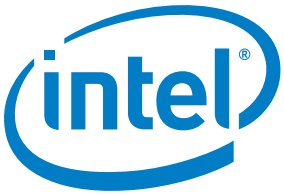


Intel® Firmware Support Package for Skylake Platform

Integration Guide

April 2016

Contents

§

1 Introduction

1.1 Purpose

The purpose of this document is to describe the steps required to integrate the Intel® Firmware Support Package (FSP) into a boot loader solution. It supports Skylake platforms with the Skylake processor with the Sunrise Point Platform Controller Hub (PCH).

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 http://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/fsp-architecture-spec.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
FSP API	Firmware Support Package Interface
PCH	Platform Controller Hub
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
- VPD/UPD Data structure definitions
- BSF File
- Integration Guide

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

2.3 Distribution Package

- Docs
    - Skylake_FSP_Integration_Guide.docx (this doc)
- Include
    - FspUpdVpd.h FSP UPD and VPD structure and related definitions
    - GpioSampleDef.h (Sample enum definitions for Gpio table)
    -
- Fsp.bsf (BSF file for configuring the data using BCT tool)
- Fsp.fd (FSP Binary)

§

FSP Integration

2.4 Assumptions Used in this Document

The FSP for the Skylake platform is built with a preferred base address of **0xFFEE0000** 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.

2.5 Boot Flow

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

2.6 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 FSP External Architecture Specification version 1.1 with a HeaderRevision of 2.

2.7 FSP Image ID and Revision

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 ImageId string in the FSP information header is "**$SKLFSP$**" and the ImageRevision field is **0x02000000**.(2.0.0.0)

2.8 FSP APIs

This release of the Skylake 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. Register usage is described in the FSP External Architecture Specification version 1.1. Any usage not described by the specification is described in the individual functions.

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

2.8.1 TempRamInit API

Please refer Chapter 6.5 in the FSP External Architecture Specification version 1.1 for complete details including the prototype, parameters and return value details for this 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 + 1. The temporary ram currently available is from 0xFEF0_0000 (ECX) to 0xFEF1_0000 (EDX).

TempRamInit also sets up the code caching of the region passed CodeCacheBase and CodeCacheLength, which are input parameters to TempRamInitApi. If CodeCacheLength is larger than available cache, it will be reduced to available cache. Also, if 0 is passed in for CodeCacheBase, the base used will be 4 GB - 1 - length to be code cached instead of starting from CodeCacheBase.

It is a requirement for Firmware to have Firmware Interface Table, which contains pointers to each microcode update. The microcode update is loaded for all logical processors before reset vector. If more than microcode update for the CPU is present, the microcode update with the latest revision is loaded.

MicrocodeRegionBase and MicrocodeRegionLength are input parameters to TempRamInit API. If these values are 0, FSP will not attempt to update microcode. If a region is passed, then if a newer microcode update revision is in the region, it will be loaded by the FSP.

This API uses starts executing stackless, so CPU registers mm5, mm6, mm7, xmm5, xmm6, and xmm7 are used to save and restore registers during this phase.

2.8.2 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 4.1.2.

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

FspInit does the programming in order of FspMemoryInit, TempRamExit, and FspSilicon APIs. Please refer to these sections for more specific programming.

This API should be called only once after the TempRamInit API.

Use of this API is mutually exclusive to the FspMemoryInit, TempRamExit and FspSilicon APIs.

2.8.3 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 4.1.2.1

The base address of HECI device (Bus 0, Device 22, Function 0) is required to be initialized prior to perform FspMemoryInit flow. The default address is programmed to 0xFED1A000.

Calculate memory map determining memory regions TSEG, IED, GTT, BDSM, ME stolen, Uncore PMRR, IOT, MOT, DPR, REMAP, TOLUD, TOUUD. Programming will be done at a different time.

2.8.4 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.

2.8.5 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 4.1.2.2.

It is expected that boot loader will program MTRRs as needed after TempRamExit but before entering FspSiliconInit. 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

It is a requirement for bootloader to have Firmware Interface Table (FIT), which contains pointers to each microcode. The microcode is loaded for all cores before reset vector. If UPD SkipMpInit is not enabled, microcode will be loaded a second time if a later microcode revision is provided as input.

MicrocodeRegionBase and MicrocodeRegionLength are both input parameters to TempRamInit and UPD for SiliconInit API. UPD has priority and will be searched for a

later revision than TempRamInit. If MicrocodeRegionBase and MicrocodeRegionLength values are 0, FSP will not attempt to update the microcode. If a microcode region is passed, and if a later revision of microcode is present in this region, FSP will load it.

FSP initializes PCH audio including selecting HD Audio verb table and initializes Codec.

PCH required initialization is done for the following HECI, USB, HSIO, Integrated Sensor Hub, Display, Sky Cam, Camera, PCI Express, Vt-d, straps (cores, hyper-threading, BIST, ..)

