector Erase Size (UBES)— RW:</b> This field identifies the erasable sector size for all Flash components.</p> <p>Valid Bit Settings: 00: 256 Byte 01: 4 KByte 10: 8 KByte 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>

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

Either **Upper Write Status Required (UWSR)** or **Upper Write Enable on Write Status (UWEWS)** 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.

- Set the **UWSR** bit to 1b and **UWEWS** 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 **UWEWS** bit AND **UWSR** bit to 1b if write enable (06h) will unlock the status register. Opcodes sequence sent to SPI flash will bit 06h 01h 00h.
- UWSR or UWEWS 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 [5.1 Unlocking SPI Flash Device Protection for Ibex Peak Family Platforms](#) and [5.2 Locking SPI Flash via Status Register](#) for more information.

**Upper Write Granularity (UWG)** 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. **Bit ranges 31:16 and 7:5** are reserved and should set to all zeros.

## 5.7 Example Host VSCC Register Settings for Ibex Peak Family Systems

Below is a table that provides general guidelines for BIOS VSCC settings for different SPI flash devices. These settings are not part recommendations, nor are they an indication these parts are supported on Intel platforms. Flash parts may change opcodes and architectures so please refer to the respective flash datasheet and flash vendor to confirm.

\*\* Please refer to [3 PCH SPI Flash Compatibility Requirements](#) and [5.4 Software Sequencing Opcode Recommendations](#), [5.6 Host Vendor Specific Component Control Registers \(LVSCC and UVSCC\) for Ibex Peak Family Systems](#) for requirements and how the below values were derived.

Vendor/Family	Jedec Vendor ID	UVSCC	LVSCC	Upper Flash Erase	Lower Flash Erase	Notes
Atmel* AT25DFxxx or AT26DFxxx1	0x1F	0x2015 (mbw), 0x2011 (sbw) or 0x201D (mbw), 0x2019 (sbw)	0x802015 (mbw), 0x802011 (sbw), or 0x80201D (mbw), 0x802019 (sbw)	4 KB	4 KB	1,4,5,6,7, 8
Macronix* MX25L	0xC2	0x2005 (mbw) or 0x2001 (sbw)	0x802005 (mbw) or 0x802001 (sbw)	4 KB	4 KB	1,4,5,6,8



Vendor/Family	Jedec Vendor ID	UVSCC	LVSCC	Upper Flash Erase	Lower Flash Erase	Notes
SST* 25VF	0xBF	0x2009	0x802009	4 KB	4 KB	1,2,4,6
Numonyx* / ST Micro* 25PE/PF/PX	0x20	0x2005 (mbw) or 0x2001 (sbw)	0x802005 (mbw) or 0x802001 (sbw)	4 KB	4 KB	1,3,4,5,6,8
Winbond* W25X <u>7</u> <u>W25Q</u>	0xEF	0x2005 (mbw) or 0x2001 (sbw)	0x802005 (mbw) or 0x802001 (sbw)	4 KB	4 KB	1,4,5,6,8

**NOTES:**

1. It is not necessary to program LVSCC if the Flash Partition boundary is 0x0.
2. SST\* is a registered trademark of Silicon Storage Technology, Inc.
3. Verify the Erase granularity as it may change with different revisions of flash part. 256 B erase is not supported in any Intel® ME Firmware.
4. Flash performance may improve with larger erase granularity settings in BIOS only platforms.
5. Use sbw setting if BIOS does not prevent the writing across 256 Byte page boundaries with multiple byte writes.
6. It is strongly recommended to set bit 23 of LVSCC on shipping platforms. See [5.5.2 Vendor Component Lock](#) for more details.
7. When using values of 0x2015, 0x2011, 0x802015, and/or 0x802011 you must unlock the status register. See [5.1 Unlocking SPI Flash Device Protection for Ibex Peak Family Platforms](#) for details
8. mbw = multiple byte write capable. sbw = single byte write capable.

§





## 6 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 Ibex Peak 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.

Note: 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.

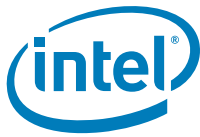
### 6.1 Flash Image Details

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 6-1. Firmware Image Components**

Descriptor	ME FW	GbE	PDR	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.

### 6.1.1 Flash Space Allocation

FIT/Ftoolc 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 is expands to use the remaining space.
4. If there is leftover space and the BIOS region is not implemented, then the GbE region is expands to contain the remaining space.

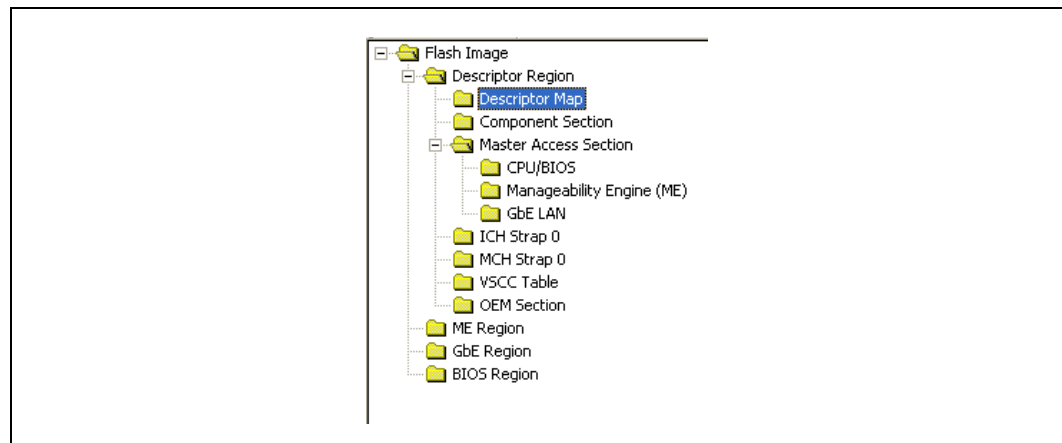
## 6.2 Modifying the Flash Descriptor Region

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.

