Intel® Firmware Support Package for Braswell Platform

Integration Guide

August 2015

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Do not finalize a design with this information.

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

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

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

Any software source code reprinted in this document is furnished for informational purposes only and may only be used or copied and no license, express or implied, by estoppel or otherwise, to any of the reprinted source code is granted by this document.

[When the doc contains software source code for a special or limited purpose (such as informational purposes only), use the conditionalized Software Disclaimer tag. Otherwise, use the generic software source code disclaimer from the Legal page and include a copy of the software license or a hyperlink to its permanent location.]

This document contains information on products in the design phase of development.

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

Code Names are only for use by Intel to identify products, platforms, programs, services, etc. ("products") in development by Intel that have not been made commercially available to the public, i.e., announced, launched or shipped. They are never to be used as "commercial" names for products. Also, they are not intended to function as trademarks.

Contents

Revision History

Date	Revision	Description
August 2015	**1.13**	Added new Upd Fields PnpSettings, SdDetectchck, RC version update to 1.1.2.0
July 2015	**1.12**	FSP Version update to 1.1.1.0
July 2015	**1.11**	FSP Version update to 1.1.0.0
June 2015	**1.10**	Add description for MTRR programming in FspInit API, TempRamExit API, and FspSilicionInit API.
June 2015	**1.9**	Added new UPD element for Memory Type Selection and DVFS. And Updated FSP revision to 1.0.9.0
June 2015	**1.8**	Added new upd element PcdTurboMode and updated FSP Version to 1.0.2.2
June 2015	**1.7**	Moved ISPEnable and IsPciDeviceConfig UPDs into SI_INIT_UPD
May 2015	**1.6**	Updated RC Version to 1.0.2.0
May 2015	**1.5**	Added info on UPD consumed by APIs Added FSP_SMBIOS_MEMORY_INFO_HOB
March 2015	**1.4**	Updated FSP revision 1.0.1.0 Updated UPD Signature
March 2015	**1.3**	Updated FSP revision 1.0.0.0
March 2015	**1.2**	Updated FSP revision 0.9.0.0 Updated modified FSP signature details Added new UPD elements recently introduced
January 2015	**1.1**	Updated for FSP revision 0.8.0.0 Added new UPD fields Deleted redundant data which is part of the FSP EAS spec
January 2015	1.0	Initial Release for FSP Revision 0.7.0.1

§

1 Introduction

1.1 Purpose

The purpose of this document is to describe the steps required to integrate the Intel® Firmware Support Package (FSP) for Braswell SOC into a boot loader solution.

1.2 Intended Audience

This document is targeted at all platform and system developers who need to consume FSP binaries in their boot loader solutions. This includes, but is not limited to: system BIOS developers, boot loader developers, system integrators, as well as end users.

1.3 Related Documents

- *Platform Initialization (PI) Specification* located at http://www.uefi.org/specifications
- *UEFI Specification* located at http://www.uefi.org/specifications
- *Intel® Firmware Support Package:* External Architecture Specification v1.1

  http://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/fsp-architecture-spec-v1-1.pdf
- *Binary Configuration Tool for Intel® Firmware Support Package* – available at www.intel.com/fsp
- *Intel Configuration Editor* – Please contact your Intel representative.

1.4 Acronyms and Terminology

Table 1. Acronyms and Terminology

Acronym	Definition
BCT	Binary Configuration Tool
BSP	Boot Strap Processor
BSF	Boot Setting File
BWG	BIOS Writer’s Guide
CRB	Customer Reference Board
FSP	Firmware Support Package
SBSP	System BSP
SMI	System Management Interrupt
SMM	System Management Mode
TSEG	Memory Reserved at the Top of Memory to be used as SMRAM
UPD	Updatable Product Data
VPD	Vital Product Data

§

2 FSP Overview

2.1 Technical Overview

The Intel® Firmware Support Package (FSP) provides chipset and processor initialization in a format that can easily be incorporated into many existing boot loaders.

The FSP will perform the necessary initialization steps as documented in the BWG including initialization of the CPU, memory controller, chipset and certain bus interfaces, if necessary.

FSP is not a stand-alone boot loader; therefore it needs to be integrated into a host boot loader to carry out other boot loader functions, such as: initializing non-Intel components, conducting bus enumeration, and discovering devices in the system and all industry standard initialization.