FSP initializes CPU features if UPD SkipMpInit is not enabled: XD, VMX, AES, IED, HDC, x(2)Apic, Intel® Processor Trace, Three strike counter, Machine check, Cache pre-fetchers, Core PMRR, Power management.

Initializes HECI, DMI, Internal Graphics. Publish EFI_PEI_GRAPHICS_INFO_HOB during normal boot but this HOB will not be published during S3 resume as FSP will not launch the PEI Graphics PEIM during S3 resume.

Programs SA Bars: MchBar, DmiBar, EpBar, GdxcBar, EDRAM (if supported). Please refer to section 2.8 (MemoryMap) for the corresponding Bar values. GttMmadr (`0xDF000000`) and GmAdr(0xC0000000) are temporarily programmed and cleared after use in FSP.

On normal boot FSP CPU S3 Resume Hob is produced in this phase. This FSP CPU S3 Resume HOB is described in section 3.3. Unless SkipMpInit is enabled, on S3 resume, this data (not the entire HOB) must be passed through UPD CpuS3ResumeData, and optionally final S3 boot MTRRs is passed through UPD CpuS3ResumeMtrrData. During S3 resume unless SkipMpInit is enabled, GDT base and length and IDT base and length on APs are programmed to that of the BSP.

2.8.6 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.

2.8.6.1 Phase1

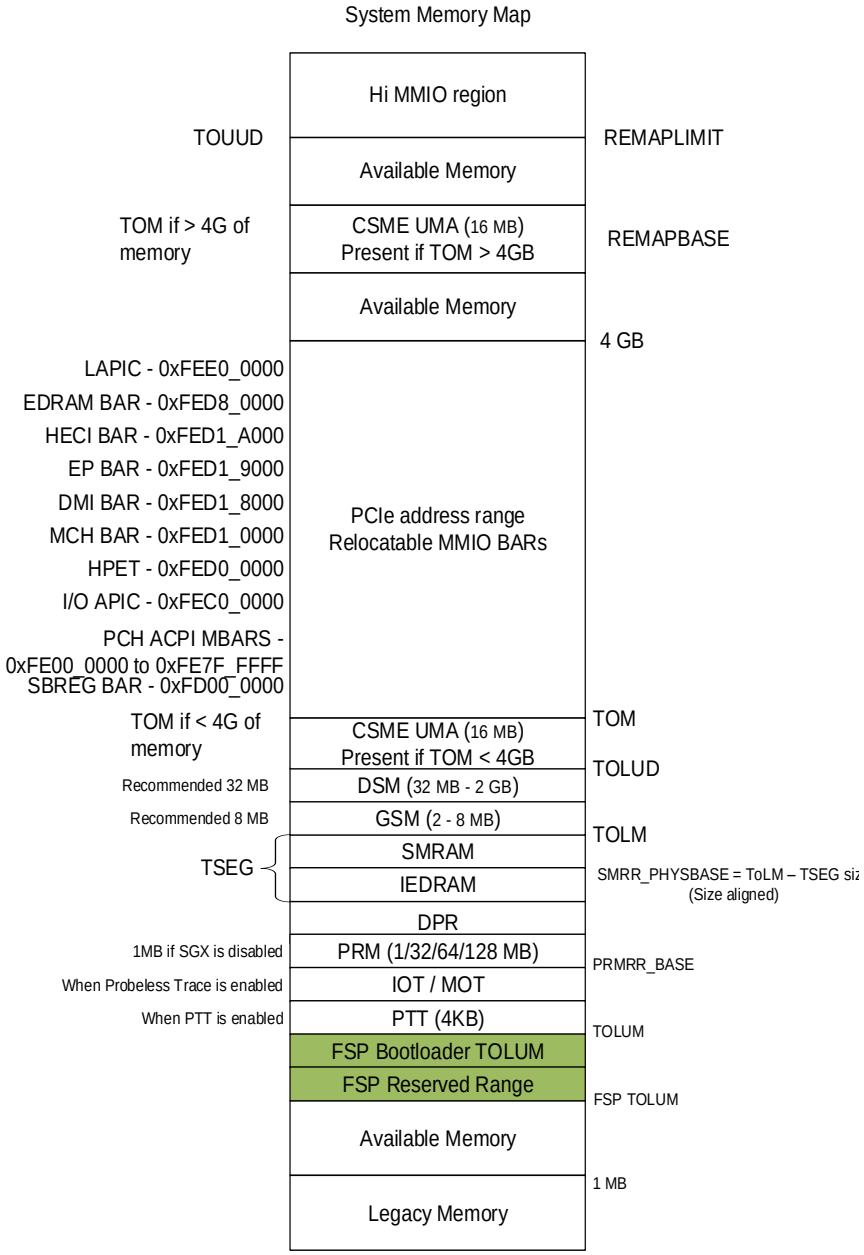
This phase EnumInitPhaseAfterPciEnumeration is to be called after PCI enumeration but before execution of third party code such as option ROMs. Currently, nothing is done in this phase, but in the future updates, programming may be done in this phase.