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

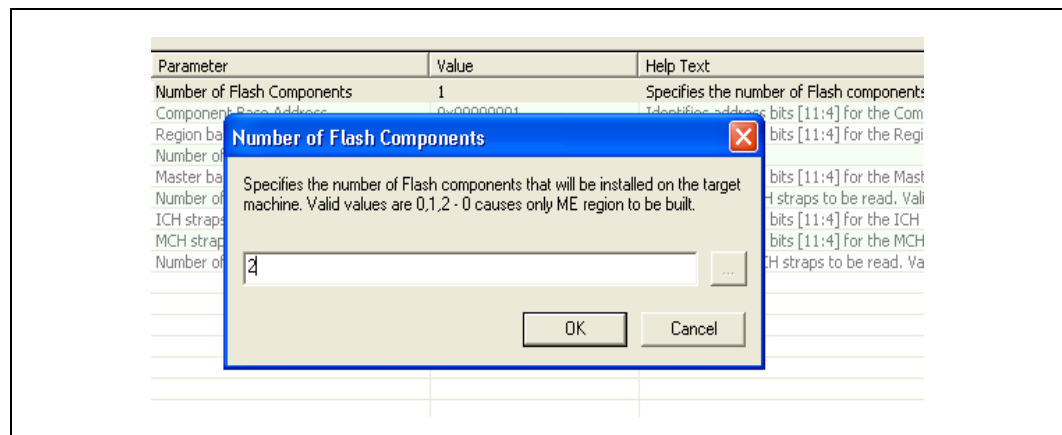
To set the number of flash components, expand the "Descriptor Region" node in the tree on the left side of the main window. Then, select the "Descriptor Map" node (See 3). All of the parameters for the descriptor map section will appear in the list on the right side of the main window.

Figure 6-2. Editable Flash Image Region List



Double-click the list item named “Number of Flash Components” (See [Section 6.3](#)). 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 6-3. 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 Ibex Peak ES1 (A-Step) samples, the fast read clock frequency should be set to 33 MHz, for ES2 (B-Step) samples, this should be set to 50 MHz for Intel AMT enabled enabled platforms.

Figure 6-4. Descriptor Region – Fast Read Support Options

Read ID and Read Status clock frequency	20MHz	If more that one Flash component exists, this field must be th
Write and erase clock frequency	20MHz	If more that one Flash component exists, this field must be th
Fast read clock frequency	33MHz	This field is undefined if the Fast Read Support is set to false.
Fast read support	true	Enables/disables "Fast Read" support.
Read clock frequency	20MHz	Sets the Flash read frequency
Flash component 1 density	512KB	This field identifies the size of the 1st Flash component.
Flash component 2 density	512KB	This field identifies the size of the 2nd Flash component.
Illegal Instruction 0	0	Op-code for an illegal instruction that the Flash Controller sho
Illegal Instruction 1	0	Op-code for an illegal instruction that the Flash Controller sho
Illegal Instruction 2	0	Op-code for an illegal instruction that the Flash Controller sho
Illegal Instruction 3	0	Op-code for an illegal instruction that the Flash Controller sho

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 6-5. Descriptor Region – Component Section Options

Parameter	Value	Help Text
Read ID and Read Status clock frequ...	20MHz	If more that one Flash component exists, this f
Write and erase clock frequency	20MHz	If more that one Flash component exists, this f
Fast read clock frequency	33MHz	This field is undefined if the Fast Read Support
Fast read support	true	Enables/disables "Fast Read" support.
Read clock frequency	20MHz	Sets the Flash read frequency
Flash component 1 density	2MB	This field identifies the size of the 1st Flash cor
Flash component 2 density	2MB	This field identifies the size of the 2nd Flash cor
Illegal Instruction 0	0	Op-code for an illegal instruction that the Flash
Illegal Instruction 1	0	Op-code for an illegal instruction that the Flash
Illegal Instruction 2	0	Op-code for an illegal instruction that the Flash
Illegal Instruction 3	0	Op-code for an illegal instruction that the Flash

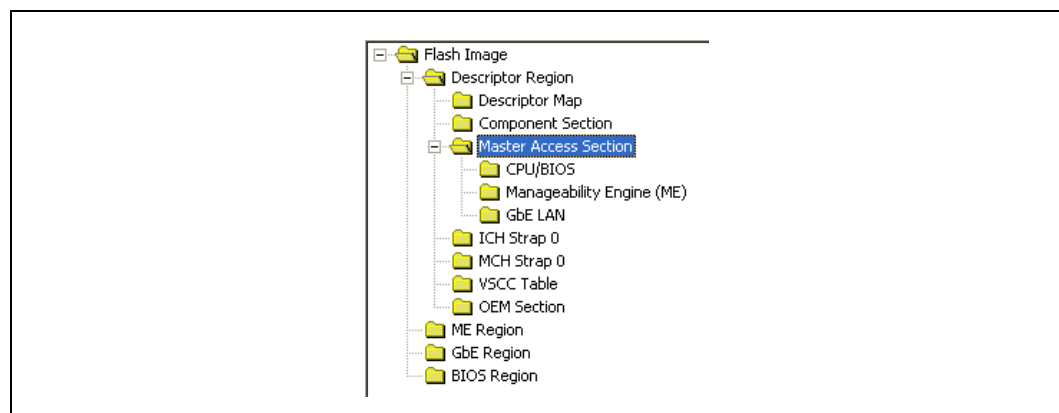
The Upper and Lower Flash Erase sizes and Flash Partition Boundary address is not editable from this view. In order to modify these entries you must enter the Build Settings dialog box. Note that Assymmetric flash parts are no longer supported.

**Figure 6-6. Descriptor Region – Flash partition Boundary Address and Upper and Lower Flash Erase Size.**

#### Region Access Control

Parameter	Value	Help Text
Read ID and Read Status clock frequ...	20MHz	If more that one Flash component exists, this field must be the lowest commo...
Write and erase clock frequency	20MHz	If more that one Flash component exists, this field must be the lowest commo...
Fast read clock frequency	20MHz	This field is undefined if the Fast Read Support is set to false.
Fast read support	false	Enables/disables Fast Read support.
Read clock frequency	20MHz	Sets the Flash read frequency
Flash component 1 density	2MB	This field identifies the size of the 1st Flash component.
Flash component 2 density	2MB	This field identifies the size of the 2nd Flash component.
Illegal Instruction 0	0	Op-code for an illegal instruction that the Flash Controller should protect again...
Illegal Instruction 1	0	Op-code for an illegal instruction that the Flash Controller should protect again...
Illegal Instruction 2	0	Op-code for an illegal instruction that the Flash Controller should protect again...
Illegal Instruction 3	0	Op-code for an illegal instruction that the Flash Controller should protect again...
Upper Flash Erase Size	0x00000	For Asymmetric flash parts, this is the upper of the two erase sizes.
Lower Flash Erase Size	0x00000	For Asymmetric flash parts, this is the lower of the two erase sizes.
Flash Partition Boundary	0x00000000	For Asymmetrix flash parts, this is address bits[24:12] of the boundary that lo...