The FSP binary can be integrated easily into many different boot loaders, such as Coreboot, etc. and also into the embedded OS directly.

Below are some required steps for the integration:

- **Customizing**

  The static FSP configuration parameters are part of the FSP binary and can be customized by external tools that will be provided by Intel.

- **Rebasing**

  The FSP is not Position Independent Code (PIC) and the whole FSP has to be rebased if it is placed at a location which is different from the preferred address during build process.

- **Placing**

  Once the FSP binary is ready for integration, the boot loader build process needs to be modified to place this FSP binary at the specific rebasing location identified above.

- **Interfacing**

  The boot loader needs to add code to setup the operating environment for the FSP, call the FSP with the correct parameters and parse the FSP output to retrieve the necessary information returned by the FSP.

2.2 FSP Distribution Package

The FSP distribution package contains the following:

- FSP Binary - Fsp.fd
- VPD/UPD Data structure definitions - FspUpdVpd.h
- BSF File - Fsp.bsf
- Integration Guide - Braswell_FSP_Integration_Guide.docx

The FSP configuration utility called BCT is available as a separate package.

2.2.1 Layout

- BraswellFspBinPkg
    - Docs
        - Braswell_FSP_Integration_Guide.docx (this doc)
        - FspApi.chm (Doxygen generated version)
    - Include
        - FspUpdVpd.h (FSP UPD and VPD structure and related definitions)
    - BraswellFspBinPkg.dec (EDKII declaration file for package)
    - Fsp.bsf (BSF file for configuring the data using BCT tool)
    - Fsp.fd (FSP Binary)

§

3 FSP Integration

3.1 Assumptions Used in this Document

The FSP for the Braswell platform is built with a preferred base address of **0xFFF20000** and so the reference code provided in the document assumes that the FSP is placed at this base address during the final boot loader build. Users may rebase the FSP binary at a different location with Intel's Binary Configuration Tool (BCT) before integrating to the boot loader.

For other assumptions and conventions, please refer sections 6 in the FSP External Architecture Specification version 1.1.

3.2 FSP INFO Header

The FSP has an Information Header that provides critical information that is required by the bootloader to successfully interface with the FSP. The structure of the FSP Information Header is documented in the section 5.1 of FSP External Architecture Specification version 1.1. Header can be located using the generic algorithm mentioned in the section 5.1.3 of FSP EAS version 1.1.

3.3 FSP Image ID and Revision

The FSP information header contains an Image ID field and an Image Revision field that provide the identification and revision information of the FSP binary. It is important to verify these fields while integrating the FSP as API parameters could change over different FSP IDs and revisions. The FSP API parameters documented in this integration guide are applicable for the Image ID and Revision specified as below. The current FSP version is Alpha release. The ImageId string in the FSP information header is "**$BSWFSP$**" and the ImageRevision field is **0x01010200(1.1.2.0).**

3.4 FSP APIs

This release of the Braswell FSP supports the six APIs as documented in the FSP External Architecture Specification version 1.1.

The FSP information header contains the address offset for these APIs.

The below sections will highlight any changes that are specific to this platform.

3.4.1 TempRamInit API

TempRamInit does basic early initialization primarily setting up temporary RAM using cache. It returns **ECX** pointing to beginning of temporary memory and **EDX** pointing to end of temporary memory. The temporary ram currently available is from **0xFEF0_0000 (ECX)** to **0xFEF0_3FFF (EDX).**

Please refer Chapter 6.5 in the FSP External Architecture Specification version 1.1 for the prototype, parameters and return value details for this API. As this API is executing in stack less environment, it will use XMM6, XMM7, MM2, MM5 and MM7 registers to save/restore all general purpose registers except EAX, ECX and EDX as mentioned in EAS.

Upper CMOS 0x51[7:4] and 0x51[3:0] are reserved as reset indicator, TXE will recognize global reset/ warm reset by these bits.

All known base addresses are programmed as following and enabled in this phase,

ACPI_BASE_ADDRESS : 0x400

GPIO_BASE_ADDRESS : 0x500

SMBUS_BASE_ADDRESS : 0xEFA0

PMC_BASE_ADDRESS : 0xFED03000

ILB_BASE_ADDRESS : 0xFED08000

IO_BASE_ADDRESS : 0xFED80000

SPI_BASE_ADDRESS : 0xFED01000

MPHY_BASE_ADDRESS : 0xFEF00000

PUNIT_BASE_ADDRESS : 0xFED05000

RCBA_BASE_ADDRESS : 0xFED1C000

SMBM_BASE_ADDRESS : 0xFED04000

3.4.2 FspInit API

Please refer Chapter 6.6 in the FSP External Architecture Specification version 1.1 for the prototype, parameters and return value details for this API.