2.8.6.2 Phase2

This phase EnumInitPhaseReadyToBoot is to be called before giving control to boot. Finalize SATA configuration, set flash protection, final locking of chipset registers. Send ME Message EOP (End of Post).

2.9 Memory Map

Below diagram represents the memory map allocated by FSP including the FSP specific regions.


System Memory Map
Hi MMIO region
TOUUD
REMAPLIMIT
Available Memory
TOM if > 4G of memory
CSME UMA (16 MB)
Present if TOM > 4GB
REMAPBASE
Available Memory
4 GB
LAPIC - 0xFEE0_0000
EDRAM BAR - 0xFED8_0000
HECI BAR - 0xFED1_A000
EP BAR - 0xFED1_9000
DMI BAR - 0xFED1_8000
MCH BAR - 0xFED1_0000
HPET - 0xFED0_0000
I/O APIC - 0xFEC0_0000
PCH ACPI MBARS - 0xFE00_0000 to 0xFE7F_FFFF
SBREG BAR - 0xFD00_0000
PCIe address range
Relocatable MMIO BARs
TOM if < 4G of memory
CSME UMA (16 MB)
Present if TOM < 4GB
TOM
TOLUD
Recommended 32 MB
DSM (32 MB - 2 GB)
Recommended 8 MB
GSM (2 - 8 MB)
TOLM
TSEG
SMRAM
IEDRAM
SMRR_PHYSBASE = ToLM – TSEG siz
(Size aligned)
DPR
1MB if SGX is disabled
PRM (1/32/64/128 MB)
PRMRR_BASE
When Probeless Trace is enabled
IOT / MOT
When PTT is enabled
PTT (4KB)
TOLUM
FSP Bootloader TOLUM
FSP Reserved Range
FSP TOLUM
Available Memory
1 MB
Legacy Memory

§

3 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.

3.1 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 } }
```

3.2 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. These structures are included as part of FspUpdVpd.h

```
#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 DDR3 is 18 bytes however for DRR4 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;
  /** 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;
```

3.3 FSP CPU S3 Resume HOB

The FSP will report the CPU S3 Resume data through a GUIDED HOB with below GUID. This data (not the entire HOB) must be passed during S3 resume passed in UPD CpuS3ResumeData except if UPD SkipMpInit is enabled.