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”



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

**Figure 6-7. Descriptor Region – Master Access Section Options**

Parameter	Value	Help Text
PCI Bus ID	0	
PCI Device ID	0	
PCI Function ID	0	
Read access	0x00	Each bit corresponds to Regions [7:0]. If the bi
Write access	0x00	Each bit corresponds to Regions [7:0]. If the bi

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

The following is the minimum set of the read/write parameters. This sample will lock down descriptor region with a necessary level of security for Management Engine enabled systems. The settings below will lock the flash region and prevent any future changes to the flash device. This includes any changes made via the fixed offset variable mechanism. If using the fixed offset variable mechanism, manufacturers can alternatively lock the descriptor region during manufacturing. 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 more difficult.

## 6.3 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.

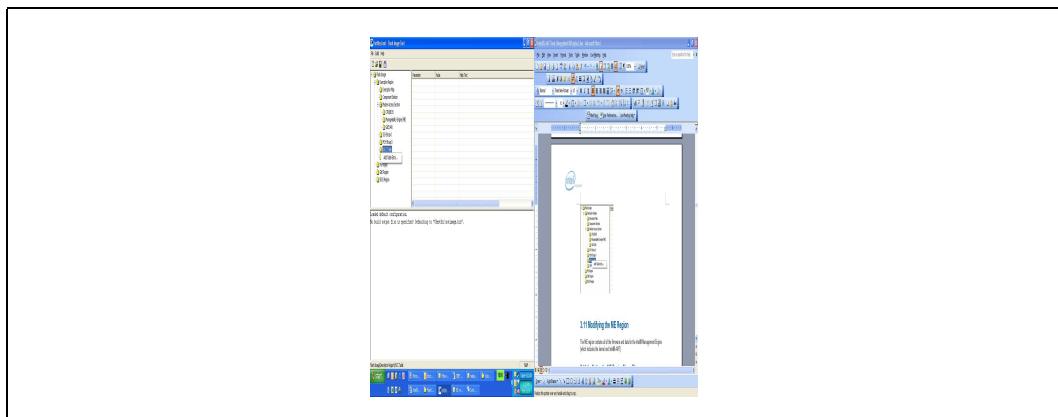
## 6.4 Management Engine VSCC Table

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.

### 6.4.1 Adding a New Table Entry

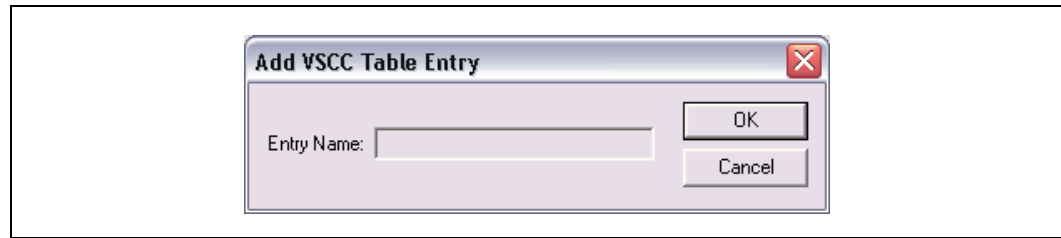
To add a new table, right click on VSCC table and select add a new table entry.

**Figure 6-8. 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 6-9. 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 [7.3.2 Device ID](#) 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.

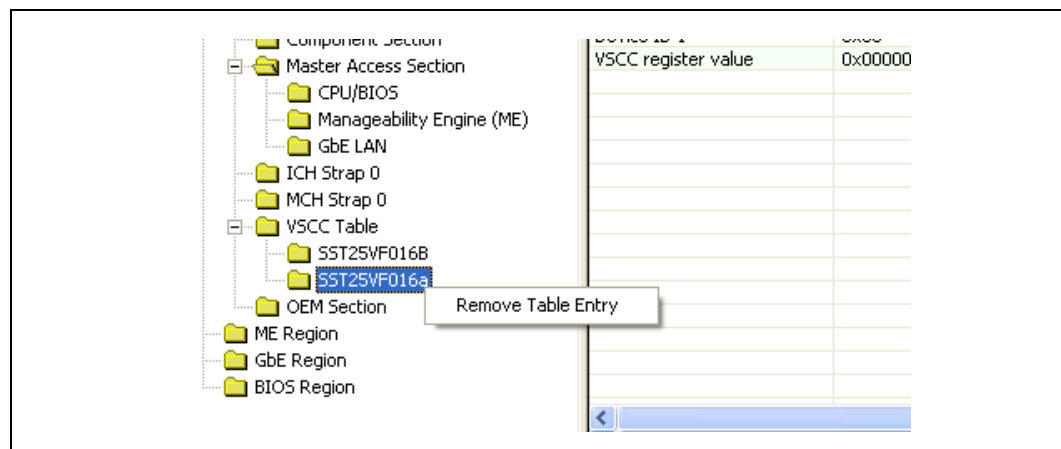
Figure 6-10. VSCC Table Entry

Parameter	Value	Help Text
Vendor ID	0x00	The vendor specific byte of the JEDEC ID.
Device ID 0	0x00	The first device specific byte of the JEDEC ID.
Device ID 1	0x00	The second device specific byte of the JEDEC ID.
VSCC register value	0x00000000	The device specific VSCC register value.

## 6.4.2 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 6-11. Remove VSCC Table Entry





# 7 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:

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

This tool can program an individual region, or the entire flash device.

## 7.1 BIOS Support

FPT requires proper opcodes programmed if the FLOCKDN bit is set. Please refer to [5.4 Software Sequencing Opcode Recommendations](#) and [5.3 SPI Protected Range Register Recommendations](#) for more details.

## 7.2 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) Unused
- 8) Chip Erase Command

## 7.3 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 (50h opcode required to unlock status register)
- 8) Chip Erase Command

### 7.3.1 Display Name

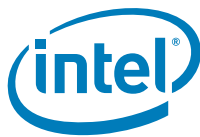
This is a user defined field that FPT will display on the screen to describe that flash part. It is recommended to use the part number to ensure unique and identifiable entry.

### 7.3.2 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.



### 7.3.3 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.

### 7.3.4 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 Ibex Peak 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.

### 7.3.5 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.

### 7.3.6 Write Granularity (1 or 64)

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

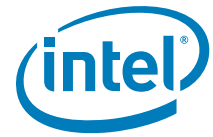
The Ibex Peak only supports 1 or 64 B writes. Flash devices that allow writes more than a single byte at a time usually support up to 256 bytes at a time. Look to see how many bytes the 02h opcode can support.

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.

### 7.3.7 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 [5.1 Unlocking SPI Flash Device Protection for Ibex Peak 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.



### 7.3.8 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

§



## 8 SPI Flash Programming Procedures

---

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

### 8.1 Updating BIOS

If the target system does not have a 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.

#### 8.1.1 Updating BIOS in Descriptor Mode

If updating BIOS in a system where the BIOS region is defined in the descriptor, you can use the following command.