This revision of FSP doesn't have any additional fields other than the FSP_INIT_RT_COMMON_BUFFER mentioned in the FSP EAS version1.1

FSP_INIT_RT_BUFFER contains pointer to an updatable platform configuration data structure UPD_DATA_REGION which is described in section 5.1.2.

MTRRs are programmed to the default values to have the following memory map.

0 – 0x9_FFFF	Write back
0xC_0000 – Top of Low Memory	Write back
FSP Code region in Flash	Write protect
0x1_0000_00000 – Top of High Memory	Write back

3.4.3 NotifyPhase API

Please refer Chapter 6.7 in the FSP External Architecture Specification version 1.1 for the prototype, parameters and return value details for this API.

3.4.4 FspMemoryInit API

Please refer to Chapter 6.8 in the FSP external Architecture Specification version 1.1 for the prototype, parameters and return value details for this API.

FSP_INIT_RT_BUFFER-> Common -> UpdDataRgnPtr is a pointer to an updatable platform configuration data structure MEMORY_INIT_UPD which is described in section 5.1.2.1

3.4.5 TempRamExit API

Please refer to Chapter 6.9 in the FSP external Architecture Specification version 1.1 for the prototype, parameters and return value details for this API

This revision of FSP doesn't have any fields/structure to pass as parameter for this API. Pass Null for *TempRamExitParamPtr.*

At the end of TempRamExit, Code Caching and Cache as Ram is disabled, and all MTRRs are reset to 0.

3.4.6 FspSiliconInit API

Please refer to Chapter 6.10 in the FSP external Architecture Specification version 1.1 for the prototype, parameters and return value details for this API

FspSiliconInitParamPtr is a pointer to an updatable platform configuration data structure SILICON_INIT_UPD which is described in section 5.1.2.2.

It is expected that Firmware will program MTRRs as needed after TempRamExit but before entering FspSilicionInit. If MTRRs are not programmed, FspSiliconInit will program MTRRs default values to have the following memory map.

0 – 0x9_FFFF	Write back
0xC_0000 – Top of Low Memory	Write back
FSP Code region in Flash	Write protect
0x1_0000_00000 – Top of High Memory	Write back

3.5 Boot Flow

Please refer Chapter 4 in the FSP External Architecture Specification version 1.1 for Boot flow chart.

§

4 FSP Output

The FSP builds a series of data structures called the Hand-Off-Blocks (HOBs) as it progresses through initializing the silicon.

Please refer to the *Platform Initialization (PI) Specification - Volume 3: Shared Architectural Elements* specification for PI Architectural HOBs.

Please refer Chapter 7 in the FSP External Architecture Specification version 1.1 for details about FSP Architectural HOBs.

Below section describe the HOBs not covered in the above two specifications.

4.1 SMRAM Resource Descriptor HOB

The FSP will report the system SMRAM T-SEG range through a generic resource HOB if T-SEG is enabled. The owner field of the HOB identifies the owner as T-SEG.

```
#define FSP_HOB_RESOURCE_OWNER_TSEG_GUID \

{ 0xd038747c, 0xd00c, 0x4980, { 0xb3, 0x19, 0x49, 0x01, 0x99,
0xa4, 0x7d, 0x55 } }
```

4.2 SMBIOS INFO HOB

The FSP will report the SMBIOS through a HOB with below GUID. This information can be consumed by the bootloader to produce the SMBIOS tables.

```
#define FSP_SMBIOS_MEMORY_INFO_HOB_GUID \

{ 0x1a1108c, 0x9dee, 0x4984, { 0x88, 0xc3, 0xee, 0xe8, 0xc4,
0x9e, 0xfb, 0x89 } };

#define MAX_CHANNELS_NUM 2
#define MAX_DIMMS_NUM    2

typedef struct {
  UINT8          DimmId;
  UINT32         SizeInMb;
  UINT16         MfgId;
   /* Module part number for DRR3 is 18 bytes but DRR4 is 20
bytes as per JEDEC Spec, so reserving 20 bytes */
  UINT8          ModulePartNum[20];
} DIMM_INFO;

typedef struct {
  UINT8          ChannelId;
  UINT8          DimmCount;
  DIMM_INFO      DimmInfo[MAX_DIMMS_NUM];
} CHANNEL_INFO;

typedef struct {
  UINT8          Revision;
  UINT8          DataWidth;
  /** As defined in SMBIOS 3.0 spec
    Section 7.18.2 and Table 75
  **/
  UINT8          MemoryType;
  UINT16         MemoryFrequencyInMHz;
  /** As defined in SMBIOS 3.0 spec
    Section 7.17.3 and Table 72
  **/
  UINT8          ErrorCorrectionType;
  UINT8          ChannelCount;
  CHANNEL_INFO   ChannelInfo[MAX_CHANNELS_NUM];

} FSP_SMBIOS_MEMORY_INFO;
```