```
#define FSP_CPU_S3_RESUME_HOB_GUID \
{0x6b819940, 0xdc0e, 0x4d52, { 0x8f, 0xa, 0x92, 0xcb, 0xe4, 0xfd,
0xff, 0x58 }}
```

4 FSP Configuration Firmware File

The FSP binary contains a configurable data region which will be used by the FSP during the initialization. The configurable data region has two sets of data

VPD – Vital Product Data, which can only be configured statically

UPD – Updatable Product Data, which can be configured statically for default values, but also can be overridden during boot at runtime.

Both the VPD and the UPD parameters can be statically customized using a separate tool called the Binary Configuration Tool (BCT) as explained in the tools section. The tool will use a Boot Setting File (BSF) to understand the layout of the configuration region within the FSP.

In addition to static configuration, the UPD data can be overridden by the boot loader during runtime. The UPD data is organized as a structure. The FspInit and FspMemoryInit APIs parameter includes an **`UpdDataRgnPtr`** pointer which can be initialized to point to the UPD data structure. Please refer to Chapter 6 and Chapter 8 in the FSP External Architecture Specification version 1.1 for details.

Note: To update these configuration options statically using the BCT, a BSF file will be required. This file contains the detailed information on all configurable options, including description, help information, valid value range and the default value. Refer to the **Fsp.bsf** file in the release package for more information.

4.1 VPD/UPD Data Structure

As stated above, the VPD/UPD data structure and related structure definitions are provided in the sample file FspUpdVpd.h. 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.

4.1.1 VPD Data Region

This VPD data region (VPD_DATA_REGION) can only be configured statistically 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. This should match the ImageId in FSP_INFO_HEADER. For the FSP for the Skylake platform, this field is "**$SKLFSP$**".

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. This should match the ImageRevision in FSP_INFO_HEADER.

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.

PcdSerialIoUartDebugEnable

This field is set TRUE if Serial IO for debugging is enabled.

PcdSerialIoUartNumber

This field specifies which UART is used, and is in the range of 0 to 2.

PcdSerialIoUartInputClock

This field specifies UART clock frequency. This currently isn't used.

4.1.2 UPD Data Region

This UPD data region (UPD_DATA_REGION) can not only be configured statistically 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 some additional information for some 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 Skylake platform, this field is **$SKLUPD$**.

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 4.1.2.1 and 4.1.2.2 respectively.

PcdRegionTerminator
This field is not an option and is a termination field at the end of the data structure. This field is will have a value 0x55AA indicating the end of UPD data.The boot loader should never override this field.

Reserved/Unused
The UPD_DATA_REGION may contain some reserved or unused fields in the data structure. These fields are required to use the default values provided in the FSP binary. Intel always recommends copying the whole UPD_DATA_REGION from the flash to a local structure in the stack before overriding any field.

4.1.2.1 MemoryInitUpd

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

Signature
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 Skylake platform, this field is **"$MEMUPD$"**.

Revision
Revision version of the **MemoryInitUpd** Region

PlatformMemorySize
The minimum platform memory size required to pass control into DXE

MemorySpdPtr00
Pointer to SPD data in Memory for Channel-0:DIMM-0. Default: 0 for empty channel.

MemorySpdPtr01
Pointer to SPD data in Memory for Channel-0:DIMM-1. Default: 0 for empty channel.

MemorySpdPtr10

Pointer to SPD data in Memory for Channel-1:DIMM-0. Default: 0 for empty channel.

MemorySpdPtr11
Pointer to SPD data in Memory for Channel-1:DIMM-1. Default: 0 for empty channel.

MemorySpdDataLen
Length of SPD data 256 or 512 bytes. Default: 256

DqByteMapCh0
Byte mask array mapping CPU DQ lanes to CMD/CTL/CLK to DRAM Channel 0.
Default: [0x0F, 0xF0, 0x00, 0xF0, 0x0F, 0xF0, 0x0F, 0x00, 0xFF, 0x00, 0xFF, 0x00]

DqByteMapCh1
Byte mask array mapping CPU DQ lanes to CMD/CTL/CLK to DRAM Channel 1.
Default: [0x0F, 0xF0, 0x00, 0xF0, 0x0F, 0xF0, 0x0F, 0x00, 0xFF, 0x00, 0xFF, 0x00]