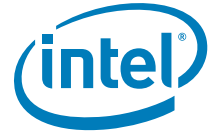
```
C:\ fpt /f <file> /bios
```

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.

#### 8.1.2 Updating BIOS in Non-Descriptor Mode

A BIOS only image without a descriptor is not a valid production option for Ibex peak based platforms. See the *Intel Ibex Peak Family External Design Specification (EDS)* for all the features of descriptor mode.

Unless there is a descriptor, the PCH family parts automatically look for the reset vector on the top of the flash's address space on chip select 0. If the BIOS is not programmed in this location, the system will not boot. Programming can be performed either in system with FPT or with a third party programmer.



**Example:** 1 MByte BIOS image (1MB.bin), 2 MByte SPI flash on platform.

1. 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. Use fpt to program the one MByte binary image at offset 0x100000.

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

3<sup>rd</sup> Party out of system programmer. This is the only option if you do not have a booting system. Begin programming at offset 10 0000h.

§



## 9 Intel® Managment Engine Disable for debug/flash burning Purposes

---

This section is purely for debug purposes. Intel ME firmware is the only supported configuration for Ibex Peak based system.

### 9.1 Intel® ME Ignition disable

It is not necessary to disable Intel ME Ignition firmware for flash burning purposes.

The two ways to disable Intel ME IFW is to either

1. Erase the ME region
2. Use non-descriptor mode

#### 9.1.1 Erasing/programming Intel® ME IFW 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 GPIO33 (Flash descriptor override strap) low 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 [5.3 SPI Protected Range Register Recommendations](#)) for more detail.

**Note:** 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.

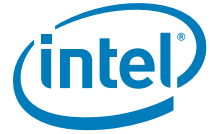
#### 9.1.2 Non-descriptor mode

This can be performed by either erasing the descriptor or by corrupting the Flash valid signature at address 0x0(ES1) or 0x10 (ES2 samples).

If there is no write access to the descriptor, then one must assert GPIO33 (Flash descriptor override strap) low during the rising edge of PWROK.

**Note:** This requires a single flash topology or a topology where BIOS is in FWH or behind an embedded controller. If there is no descriptor the PCH automatically goes to the flash part on SPI chipselect 0 to fetch BIOS code. If you have a 2 flash part system, most likely BIOS is on SPI Chip Select 1. Chip select 1 is not accessible in non-descriptor mode.

**Note:** 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.



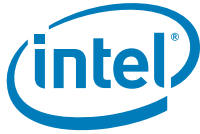
## 9.2 Non-Intel ME Ignition disable

Here are the ways one can disable the Intel® ME for purposes of in system programming the flash. None of these options are necessary for Intel ME Ignition FW.

1. Temporarily disable the Intel® ME through the MEBX. Power off or cold reset. - This option is only applicable to non-Intel ME Ignition firmware.
2. GPIO 33 (Manufacturing mode jumper or Flash descriptor override jumper) asserted low 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. Overwrite the descriptor. WARNING: If using a two flash part platform, this may cause the platform not to boot. The platform will boot in non-descriptor mode, so if the clock configuration is necessary for your platform to boot, this may not be an option.
4. 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 Recommendations for SPI Flash Programming in Manufacturing Environments for Ibex Peak

---

It is recommended that the Intel® ME be disabled when you are programming the ME region. Non Intel Management Engine Ignition 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 non - Intel ME Ignition FW.**

1. Program via In Circuit Test – System is not fully powered here.
2. Program via external flash burning solution.
3. 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.
4. Assert GPIO33 low (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.

With Intel ME Ignition FW, there is no need to disable Intel ME, as Intel ME does not perform writes or erases.

§





# 11 FAQ and Troubleshooting

## 11.1 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. Improperly 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 LVSCC Register and Host UVSCC Register) that is in memory mapped space and one table of VSCC entries (Management Engine VSCC Table) that is 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.

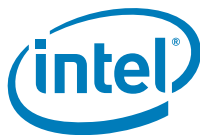
The host based VSCC registers must be programmed for any host based application, or integrated GbE software to access the SPI flash. This will have to be done by your BIOS and NOT by FITc! See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) and/or [5.5 Recommendations 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 Ibex Peak Family External Design Specification (EDS)* See [4.4 Intel® Management Engine \(Intel® ME\) Vendor-Specific Component Capabilities Table](#) and [6.4 Management Engine VSCC Table](#) for more information.

**Q:** How do I find 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® Quiet System 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)



Target	Platform Name In VIP/ARMS	Kit Name
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)
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)

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

**A:** Intel Ibex Peak family 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 Ibex Peak External Design Specification (EDS)* Intel® Management Engine (Intel® ME) Firmware SPI Flash Requirements Intel® Management Engine (Intel® ME) Firmware SPI Flash Requirements Intel® Management Engine (Intel® ME) Firmware SPI Flash Requirements and Support may be added to FPT by referring to [7.3 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 Ibex Peak 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 [6.4 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/v 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 [10 Recommendations for](#)



[SPI Flash Programming in Manufacturing Environments for Ibex Peak](#) for more information.

***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 GPIO33 (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 Ibex peak 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.

## 11.2 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: Timeout waiting for hardware to complete read operation!
      SSFSTS register: 0x00

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

>>> Error: Timeout 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 Ibex Peak 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/ftoolc and set the number of flash components to 1. See [6.2 Modifying the Flash Descriptor Region](#) for more information.

This error can also result if BIOS has not correctly set up software sequencing. See [5.4 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)

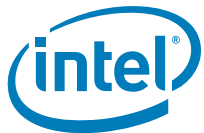
Flash Address: FPSBA + 000h  
Default Flash Address: 100h

Size: 32 bits

Bits	Description	Usage
31	Reserved, set to '0'	
30:29	<p><b>BIOS Boot-Block size (BBBS):</b> Sets BIOS boot-block size</p> <p>00: 64 KB. Invert A16 if Top Swap is enabled (Default) 01: 128 KB. Invert A17 if Top Swap is enabled 10: 256 KB. Invert A18 if Top Swap is enabled 11: 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 Boot block size. This value has to be determined by how BIOS implements Boot-Block.</li> <li>Not implemented in A-stepping of Ibex Peak.</li> </ol>	<p>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> (GPIO[55] 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 Ibex Peak 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 SPI 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 SPI flash then this bit would need to set to '1'.</p> <p>Platforms that have a single processor socket set to '0'</p>



Bits	Description	Usage
23:22	Reserved, set to '0'	
21	<b>Chipset configuration Softstrap 1:</b> Must be set to 1b.	
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>Note:</b> If not using Intel integrated wired LAN or if disabling it, then set to '0'</p> <p><b>Note:</b> 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.</p> <p>Must be programmed to 01b (100 kHz)</p>	100 kHz will be the only supported speed of SMLink0 interface.
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). All other values reserved.</p>	100 kHz will be the only supported speed of Intel ME SMBus interface.
11:10	<p><b>SMLink1 Frequency (SML1FRQ) Frequency:</b> The value of these bits determine the physical bus speed supported by the HW.</p> <p>Must be programmed to 01b (100 kHz). All other values reserved.</p>	100 kHz will be the only supported speed of SMLink1 interface.