§

5 FSP Configuration Firmware File

The FSP binary contains a configurable data region which will be used by the FSP during the initialization. Please refer Chapter 8 in the FSP External Architecture Specification version 1.1 for details.

Note: When calling the FspInit API, the stack is in temporary memory where the UPD data structure is copied, updated, and passed to the FSP API. When permanent memory is initialized, the FSP will set up a new stack in the permanent memory and tear down the temporary memory. However, the FSP will save the whole boot loader temporary memory region in a GUID HOB. If the boot loader wishes to access the old data in the temporary memory, it can be done by parsing the HOB to retrieve the previous temporary memory data. The migrated temporary memory contains an identical copy of the original data. If pointers are stored in this region, they need to be fixed to point to the new migrated region before using.

5.1 VPD/UPD Data Structure

As stated above, the VPD/UPD data structure and related structure definitions are provided in the FspUpdVpd.h file in the release package. The basic information for each option is provided in the BCT configuration file. The user can use the BCT tool to load this BSF file to get the detailed configuration option information.

5.1.1 VPD Data Region

This VPD data region (VPD_DATA_REGION) can only be configured statically by the BCT tool, and only very limited options in this region can be configured. Most of the configurable options are provided in the UPD data region.

Below is some additional information for some of the fields in VPD_DATA_REGION.

PcdVpdRegionSign
This field is not an option and is a signature for the VPD data region. It can be used by the boot loader to validate the VPD region. This field will not change across different FSP releases for the same silicon set. For the FSP for the Braswell platform, this field is **"$BSWFSP$".**

PcdImageRevision
This field is not an option and is a revision ID for the FSP release. It can be used by the boot loader to validate the VPD/UPD region. If the value in this field is changed for an FSP release, the boot loader should not assume the same layout for the UPD_DATA_REGION/VPD_DATA_REGION data structure. Instead it should use the new FspUpdVpd.h from the FSP release package.

PcdUpdRegionOffset
This field is not an option and contains the offset of the UPD data region within the FSP release image. The boot loader can use it to find the location of UPD_DATA_REGION.

PcdFspReservedMemoryLength
This option is used to specify the reserved memory size for the FSP usage. FSP will consume certain memory resource during the initialization, and this memory range must be reserved. This range will be reported through the HOB described in FSP External Architecture Specification version 1.1 Section 7.2.

5.1.2 UPD Data Region

This UPD data region (UPD_DATA_REGION) can not only be configured statically by the BCT tool in the same way as VPD data region, but also can be overridden by the boot loader at runtime. This provides more flexibility for the boot loader to customize these options dynamically as needed.

Below is the additional information of the fields in UPD_DATA_REGION.

Signature
This field is not an option and is a signature for the UPD data region. It can be used by boot loader to validate the UPD region. The boot loader should never override this field. For the FSP of the Braswell platform, this field is **"$BSWUPD$**".

Revision
Revision version of the UPD_DATA_REGION.

MemoryInitUpdOffset
This field contains the offset of the MemoryInitUpd structure relative to UPD_DATA_REGION

SiliconInitUpdOffset
This field contains the offset of the SiliconInitUpd structure relative to UPD_DATA_REGION

MemoryInitUpd and **SiliconInitUpd** fields are described in section 5.1.2.1 and 5.1.2.2 respectively. **PcdRegionTerminator**
This field is will have a value 0x55AA indicating the end of UPD data. We should not change it.