Description of DqByteMap:
DqByteMap[0] – Maps which CPU bytes are tied to CLK0.
DqByteMap[1] – Maps which CPU bytes are tied to CLK1.
DqByteMap[2] – Unused, set to 0.
DqByteMap[3] – Maps which CPU Bytes are tied to the DRAM device routed to Command/Address B signals.
DqByteMap[4] – Maps which CPU Bytes are tied to the DRAM device routed to Command/Address A signals.
DqByteMap[5] – Maps which CPU Bytes are tied to the DRAM device routed to Command/Address B signals.
DqByteMap[6] – Set which CPU Bytes are tied to the DRAM device routed to Command/Address A signals.
DqByteMap[7] – Unused, set to 0.
DqByteMap[8] – CtlDqByteMap Always 0xFF for 1 CTL/rank.
DqByteMap[9] – Unused, set to 0.
DqByteMap[10] – CmdVDQByteMap Always 0xFF, 0 for 1 CA Vref.
DqByteMap[11] – Unused, set to 0.

DqsMapCpu2DramCh0
Dqs swizzling between CPU and DRAM, Byte0, Byte1, … byte 7 for Channel 0.
Default: [2, 0, 1, 3, 6, 4, 7, 5, 0]

DqsMapCpu2DramCh1
Dqs swizzling between CPU and DRAM, Byte0, Byte1, … byte 7 for Channel 1.

RcompResistor
RCOMP resister values on motherboard in Ohms. Value specified by platform design guide.
Three 16-bit values. Default: 200, 81, 162

RcompTarget
RCOMP target values for RdOdt, WrDS, WrDSCmd, WrDSCtl, WrDSClk in Ohms from platform design guide.
Five 16-bit values. Default: 100, 40, 40, 23, 40

DqPinsInterleaved
Interleaving mode of DQ/DQS pins of board routed input: Enable or Disable.
Default: Disabled

CaVrefConfig
This platform design:
0 = VREF_CA goes to both CH_A and CH_B
1 = VREF_CA to CH_A and VREF_DQ_A to CH_B
2 = VREF_CA to CH_A and VREF_DQ_B to CH_B

Default: 0

SmramMask
The SMM Regions AB-SEG and/or H-SEG reserved:
0: Neither
1:AB-SEG
2:H-SEG
3: Both
Default: 0

MrcFastBoot
Enable/Disable MRC Fastboot. Default: Enable

IedSize
Reserved SMRAM for (IED) Intel Enhanced Debug. 0=0MB=Disable, 0x400000=4MB=Enabled with 4MB SMRAM occupied. Default: 0 (Disabled)

TsegSize
Size of SMRAM (TSEG) memory reserved in bytes. Default: 0x400000 (4MB)

MmioSize
Size of memory address space reserved for MMIO (Memory Mapped I/O) in megabytes. Default: 0x400 (1 GB).

ProbelessTrace
Enable/Disable Probeless Trace. Enabling Probeless Trace will reserve 128 MB, and IED is required to be enabled. Default: 0 (Disabled)

EnableTraceHub
Enable/Disable PCH Trace Hub, Default:0(Disable)

IgdDvmt50PreAlloc
Size of memory preallocated for internal graphics.
0: 0 MB, 1: 32 MB, 2: 64 MB, 3: 128 MB, 4: 256 MB, 5: 512 MB
Default: 1

InternalGfx

Enable/disable internal graphics. Default : 1 (Enabled)

ApertureSize
Aperture Size. 0:128 MB, 1:256 MB, 2:512 MB. Default: 1

SaGv
SA GV Configuration. System Agent dynamic frequency support and when enabled memory will be training at two different frequencies. Only affects ULX/ULT CPUs. 0=Disabled, 1=FixedLow, 2=FixedHigh, and 3=Enabled. Default: 0

RMT
Enable/disable Rank Margin Tool. Default: 0

DdrFreqLimit

Memory Frequency Limit in Mhz, Options: 1067, 1333, 1600, 1867, 2133, 2400, Auto(MAX frequency limited by MRC to meet MC capability, OC or DDR type limitations.)

UserBd
MrcBoardType, Options are 0=Mobile/Mobile Halo, 1=Desktop/DT Halo, 5=ULT/ULX/Mobile Halo, 7=UP Server

MmaTestContentPtr
Pointer to MMA Test Content in Memory.

MmaTestContentSize
Size of MMA Test Content in Memory.

MmaTestConfigPtr
Pointer to MMA Test Config in Memory.

MmaTestConfigSize
Size of MMA Test Config in Memory.

FspCarBase
FSP CAR Base.

FspCarSize
FSP CAR Size.

4.1.2.2 SiliconInitUpd

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

Revision

Revision version of the **SiliconInitUpd** Region

LogoPtr
Points to PEI Display Display Logo BMP Image.

LogoSize
Points to PEI Display Logo Size.

GraphicsConfigPtr
Points to PEI VBT.

Device4Enable

Enable/Disable Device 4 (Camarillo Thermal Device). Default: 0 (Disabled)

EnableAzalia
Enable/disable Azalia controller. Default : 1 (Enabled)

DspEnable
Enable/disable HD Audio DSP feature. Default: 1(Enable)

IoBufferOwnership

Indicates the ownership of the I/O buffer between Intel HD Audio link vs I2S0 / I2S port.

0: Intel HD-Audio link owns all the I/O buffers.

1: Intel HD-Audio link owns 4 of the I/O buffers for 1 HD-Audio codec connection, and I2S1 port owns 4 of the I/O buffers for 1 I2S codec connection.

2: Reserved.

3: I2S0 and I2S1 ports own all the I/O buffers.

Cio2Enable
Enable/disable SKYCAM CIO2 Controller. Default: 1 (Enable)

ScsEmmcEnabled
Enable/disable eMMC Controller. Default: 1 (Enable)

ScsEmmcHs400Enabled
Enable eMMC HS400 Mode. Default:1 (Enable)