Bits	Description	Usage
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.  <b>Note:</b> This is required to be set to '1' for all Mobile platform.</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>This is required for all Mobile platforms.</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>
8	<p><b>SMLink0 Enable (SML0_EN):</b> Configures if SMLink0 segment is enabled  0: Disabled  1: Enabled</p> <p><b>Notes:</b>  1. This bit MUST be set to '1' when utilizing Intel integrated wired LAN.  2. The Intel PHY SMBus controller must be routed to this SMLink 0 Segment.  3. If not using Intel integrated wired LAN solution or if disabling it, then this segment must be set to '0'.</p>	<p>This bit MUST be set to '1' when utilizing Intel integrated wired LAN.</p> <p>The Intel PHY SMBus controller must be routed to this SMLink 0 Segment.</p> <p>If not using Intel integrated wired LAN solution or if disabling it, then this segment must be disabled (set to '0').</p>
7	<p><b>Intel ME SMBus Select (SMB_EN):</b> Configures if the ME SMBus segment is enabled  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>
6:2	Reserved, set to '0'	
1	<b>Chipset configuration Softstrap 2:</b> Must be set to 1b.	
0	Reserved, set to '0'	





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

Flash Address: FPSBA + 004h  
Default Flash Address: 104h

Default Value: 0000000Fh

Size: 32 bits

Bits	Description	Usage
31:4	Reserved, set to '0'	
3:0	<b>Chipset configuration Softstrap 3:</b> Must be set to Fh.	

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

Flash Address: FPSBA + 008h  
Default Flash Address: 108h

Size: 32 bits

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 on Intel® ME Ignition FW.	This address is only used by Intel® ME Ignition 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 Ignition FW	This field should only be set to '1' for testing purposes on platforms that use Intel® ME Ignition FW.
23:16	Reserved, set to '0'	



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'	

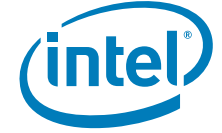
## A.5 PCHSTRP3—Strap 3 Record (Flash Descriptor Records)

Flash Address: FPSBA + 00Ch  
Default Flash Address: 10Ch

Default Value: 00000000h

Size: 32 bits

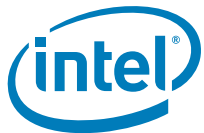
Bits	Description	Usage
31:0	Reserved, set to '0'	



## 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.



Bits	Description	Usage
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'.
7:2	Reserved, set to '0'	
01:00	<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  
Default Flash Address: 118h

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



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

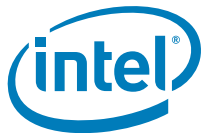
Flash Address: FPSBA + 01Ch      Default Value: 00000000h      Size: 32 bits  
 Default Flash Address: 11Ch

Bits	Description	Usage
31:0	<b>Intel ME SMBus Subsystem Vendor and Device ID (MESMA2UDID):</b> 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: 120hs

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



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

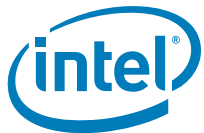
Flash Address: FPSBA + 024h  
Default Flash Address: 124h

Size: 32 bits

Bits	Description	Usage
31:12	Reserved, set to '0'.	
11	<p><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</p> <p><b>Note:</b> This bit must be "1" if using Intel integrated wired LAN solution.</p>	<p>This bit MUST be set to '1' if using Intel integrated wired LAN solution.</p> <p>If not using, or if disabling Intel integrated wired LAN solution then set this to '0'.</p>
10:8	<p><b>Intel PHY PCIe* Port Select (PHY_PCIEPORTSEL):</b> Sets the default PCIe port to use for Intel integrated wired PHY.</p> <p>000: Port 1 001: Port 2 010: Port 3 011: Port 4 100: Port 5 101: Port 6 110: Port 7 111: Port 8</p> <p><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'</p>	<p>This field tells the PCH which PCI Express port an Intel PHY is connected.</p> <p>If PHY_PCIE_EN is = '0', then this field is ignored.</p> <p><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.</p>
7	<u>Reserved, set to '0'.</u>	
6	<p><b>DMI Lane Reversal (DMILR).</b></p> <p>0 = DMI Lanes 0 - 3 are not reversed. 1 = DMI Lanes 0 - 3 are reversed.</p>	<p>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.</p> <p><b>Note:</b> This setting is dependent on the board design. The platform hardware designer must determine if DMI needs lane reversal.</p>



Bits	Description	Usage
5	<p><b>PCIe Lane Reversal 2 (PCIELR2).</b> This bit lane reversal behavior for PCIe Port 5 if configured as a x4 PCIe port.</p> <p>0 = PCIe Lanes 4-7 are not reversed. 1 = PCIe Lanes 4-7 are reversed when Port 5 is configured as a 1x4.</p> <p><b>Note:</b> This field only is in effect if PCIEPCS2 is set to '11'b.</p>	<p>If configuring PCIe* port 5 as a x4 PCIe bus, reversing the lanes of this port is done via this strap.</p> <p>PCI Express* port lane reversal can be done to aid in the laying out of the board.</p> <p><b>Note:</b> This setting is dependent on the board design. The platform hardware designer must determine if this port needs lane reversal.</p>
4	<p><b>PCIe Lane Reversal 1 (PCIELR1).</b> This bit lane reversal behavior for PCIe Port 1 if configured as a x4 PCIe port.</p> <p>0 = PCIe Lanes 0-3 are not reversed. 1 = PCIe Lanes 0-3 are reversed when Port 1 is configured as a 1x4.</p> <p><b>NOTE:</b> This field only is in effect if PCIEPCS1 is set to '11'b.</p>	<p>If configuring PCIe* port 5 as a x4 PCIe bus, reversing the lanes of this port is done via this strap.</p> <p>PCI Express port lane reversal can be done to aid in the laying out of the board.</p> <p><b>Note:</b> This setting is dependent on the board design. The platform hardware designer can determine if this port needs lane reversal</p>
3:2	<p><b>PCI Express Port Configuration Strap 2 (PCIEPCS2).</b> These straps set the default value of the PCI Express port Configuration 2 register covering PCIe ports 5-8.</p> <p>"11": 1x4 Port 5 (x4), Ports 6-8 (disabled) "10": 2x2 Port 5 (x2), Port 7 (x2), Ports 6, 8 (disabled) "01": 1x2, 2x1 Port 5 (x2), Port 6 (disabled), Ports 7, 8 (x1) "00": 4x1 Ports 5-8 (x1)</p> <p><b>NOTE:</b> x2 configurations are not supported on desktop platforms</p>	<p>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'.</p> <p><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.</p>
1:0	<p><b>PCI Express Port Configuration Strap 1 (PCIEPCS1).</b> These straps set the default value of the PCI Express Port Configuration 1 register covering PCIe ports 1-4.</p> <p>"11": 1x4 Port 1 (x4), Ports 2-4 (disabled) "10": 2x2 Port 1 (x2), Port 3 (x2), Ports 2, 4 (disabled) "01": 1x2, 2x1 Port 1 (x2), Port 2 (disabled), Ports 3, 4 (x1) "00": 4x1 Ports 1-4 (x1)</p> <p><b>NOTE:</b> x2 configurations are not supported on desktop platforms</p>	<p>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'.</p> <p><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.</p>