5.1.2.1 MemoryInitUpd

All below UPD parameters are part of the **MemoryInitUpd** and consumed in FspMemoryInit API

MemoryInitUpdSignature
This field is not an option and is a signature for the MemoryInitUPD data region. It can be used by boot loader to validate this UPD region. The boot loader should never override this field. For the FSP of the Braswell platform, this field is **"$MEMUPD$**".

MemoryInitUpdRevision

Revision version of the **MemoryInitUpd** Region

PcdMrcInitTsegSize

This field is used to specify the Size of SMRAM (TSEG) memory reserved.

0x01 - "1 MB"
0x02 - "2 MB"
0x04 - "4 MB" (Default)
0x08 - "8 MB"

PcdMrcInitMmioSize

This field is used to specify the Size of memory address space reserved for MMIO (Memory Mapped I/O).
0x400 - "1.0 GB"
0x600 - "1.5 GB"
0x800 - "2.0 GB" (Default)

PcdMrcInitSpdAddr1

This Field is used to specify the DIMM 0 SPD SMBus Address.
Default is 0xA0

PcdMrcInitSpdAddr2

This Field is used to specify the DIMM 1 SPD SMBus Address.
Default is 0xA2

PcdMemChannel0Config and PcdMemChannel1Config

Input table describing memory population. Valid inputs are listed in OEM_MEMORY_DIMM_TYPE enum

```
typedef enum {
  DimmInstalled = 0,
  SolderDownMemory,
  DimmDisabled
} OEM_MEMORY_DIMM_TYPE;
```

DimmInstalled – Attempt to use SPD data on DIMM EEPROM. Disable channel if no SPD data is found.

SolderDownMemory – Use SPD table pointed by **PcdMemorySpdPtr** for SPD data.

DimmDisabled – Disable DIMM slot.

PcdMemorySpdPtr

Pointer to table containing SPD data for configuring DIMMS

PcdIgdDvmt50PreAlloc

This field specifies the Size of memory preallocated for internal graphics
0x01 - "32 MB" (Default)
0x02 - "64 MB"
0x03 - "96 MB"
0x04 - "128 MB"
0x05 - "160 MB"

0x06 - "192 MB"
0x07 - "224 MB"
0x08 - "256 MB"
0x09 - "288 MB"
0x0A - "320 MB"
0x0B - "352 MB"
0x0C - "384 MB"
0x0D - "416 MB"
0x0E - "448 MB"
0x0F - "480 MB"
0x10 - "512 MB"

PcdApertureSize

This field specifies the Aperture Size
0x1 - "128 MB"
0x2 - "256 MB" (Default)
0x3 - "512 MB"

PcdGttSize

This field specifies the GTT Size
0x1 - "1 MB"   (Default)
0x2 - "2 MB"

PcdLegacySegDecode

If disabled, E0000h-FFFFFh decoding will be routed to DRAM
If enabled Legacy E/F segments decoding to ROM
0x0 - "Disabled" (Default)
0x1 - "Enabled"

PcdDvfsEnable

Enable/disable Dvfs
0x00 – "Disable"
0x01 – "Enable"

PcdMemoryTypeEnable

Select Memory Type
0x00 – DDR3
0x01 – LPDDR3

PcdCaMirrorEn

Disable or enable CaMirrorEn
0x00 – Disable
0x01 – Enable

5.1.2.2 SiliconInitUpd

All below UPD parameters are part of the **SiliconInitUpd** and are consumed in FspSiliconInit API

SiliconInitUpdSignature

This field is not an option and is a signature for the **SiliconInitUpd** data region.. It can be used by boot loader to validate this UPD region. The boot loader should never override this field. For the FSP of the Braswell platform, this field is **"SI_UPD**".

SiliconInitUpdRevision

Revision version of the **MemoryInitUpd** Region

PcdSdcardMode

This Field is used to select the SD Card Mode

0x0 - "Disabled"
0x1 - "PCI Mode" (Default)
0x2 - "ACPI Mode"

PcdEnableHsuart0

Enable/disable HSUART0

0x0 - "Disabled" (Default)
0x1 - "Enabled"

PcdEnableHsuart1

Enable/disable HSUART1

0x0 - "Disabled"
0x1 - "Enabled" (Default)

PcdEnableAzalia

Enable/disable Azalia controller

0x0 - "Disabled" (Default)
0x1 - "Enabled"

AzaliaVerbTablePtr

This field is a pointer to a BL_PCH_AZALIA_CONFIG structure, which provides the configuration parameters including the codec verb table for Azalia in the Braswell SOC. Please refer to FspUpdVpd.h for the definition of BL_PCH_AZALIA_CONFIG structure.