ScsSdCardEnabled
Enable/disable SD Card Controller. Default: 1 (Enable)

IshEnable
Enable/disable ISH Controller. Default:1 (Enable)

ShowSpiController
Enable/disable to show SPI controller. Default: 0 (Disable)

HsioMessaging
Enable/Disable. 0: Disable, prevent the HSIO version check and HSIO init messages from being sent, 1: enable

Heci3Enabled
The HECI3 state from Mbp for reference in S3 path or when MbpHob is not installed. If Integrated Touch (iTouch) is supported, then this must be enabled. 0: disable, 1: enable

Default: 0

MicrocodeRegionBase
Current MicrocodeRegionBase and MicrocodeRegionSize are provided in the TempRamInit Api. If UPDs MicrocodeRegionBase and MicrocodeRegionSize are updated, this region will be used for reloading microcode. If no Microcode is found in this region, the region TempRamInit Api Region will be used.
Default: 0

MicrocodeRegionSize
Current MicrocodeRegionBase and MicrocodeRegionSize are provided in the TempRamInit Api. If UPDs MicrocodeRegionBase and MicrocodeRegionSize are updated, this region will be used for reloading microcode. If no Microcode is found in this region, the region TempRamInit Api Region will be used.
Default: 0

TurboMode
This enables or disables CPU Turbo Mode.
Default: 1

SataSalpSupport
Enable/disable SATA Aggressive Link Power Management

Default: 1(Enable)

SataPortsEnable[8]
Enable/disable SATA ports. One byte for each port, byte0 for port0, byte1 for port1, and so on.

Default: All Ports enabled { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01}

SataPortsDevSlp[8]
Enable/disable SATA DEVSLP per port. 0 is disable, 1 is enable. One byte for each port, byte0 for port0, byte1 for port1, and so on.

Default: { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}

PortUsb20Enable[16]
Enable/disable per USB2 ports. One byte for each port, byte0 for port0, byte1 for port1, and so on.

Default: { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01}

PortUsb30Enable[10]
Enable/disable per USB3 ports. One byte for each port, byte0 for port0, byte1 for port1, and so on.

Default: { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01}

XdciEnable
Enable/Disable xDCI controller. Default:1 (Enabled)

SsicPortEnable
Enable/disable XHCI SSIC port. Default:1 (Enable)

SmbusEnable
Enable/Disable smbus. Default:1 Enable

SerialIoDevMode[11]
Enable/disable SerialIo
I2C0,I2C1,I2C2,I2C3,I2C4,I2C5,SPI0,SPI1,UART0,UART1,UART2 device mode respectively. One byte for each controller, byte0 for I2C0, byte1 for I2C1, and so on.

Default: { 0x02, 0x02, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x02, 0x00, 0x04}
0:Disabled
1:ACPI Mode
2:PCI Mode
3:Hidden mode
4:Legacy UART mode

DevIntConfigPtr
The address of the table of PCH_DEVICE_INTERRUPT_CONFIG.

NumOfDevIntConfig
Number of Device Interrupt Configuration Entry. If this is not zero, the DevIntConfigPtr must not be NULL.

PxRcConfig[8]
PIRQx to IRQx mapping. The valid value is 0x00 to 0x0F for each. First byte is for PIRQA, second byte is for PIRQB, and so on. The setting is only available in Legacy 8259 PCI mode.

GpioIrqRoute
GPIO IRQ Select. The valid value is 14 or 15.

SciIrqSelect
SCI IRQ Select. The valid value is 9, 10, 11, and 20, 21, 22, 23 for APIC only.

TcoIrqSelect
TCO IRQ Select. The valid value is 9, 10, 11, 20, 21, 22, 23.

TcoIrqEnable
Enable/disable TCO IRQ.

AzaliaVerbTableNumEntries
Number of Entries in Azalia Verb Table.

AzaliaVerbTablePtr
Pointer to Array of pointers to Codec Table.

LockDownConfigRtcLock
Enable RTC lower and upper 128 byte Lock bits to lock Bytes 38h-3Fh in the upper and and lower 128-byte bank of RTC RAM.

EnableSata
Enable/disable SATA controller. Default : 1 (Enabled)

SataMode
IDE, AHCI, or RAID mode.
0 :IDE, 1:AHCI, 2:RAID Default: 1

Usb2AfePetxiset[16]
USB Per Port HS Preemphasis Bias. 000b-0mV, 001b-11.25mV, 010b-16.9mV, 011b-28.15mV, 100b-28.15mV, 101b-39.35mV, 110b-45mV, 111b-56.3mV. One byte for each port.

Usb2AfeTxiset[16]
USB Per Port HS Transmitter Bias. 000b-0mV, 001b-11.25mV, 010b-16.9mV, 011b-28.15mV, 100b-28.15mV, 101b-39.35mV, 110b-45mV, 111b-56.3mV, One byte for each port.

Usb2AfePredeemp[16]
USB Per Port HS Transmitter Emphasis. 00b - Emphasis OFF, 01b - De-emphasis ON, 10b - Pre-emphasis ON, 11b - Pre-emphasis & De-emphasis ON. One byte for each port.

Usb2AfePehalfbit[16]
USB Per Port Half Bit Pre-emphasis. 1b - half-bit pre-emphasis, 0b - full-bit pre-emphasis. One byte for each port.

Usb3HsioTxDeEmphEnable[16]