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

Flash Address: FPSBA + 028h      Size: 32 bits  
Default Flash Address: 128h

Bits	Description	Usage
31:22	Reserved, set to '0'	
21	<b>Intel® ME Reset Capture on CL_RST1#: (MER_CL1)</b> 0 = PCH Signal <b>CL_RST1#</b> does NOT assert when Intel ME performs a reset. 1 = PCH Signal <b>CL_RST1#</b> asserts when Intel ME resets.  <b>Notes:</b> 1. This option is applicable on non Intel Management Engine Ignition firmware Platforms with Wireless LAN Clink interface disabled 2. Signal <b>CL_RST1#</b> is only present on mobile PCH	This field requires proper Intel Management Engine Firmware and descriptor. This option is NOT available on Intel ME Ignition firmware.  When this field is set to '1', Intel Management Engine will assert a the <b>CL_RST1#</b> when it resets. When set to '0', Intel ME does not reflect this reset.
20:18	<b>Integrated Clocking Configuration Select (ICC_SEL)</b> Select the clocking parameters that the platform will boot with.  000 - Config '0' 001 - Config '1' 010 - Config '2' 011 - Config '3' 100 - Config '4' 101 - Config '5' 110 - Config '6' 111 - Config '0' (Default)  <b>Note:</b> If changing these softstraps in the manufacturing process it is strongly recommended to set this field to '111'.	This field chooses the set of clock parameters that are used on the target platform. Its is strongly recommended to use set this field to '111' if you will program the value on the manufacturing line.
17	Reserved, set to '0'	
16	<b>Chipset Configuration Softstrap 7</b> Set to '1'	
15:9	<b>Intel ME Memory-attached Debug Display Device Address (MMADDR):</b> SMBUS address used for MDDD status writes. If this field is 00h, the default address, 38h, is used.	This field is only used for testing purposes.





Bits	Description	Usage
8	<b>Intel® ME Memory-attached Debug Display Device Enable (MMDDE):</b> Enable Intel ME MDDD status writes over SMBUS using the address set by MMADDR.	This field is only used for testing purposes.
7:4	Reserved, set to '0'	
3	<b>Virtualization Engine Enable (VE_EN):</b> 0 = Virtualization Engine is disabled 1 = Virtualization Engine is enabled  <b>NOTE:</b> VE is required to be enabled for Braidwood Technology <b>NOTE: VE_EN(PCHSTRP10 bit 3) and VE_EN2 (PCHSTRP14 bit 8)</b> have to be set to the same value in order to properly enable VE to be enabled and disabled.	It is necessary to set this bit to '1' on any platform that will support Braidwood Technology. If set to '1', then ensure that <b>VE_SEL(PCHSTRP10 bit 16)</b> is set to choose to load Braidwood Technology.  If hardware or Intel® ME FW loaded on the platform does not support either Braidwood, then this field must be set to '0'.  This bit must always be set with the same value as <b>VE_EN2 (PCHSTRP14 bit 8)</b> in order to correctly enable (both VE_EN and VE_EN2 set to '1') or disable (both VE_EN and VE_EN2 set to '0') VE.
2	<b>Chipset configuration Softstrap 5:</b> Must be set to 1b.	
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> <u>This field should only be set to '1b' if the Intel ME binary loaded in the platform has a ME ROM Bypass image</u>	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, set to '0'	

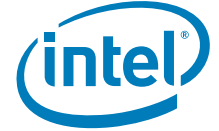


## A.13 PCHSTRP11—Strap 11 Record (Flash Descriptor Records)

Flash Address: FPSBA + 02Ch  
Default Flash Address: 12Ch

Size: 32 bits

Bits	Description	Usage
31:25	<b>SMLink1 I2C* Target Address (SML1I2CA)</b> Defines the 7 bit I2C target address for PCH Thermal Reporting on SMLink1.  <b>Notes:</b> <ol style="list-style-type: none"><li>1. This field is not active unless SML1I2CAEN is set to '1'.</li><li>2. This address MUST be set if there is a device on the SMLink1 segment that will use thermal reporting supplied by PCH.</li><li>3. 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>4. 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	<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  <b>Notes:</b> <ol style="list-style-type: none"><li>1. This bit MUST set to '1' if there is a device on the SMLink1 segment that will use PCH thermal reporting.</li><li>2. 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'	



Bits	Description	Usage
7:1	<b>SMLink1 GP Address (SML1GPA):</b> SMLink1 controller General Purpose Target Address (7:1)  <b>Notes:</b> <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>	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.  A valid address must be: <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>
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> <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 SMBus based PCH thermal reporting.</li> <li>This bit MUST be set to '0' if PCH thermal reporting is not used.</li> </ol>	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.