PcdEnableSata

Enable/disable SATA controller

0x0 - "Disabled"
0x1 - "Enabled" (Default)

PcdEnableXhci

Enable/disable XHCI controller

0x0 - "Disabled"
0x1 - "Enabled" (Default)

PcdEnableLpe

This field is to select the LPE mode

0x0 - "Disabled"
0x1 - "PCI Mode" (Default)
0x2 - "ACPI Mode"

PcdEnableDma0

Enable/disable DMA0

0x0 - "Disabled"
0x1 - "Enabled" (Default)

PcdEnableDma1

Enable/disable DMA1

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C0

Enable/disable I2C0

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C1

Enable/disable I2C1

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C2

Enable/disable I2C2

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C3

Enable/disable I2C3

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C4

Enable/disable I2C4

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C5

Enable/disable I2C5

0x0 - "Disabled"

0x1 - "Enabled” (Default)

PcdEnableI2C6

Enable/disable I2C6

0x0 - "Disabled"

0x1 - "Enabled” (Default)

GraphicsConfigPtr

This field is a pointer to the graphics configuration data used when initializing graphics. If this field is NULL, then FSP will skip graphics initialization. The graphics configuration data is generated via the Intel CED-Lite tool. Please contact your Intel representative to obtain this tool.

GpioFamilyInitTablePtr

This field is a pointer to an array of BL_GPIO_FAMILY_INIT structures, which provide the initialization parameters for the GPIO families in the Braswell SOC. The very last entry of the array must be filled with FFs indicating the end of the array. If the pointer is set to NULL, FSP will not initialize the GPIO families in the Braswell SOC. Please refer to FspUpdVpd.h for the definition of BL_GPIO_FAMILY_INIT structure.

GpioPadInitTablePtr

This field is a pointer to an array of BL_GPIO_PAD_INIT structures, which provides the initialization parameters for the GPIO pads in the Braswell SOC. The very last entry of the array must be filled with FFs indicating the end of the array. If the pointer is set to NULL, FSP will not initialize the GPIO pads in the Braswell SOC. Please refer to FspUpdVpd.h for the definition of BL_GPIO_PAD_INIT structure. If the pointer is set to NULL, FSP will not initialize GPIO.

PunitPwrConfigDisable

P-Unit Power Config disable
0x0 - "Disabled" (Default)
0x1 - "Enabled"

ChvSvidConfig

BSW SVID config: default config0

Rails	**Silicon**		**A3 onwards platform config recommendation**		
	Current Ax config	A3 onwards (**default**)	Config 1	Config 2 (BSW+ PMIC)	Config 3
Vcc0	SVID Split Vcc	**Fixed** / Merged variable VID / Split variable VID **SVID** / I2C	Merged variable VID SVID	Merged variable VID I2C	Merged variable VID SVID
Vcc1	SVID Split Vcc	**Fixed** / Merged variable VID / Split variable VID **SVID** / I2C			
Vgg	SVID	**Fixed** / Variable VID	Variable VID	Variable VID	Variable

		SVID / I2C	SVID	I2C	SVID
Vnn	SVID	**Fixed** / Variable VID **SVID** / I2C	Variable VID SVID	Variable VID I2C	Fixed VID

0x0 – Config 0 (default)
0x1 – Config 1
0x2 – config 2
0x3 – config 3

DptfDisable

This Field is used to Disable/Enable DPTF.
0x0 - "Disabled" (Default)
0x1 - "Enable"

PcdEmmcMode

This Field is used to select the eMMC mode.
0x0 - "Disabled" (Default)
0x1 - "PCI Mode"
0x2 - "ACPI Mode"

PcdUsb3ClkSsc

USB3 Clock Spread Spectrum feature
0x0 - "Disabled”
0x1 - "Enabled" (Default)

PcdDispClkSsc

Display Clock Spread Spectrum feature
0x0 - "Disabled”
0x1 - "Enabled" (Default)

PcdSataClkSsc

Sata Clock Spread Spectrum feature
0x0 - "Disabled”
0x1 - "Enabled" (Default)

Usb2Port0PerPortPeTxiSet

USB2 port0 PerPortPeTxiSet
7 (Default)

Usb2Port0PerPortTxiSet

USB2 port0 PerPortTxiSet
5 (Default)