Enable the write to USB 3.0 TX Output -3.5dB De-Emphasis Adjustment. Each value in arrary can be between 0-1. One byte for each port.

Usb3HsioTxDeEmph[10]
USB 3.0 TX Output -3.5dB De-Emphasis Adjustment Setting, HSIO_TX_DWORD5[21:16], Default = 29h (approximately -3.5dB De-Emphasis). One byte for each port.

Usb3HsioTxDownscaleAmpEnable[10]
Enable the write to USB 3.0 TX Output Downscale Amplitude Adjustment, Each value in arrary can be between 0-1. One byte for each port.

Usb3HsioTxDownscaleAmp[10]
USB 3.0 TX Output Downscale Amplitude Adjustment, HSIO_TX_DWORD8[21:16], Default = 00h. One byte for each port.

PcieRpEnable[20]
Enable/disable PCIE Root Ports. 0: Disable, 1: Enable. One byte for each port, byte0 for port1, byte1 for port2, and so on

Default: All ports are enabled { 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01 }

PcieRpPmSci[20]
Indicate whether the root port power manager SCI is enabled - 0: disable, 1: enable. One byte for each port, byte0 for port1, byte1 for port2, and so on.

PcieRpClkReqSupport[20]
Enable/disable PCIE Root Port CLKREQ support. 0: Disable, 1: Enable. One byte for each port, byte0 for port1, byte1 for port2, and so on

Default:{ 0x01, 0x00, 0x00, 0x00, 0x01, 0x01, 0x00, 0x00, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }

PcieRpClkReqNumber[20]
Configure Root Port CLKREQ Number if CLKREQ is supported. Each value in array can be between 0-6. One byte for each port, byte0 for port1, byte1 for port2, and so on

Default: { 0x02, 0x00, 0x00, 0x00, 0x03, 0x01, 0x00, 0x00, 0x05, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }

EnableLan
Enable/disable LAN. Default : 1 (Enabled)

LanLtrEnable
Enable/Disable LTR capabilty of PCH internal LAN. Default : 0 (Disable)

eSATASpeedLimit
When enabled, BIOS will configure the PxSCTL.SPD to 2 to limit the eSATA port speed. 0: disable, 1: enable. Default: 0

SataRstRaid0
Enable/Disable RAID0. Default: 0

SataRstRaid1
Enable/Disable RAID1. Default: 0

SataRstRaid10
Enable/Disable RAID10. Default: 0

SataRstRaid5
Enable/Disable RAID5. Default: 0

SkipMpInit
When this is skipped, boot loader must initialize processors before SilicionInit API. 0: Initialize, 1: Skip. Default: 0

PcieRpHotPlug[20]
Enable/disable PCIE Root Ports HogPlug. 0: disable, 1: enable. One byte for each port, byte0 for port1, byte1 for port2, and so on.

Default: { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }

RpFunctionSwap
Enable/disable PCIE RP functi qon swap. 0: disable, 1: enable. It allows BIOS to use root port function number swapping when root port of function 0 is disabled. NOTE: This option will not work if ports 1, 9, 17 are fused or configured for RST PCIe storage. Disabling function swap may have adverse impact on power management. Default: 1

Usb2OverCurrentPin[16]
Configure over current pin assignment per USB2 ports. Refer to USB_OVERCURRENT_PIN. 0x08 means "skip over current pin". One byte for each port, byte0 for port0, byte1 for port1, and so on.

Default: { 0x00, 0x02, 0x08, 0x08, 0x02, 0x08, 0x08, 0x08, 0x01, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08}

Usb3OverCurrentPin[10]
Configure over current pin assignment per USB3 ports. Refer to USB_OVERCURRENT_PIN. 0x08 means "skip over current pin". One byte for each port, byte0 for port0, byte1 for port1, and so on.

Default: { 0x00, 0x08, 0x08, 0x01, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08}

Psi1Threshold[5]
Power State 1 current cuttof in 1/4 Amp increments. Range is 0-128A. Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced

Default: {0, 0, 0, 0, 0}

Psi2Threshold[5]
Power State 2 current cuttof in 1/4 Amp increments. Range is 0-128A. Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced

Default: {0, 0, 0, 0, 0}

Psi3Threshold[5]
State 3 current cuttof in 1/4 Amp increments. Range is 0-128A. Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

Psi3Enable[5]
Power State 3 0: Disable 1: Enable. Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

Psi4Enable[5]

Power State 4 0: Disable 1: Enable. Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

ImonSlope[5]
Imon slope correction. Specified in 1/100 increment values. Range is 0-200. 125 = 1.25. 0: Auto Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

ImonOffset[5]
Imon offset correction. Units 1/4, Range 0-255. Value of 100 = 100/4 = 25 offset. 0: Auto Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

IccMax[5]
VR Icc Max limit. 0-255A in 1/4 A units. 400 = 100A Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