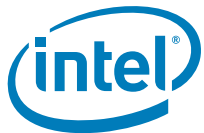
## A.14 PCHSTRP12—Strap 12 Record (Flash Descriptor Records)

Flash Address: FPSBA + 030h  
 Default Flash Address: 130h

Default Value: 00000000h

Size: 32 bits

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



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

Flash Address: FPSBA + 034h  
Default Flash Address: 134h

Default Value: 00000000h

Size: 32 bits

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

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

Flash Address: FPSBA + 038h  
Default Flash Address: 138h

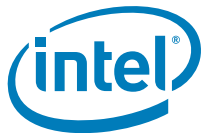
Default Value: 00000000h

Size: 32 bits

Bits	Description	Usage
31:18	Reserved, set to '0'	
17	<p><b>Braidwood NVMHCI Enabled (NVMHCI_EN):</b> Enables Braidwood NVMHCI functionality.</p> <p>'0' : Braidwood NVMHCI is disabled '1': Braidwood NVMHCI is enabled</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"><li>1. <b>VE_EN, VE_EN2 and VE_SEL</b> must all be set to '1' in order for this bit to be applicable on Braidwood Technology capable platforms.</li><li>2. For Ibex peak B0 platforms, <b>VE_BOOT_FLASH</b> must be set to '1'</li><li>3. If both <b>NVMHCI_EN</b> and <b>BW_SSD</b> are set to '0', Braidwood is disabled.</li></ol>	<p>This bit is only applicable in a platform which has the appropriate SKU of Intel Ibex Peak, Intel® Management Engine firmware SKU that supports Braidwood, and Braidwood NAND Module.</p> <p>The following softstraps must be set to '1' to enable Braidwood Technology, <b>VE_SEL (PCHSTRP10 bit 16)</b>, <b>VE_EN (PCHSTRP10 bit 3)</b> and <b>VE_EN2 (PCHSTRP14 bit 8)</b> all have to be set to '1'. Either <b>NVMHCI_EN (PCHSTRP14 bit 17)</b> or <b>BW_SSD (PCHSTRP14 bit 16)</b> or both have to be set to '1' in order to enable Braidwood Technology.</p> <p>For Ibex Peak B0 <b>VE_BOOT_FLASH</b> must be set to '1'</p> <p>If both <b>NVMHCI_EN</b> and <b>BW_SSD</b> are set to '0', Braidwood is disabled.</p>



Bits	Description	Usage
16	<p><b>Braidwood Solid State Device (SSD) enabled (BW_SSD):</b> Enables Braidwood SSD functionality.</p> <p>'0' : Braidwood SSD is disabled '1': Braidwood SSD is enabled</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li><b>VE_EN, VE_EN2 and VE_SEL</b> must all be set to '1' in order for this bit to be applicable on Braidwood Technology capable platforms.</li> <li>For Ibex peak B0 platforms, <b>VE_BOOT_FLASH</b> must be set to '1'</li> <li>If both <b>NVMHCI_EN</b> and <b>BW_SSD</b> are set to '0', Braidwood is disabled.</li> </ol>	<p>This bit is only applicable in a platform which has the appropriate SKU of Intel Ibex Peak, Intel® Management Engine firmware SKU that supports Braidwood, and Braidwood NAND Module.</p> <p>The following softstraps must be set to enable Braidwood Technology, <b>VE_SEL (PCHSTRP10 bit 16)</b>, <b>VE_EN (PCHSTRP10 bit 3)</b> and <b>VE_EN2 (PCHSTRP14 bit 8)</b> all have to be set to '1'. Either <b>NVMHCI_EN (PCHSTRP14 bit 17)</b> or <b>BW_SSD (PCHSTRP14 bit 16)</b> or both have to be set to '1' in order to enable Braidwood Technology.</p> <p>For Ibex Peak B0 <b>VE_BOOT_FLASH</b> must be set to '1'</p> <p>If both <b>NVMHCI_EN</b> and <b>BW_SSD</b> are set to '0', Braidwood is disabled.</p>
15	Reserved, set to '0'	
14	<p><b>PCH VE ROM Bypass (VE_BOOT_FLASH):</b> Allows for indicating ROM Bypass mode of operation for debug purposes</p> <p>0 = Virtualization Engine Firmware will boot from ROM, then Flash 1 = Virtualization Engine Firmware will boot then Flash</p> <p><b>Note:</b> <u>This field should only be set to '1b' if the Intel ME binary loaded in the platform has a VE ROM Bypass image.</u></p>	<p>This bit must be set to '1' on Ibex Peak B0 Stepping to enable and Braidwood Technology.</p> <p><u>In all other cases, this bit must only be set to '1' if there is a VE ROM BYPASS contained in the Intel ME firmware binary.</u></p>
13:9	Reserved, set to '0'	
8	<p><b>Virtualization Engine Enable (VE_EN2):</b></p> <p>0 = Virtualization Engine is disabled 1 = Virtualization Engine is enabled</p> <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>VE is required to be enabled for Braidwood Technology</li> <li><b>VE_EN (PCHSTRP10 bit 3)</b> and <b>VE_EN2 (PCHSTRP14 bit 8)</b> have to be set to the same value in order to properly enable VE to be enabled and disabled.</li> </ol>	<p>It is necessary to set this bit to '1' on any platform that will support Braidwood Technology.</p> <p>If hardware or Intel® ME Firmware loaded on the platform does not support either Braidwood, then this field must be set to '0'.</p> <p>This bit must always be set with the same value as <b>VE_EN2 (PCHSTRP14 bit 8)</b> in order to correctly enable (both <b>VE_EN</b> and <b>VE_EN2</b> set to '1') or disable (both <b>VE_EN</b> and <b>VE_EN2</b> set to '0') VE.</p>



Bits	Description	Usage
7:0	Reserved, set to '0'	

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

Flash Address: FPSBA + 03Ch

Size:

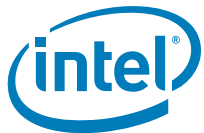
32 bits

Default Flash Address: 13Ch

Recommended Value: 000000358h (Intel LAN Enabled)  
000000318h (Intel LAN Disabled)

Bits	Description	Usage
31:10	Reserved, set to '0'	
9:8	<b>t209 min Timing(t209min)</b> . These two bits adjust PWROK active to PROCPWRGD active minimum timing value as defined below:  00: 100 ms 01: 50 ms 10: 5 ms 11: 1 ms (recommended)	This bit provides flexibility to change the minimum timing between PWROK active and PROCPWRGD active. Do not change this value from 1 ms, unless the target platform requires more time between PWROK active and PROCPWGD.  Recommended value 1 ms. "11b"
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> 1. This must be set to '1' if the platform is using Intel's integrated wired LAN solution. 2. set to '0' if not using Intel integrated wired LAN solution or if disabling it. 3. This field has no effect on A-Step Ibex Peak platforms.	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	Reserved, set to '0'	
4:3	<b>Chipset Configuration Softstrap 6</b> Set to '11b'	
2:0	Reserved, set to '0'	





## A.18 Softstrap Step through

1. General questions help in setting softstraps and certain other descriptor values. [8NET 2.0](#)

- a.
- b. [newfiletmpl.xml](#)
- c. [fitctmpl.xml](#)
- d. [fitc.ini](#)
- e. [vsccommn.bin](#)
- f.

For All configurations the following must be set.