Usb2Port0IUsbTxEmphasisEn

USB2 port0 UsbTxEmphasisEn
2 (Default)

Usb2Port0PerPortTxPeHalf
USB2 port0 PerPortTxPeHalf
1 (Default)

Usb2Port1PerPortPeTxiSet
USB2 port1 PerPortPeTxiSet
7 (Default)

Usb2Port1PerPortTxiSet
USB2 port1 PerPortTxiSet
3 (Default)

Usb2Port1IUsbTxEmphasisEn
USB2 port0 UsbTxEmphasisEn
2 (Default)

Usb2Port1PerPortTxPeHalf
USB2 port0 PerPortTxPeHalf
1 (Default)

Usb2Port2PerPortPeTxiSet
USB2 port2 PerPortPeTxiSet
7 (Default)

Usb2Port2PerPortTxiSet
USB2 port2 PerPortTxiSet
3 (Default)

Usb2Port2IUsbTxEmphasisEn
USB2 port2 UsbTxEmphasisEn
2 (Default)

Usb2Port2PerPortTxPeHalf
USB2 port2 PerPortTxPeHalf
1 (Default)

Usb2Port3PerPortPeTxiSet
USB2 port3 PerPortPeTxiSet
7 (Default)

Usb2Port3PerPortTxiSet
USB2 port3 PerPortTxiSet
3 (Default)

Usb2Port3IUsbTxEmphasisEn
USB2 port3 UsbTxEmphasisEn
2 (Default)

Usb2Port3PerPortTxPeHalf
USB2 port3 PerPortTxPeHalf
1 (Default)

Usb2Port4PerPortPeTxiSet

USB2 port4 PerPortPeTxiSet
7 (Default)

Usb2Port4PerPortTxiSet

USB2 port4 PerPortTxiSet
3 (Default)

Usb2Port4IUsbTxEmphasisEn

USB2 port4 UsbTxEmphasisEn
2 (Default)

Usb2Port4PerPortTxPeHalf

USB2 port4 PerPortTxPeHalf
1 (Default)

Usb3Lane0Ow2tapgen2deemph3p5

USB3 Lane0 ow2tapgen2deemph3p5
0x3a (Default)

Usb3Lane1Ow2tapgen2deemph3p5

USB3 Lane1 ow2tapgen2deemph3p5
0x64 (Default)

Usb3Lane2Ow2tapgen2deemph3p5

USB3 Lane2 ow2tapgen2deemph3p5
0x64 (Default)

Usb3Lane3Ow2tapgen2deemph3p5

USB3 Lane3 ow2tapgen2deemph3p5
0x3a (Default)

PcdSataInterfaceSpeed

This field specifies the SATA controller Interface Speed
1 - "GEN1"
2 - "GEN2"
3 - "GEN3" (Default)

PcdPchUsbSsicPort

SSIC Port Enable/Disable
0x0 - "Disabled" (Default)
0x1 - "Enabled"

PcdPchUsbHsicPort

HSCI Port Enable/Disable
0x0 - "Disabled" (Default)
0x1 - "Enabled"

PcdPcieRootPortSpeed

Pci Root Port speed configuration
0x0 – "Auto" (Default)
0x1 - Gen1
0x2 – Gen2

PcdPchSsicEnable

PCH USB Ssic Enable/Disable
0x0 - "Disabled"
0x1 - "Enabled" (Default)

PcdLogoPtr

Pointer holding BIOS Logo image location
0x00000000 – (Default)

PcdLogoSize

Size of the BIOS Logo Image
0x00000000 – (Default)

PcdRtcLock

To lock RTC register not to protect from others
0x0 – unlock (Default)

0x1 – lock**PMIC_I2CBus**

This is used to convey about which I2C bus used to communicate with PMIC and which is based on the Platform design
0 – I2C Bus 0

ISPEnable

Enable/disable ISP
0x0 - "Disabled"
0x1 - "Enabled" (Default)

ISPPciDevConfig

0x2: ISP as PCI Device 2, 0x3: ISP as PCI Device 3
0x3 – (Default)

PcdTurboMode

0x01 – Enable
0x00 – Disable

PcdPnpSettings

Select the table for Power and Performance Setting
0x00 - Disable
0x01 - Power
0x02 - Performance
0x03 - power & performance

PcdSdDetectChk

Sd detect check is required or not
0x00 - Disable
0x01 - Enable