VrVoltageLimit[5]
VR Icc Max limit. 0-255A in 1/4 A units. 400 = 100A Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

VrConfigEnable[5]
BIOS configuration of VR 0: Disable 1: Enable. Array index maps to VR 0 = System Agent, 1 = IA Core, 2 = Ring, 3 = GT unsliced, 4 = GT sliced.

Default: {0, 0, 0, 0, 0}

CpuS3ResumeHobData
CPU S3 Resume Hob Data. Default: 0

CpuS3ResumeMtrrData
Pointer CPU S3 Resume MTRR Data. Default: 0

CpuS3ResumeMtrrDataSize
Size of S3 resume MTRR data. Default: 0

LockDownConfigGlobalSmi
Enable SMI_LOCK bit to prevent writes to the Global SMI Enable bit. Value 0: Disable, 1: Enable.

LockDownConfigBiosInterface

Enable BIOS Interface Lock Down bit to prevent writes to the Backup Control Register. Top Swap bit and the General Control and Status Registers Boot BIOS Straps. Value 0: Disable, 1: Enable.

LockDownConfigBiosLock
When enabled, the BIOS Region can only be modified from SMM after EndOfDxe protocol is installed. Value 0: Disable, 1: Enable.

LockDownConfigSpiEiss
Enable InSMM.STS (EISS) in SPI If this bit is set, then WPD must be a '1' and InSMM.STS must be '1' also in order to write to BIOS regions of SPI Flash. If this bit is clear, then the InSMM.STS is a don't care. The BIOS must set the EISS bit while BIOS Guard support is enabled. Value 0: Clear EISS bit, 1: Set EISS bit.

PchConfigSubSystemVendorId
Subsystem Vendor ID of the PCH devices.

PchConfigSubSystemId
Subsystem ID of the PCH devices.

WakeConfigWolEnableOverride
Corresponds to the "WOL Enable Override" bit in the General PM Configuration B (GEN_PMCON_B) register. Value 0: Disable, 1: Enable.

WakeConfigPcieWakeFromDeepSx
Determine if enable PCIe to wake from deep Sx. Value 0: Disable, 1: Enable.

PmConfigDeepSxPol
Deep Sx Policy. Values 0: PchDeepSxPolDisable, 1: PchDpS5BatteryEn, 2: PchDpS5AlwaysEn, 3: PchDpS4S5BatteryEn, 4: PchDpS4S5AlwaysEn, 5: PchDpS3S4S5BatteryEn, 6: PchDpS3S4S5AlwaysEn.

PmConfigSlpS3MinAssert
SLP_S3 Minimum Assertion Width Policy. Values 0: PchSlpS360us, 1: PchSlpS31ms, 2: PchSlpS350ms, 3: PchSlpS32s.

PmConfigSlpS4MinAssert
SLP_S4 Minimum Assertion Width Policy. Values 0: PchSlpS4PchTime, 1: PchSlpS41s, 2: PchSlpS42s, 3: PchSlpS43s, 4: PchSlpS44s.

PmConfigSlpSusMinAssert
SLP_SUS Minimum Assertion Width Policy. Values 0: PchSlpSus0ms, 1: PchSlpSus500ms, 2: PchSlpSus1s, 3: PchSlpSus4s.

PmConfigSlpAMinAssert
SLP_A Minimum Assertion Width Policy. Values 0: PchSlpA0ms, 1: PchSlpA4s, 2: PchSlpA98ms, 3: PchSlpA2s.

PmConfigPciClockRun

This member describes whether or not the PCI ClockRun feature of PCH should be enabled. Values 0: Disabled, 1: Enabled

PmConfigSlpStrchSusUp
SLP_X Stretching After SUS Well Power Up. Values 0: Disabled, 1: Enabled

PmConfigPwrBtnOverridePeriod
PCH power button override period. Values: 0x0 - 4s, 0x1 - 6s, 0x2 - 8s, 0x3 - 10s, 0x4 - 12s, 0x5 - 14s.

PmConfigPwrCycDur
Reset Power Cycle Duration could be customized in the unit of second. PCH HW default is 4 seconds, and range is 1~4 seconds. Values: 0x0 - 0s, 0x1 - 1s, 0x2 - 2s, 0x3 - 3s, 0x4 - 4s.

SerialIrqConfigSirqEnable
Determines if enable Serial IRQ. Values 0: Disabled, 1: Enabled

SerialIrqConfigSirqMode
Serial IRQ Mode Select. Values: 0: PchQuietMode, 1: PchContinuousMode.

SerialIrqConfigStartFramePulse
Start Frame Pulse Width. Values: 0: PchSfpw4Clk, 1: PchSfpw6Clk, 2: PchSfpw8Clk.

PsfUnlock
The PSF registers will be locked before 3rd party code execution. This policy unlock the PSF space. NOTE: Do not set this policy "PsfUnlock" unless necessary.

SerialIoI2cVoltage
Selects the IO voltage for I2C controllers, 0: PchSerialIoIs33V, 1: PchSerialIoIs18V.

Early8254ClockGatingEnable
Set 8254CGE=1 is required for C11 support. However, set 8254CGE=1 in POST time might fail to boot legacy OS which using 8254 timer. Make sure it won't break legacy OS boot before enabling this.

SendVrMbxCmd
VR specific mailbox commands. 000b - no VR specific command sent. 001b - A VR mailbox command specifically for the MPS IMPV8 VR will be sent. 010b - VR specific command sent for PS4 exit issue. 100b - VR specific command sent for MPS VR decay issue.

PmonSlope
PCODE MMIO Mailbox: Platform Pmon slope correction. 0 - Auto Specified in 1/100 increment values. Range is 0-200. 125 = 1.25.

PmonOffset
PCODE MMIO Mailbox: Platform Pmon offset correction. 0 - Auto Units 1/4, Range 0-255. Value of 100 = 100/4 = 25 offset.

PsysPmax

PCODE MMIO Mailbox: Platform Power Pmax. 0 - Auto Specified in 1/8 Watt increments. Range 0-1024 Watts. Value of 800 = 100W.