Name	Location	Value
SMB_EN	PCHSTRP0[7]	1b

1. Does the target plaform use the Intel integrated wired LAN solution?

- a. If Yes,

Name	Location	Value
SMLO_EN	PCHSTRP0[8]	1b
GBEPHY_SMBUS_ADDR	PCHSTRP4[23:17]	64h
GBEMAC_SMBUS_ADDR	PCHSTRP4[15:9]	70h
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





- i. 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

- ii. Is the signal GPIO12 from the PCH routed to the signal LAN\_DISABLE\_N on the Intel wired PHY?
  1. If yes:

Name	Location	Value
LANPHYPC_GP12_SEL	PCHSTRP0[20]	1b

2. If no:

Name	Location	Value
LANPHYPC_GP12_SEL	PCHSTRP0[20]	0b

- b. If No, then set all LAN Disabled softstraps

Name	Location	Value
LANPHYPC_GP12_SEL	PCHSTRP0[20]	0b
SMLO_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

**2. Are DMI Lanes reversed on target design?**

a. If Yes:

Name	Location	Value
DMILR	PCHSTRP9[6]	1b

b. If No:

Name	Location	Value
DMILR	PCHSTRP9[6]	0b

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

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

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

i. If 1X4, is PCIe lane 1 reversed?

1. If Reversed:

Name	Location	Value
PCIELR1	PCHSTRP9[4]	1b

2. If NOT Reversed:

Name	Location	Value
PCIELR1	PCHSTRP9[4]	0b



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

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

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

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

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

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

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

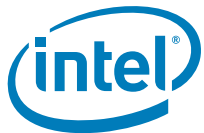
- a. 1x4 – one 4 lane PCIe port

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

- i. Is PCIe lane 5 reversed?

1. If Reversed:

Name	Location	Value
PCIELR2	PCHSTRP9[5]	1b



2. If NOT Reversed:

Name	Location	Value
PCIELR2	PCHSTRP9[5]	0b

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

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

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

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

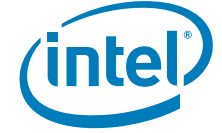
d. 4x1: Ports 5-8 (x1)

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

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

a. If Yes,

Name	Location	Value
SM1_EN	PCHSTRP0[9]	1b
SML112CA	PCHSTRP11[31:25]	See PCHSTRP11[31:25] usage



SML1I2CAEN	PCHSTRP11[24]	1b
SML1GPA	PCHSTRP11[7:1]	See PCHSTRP11[7:1] usage
SML1GPEN	PCHSTRP11[0]	1b

b. 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

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

a. If 64 KB,

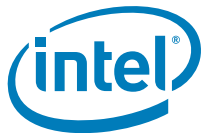
Name	Location	Value
BBBS	PCHSTRP0[30:29]	00b

b. If 128 KB,

Name	Location	Value
BBBS	PCHSTRP0[30:29]	01b

c. If 256 KB,

Name	Location	Value
BBBS	PCHSTRP0[30:29]	10b



**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**

a. 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

b. If No,

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

**8. Is target platform a non-Intel ME Ignition firmware consumer platform? If no, skip to Step 9.**

a. Enable Braidwood?

i. Enable Braidwood then set the following. Note that there are three options for Braidwood functionality.

Name	Location	Value
VE_EN	PCHSTRP10[3]	1b
VE_EN2	PCHSTRP14[8]	1b
VE_BOOT_FLASH	PCHSTRP14[14]	This value should be set to 1b <u>only if Intel ME firmware binary contains a VE ROM image</u>

1. If Braidwood enabled, then enable support for Braidwood NVMHCI (caching) and disable Braidwood SSD functionality.

Name	Location	Value
NVMHCI_EN	PCHSTRP14[17]	1b
BW_SSD_EN	PCHSTRP14[16]	0b



2. Enable support for Braidwood SSD functionality and disable Braidwood NVMHCI.

Name	Location	Value
NVMHCI_EN	PCHSTRP14[17]	0b
BW_SSD_EN	PCHSTRP14[16]	1b

3. Enable both Braidwood SSD Functionality and Braidwood NVMHCI.

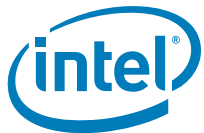
Name	Location	Value
NVMHCI_EN	PCHSTRP14[17]	1b
BW_SSD_EN	PCHSTRP14[16]	1b

- b. If no Braidwood

Name	Location	Value
VE_EN	PCHSTRP10[3]	0b
NVMHCI_EN	PCHSTRP14[17]	0b
BW_SSD	PCHSTRP14[16]	0b
VE_BOOT_FLASH	PCHSTRP14[14]	0b
VE_EN2	PCHSTRP14[8]	0b

9. If target platform is an Intel Management Engine Ignition Firmware platform then set the following:

Name	Location	Value
VE_EN	PCHSTRP10[3]	0b
NVMHCI_EN	PCHSTRP14[17]	0b
BW_SSD	PCHSTRP14[16]	0b



VE_BOOT_FLASH	PCHSTRP14[14]	0b
VE_EN2	PCHSTRP14[8]	0b
VE_BOOT_FLASH	PCHSTRP14[14]	0b

**10. Are there multiple processors in the target system?**

a. If no,

Name	Location	Value
DMI_REQID_DIS	PCHSTRP0[24]	0b

b. If yes,

Name	Location	Value
DMI_REQID_DIS	PCHSTRP0[24]	1b

**11. Enable Logging for Intel MDDD (Intel ME Memory-attached Debug Display Device) and Intel MESSDC (ME SMBus Debug Console) ? Note: All production systems must have logging disabled.**a. If yes. **NOTE:** All pre-production platforms should enable Logging.

Name	Location	Value
MESMI2CEN	PCHSTRP2[24]	1b
MESMI2CA	PCHSTRP2[31:25]	48h
MMDDDE	PCHSTRP10[24]	1b
MMADDR	PCHSTRP10[15:9]	38h

b. If No, **NOTE:** All production platforms **MUST** disable Logging.

Name	Location	Value
MESMI2CEN	PCHSTRP2[24]	0b





MESMI2CA	PCHSTRP2[31:25]	00
MMDDE	PCHSTRP10[24]	0b
MMADDR	PCHSTRP10[15:9]	00h

**12. What is t209 minimum timing for target system. Note: Default value is 1 ms. The Platform Hardware developer will have to determine if the target platform requires more time between PWROK active and PROCPWRGD active.**

a. If 100 ms,

Name	Location	Value
t209min	PCHSTRP0[8:9]	00b

b. If 50 ms,

Name	Location	Value
t209min	PCHSTRP0[8:9]	01b

c. If 5 ms,

Name	Location	Value
t209min	PCHSTRP0[8:9]	10b

d. If 1 ms, **(default)**

Name	Location	Value
t209min	PCHSTRP0[8:9]	11b

§ §

