



Cedar Island Intel® Firmware Support Package (FSP) Integration Guide

Tue Aug 17 2021 15:48:05

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below. You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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.

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.

Intel, Intel Atom, [include any Intel trademarks which are used in this document] and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

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

Copyright ©Intel Corporation. All rights reserved.

Contents

- 1 Introduction** **1**

- 2 Overview** **3**

- 3 FSP Integration** **5**

- 4 FSP Dispatch Mode** **9**
 - 4.1 Dispatch Mode Policy Init 10
 - 4.2 Dispatch Mode Policy Data Structures 10
 - 4.3 FSP Error Information 11
 - 4.4 Dispatch Mode Integration 11

- 5 FSP API Mode** **13**
 - 5.1 FSP APIs 13
 - 5.2 Reset Return Codes 16

- 6 Porting Recommendations** **17**

- 7 Todo List** **19**

- 8 Deprecated List** **21**

- 9 Class Index** **23**
 - 9.1 Class List 23

- 10 File Index** **29**
 - 10.1 File List 29

- 11 Class Documentation** **31**
 - 11.1 [_EFI_MM_RESERVED_MMRAM_REGION Struct Reference](#) 31
 - 11.1.1 Detailed Description 31
 - 11.1.2 Member Data Documentation 31
 - 11.1.2.1 MmramReservedSize 31
 - 11.1.2.2 MmramReservedStart 31
 - 11.2 [_EFI_PEI_MP_SERVICES_PPI Struct Reference](#) 32

11.2.1 Detailed Description	32
11.3 _LIST_ENTRY Struct Reference	32
11.3.1 Detailed Description	32
11.4 _MEMORY_POLICY_PPI Struct Reference	33
11.4.1 Detailed Description	33
11.5 _PCH_POLICY Struct Reference	33
11.5.1 Detailed Description	36
11.5.2 Member Data Documentation	36
11.5.2.1 IoApicConfig	36
11.5.2.2 Revision	36
11.5.2.3 SataConfig	41
11.5.2.4 TempMemBaseAddr	41
11.5.2.5 TempPciBusMin	41
11.6 _RAS_IMC_S3_DATA_PPI Struct Reference	41
11.6.1 Detailed Description	42
11.7 _UPI_POLICY_PPI Struct Reference	42
11.7.1 Detailed Description	42
11.7.2 Member Data Documentation	42
11.7.2.1 Revision	43
11.8 AdvMemTestRankData Union Reference	43
11.8.1 Detailed Description	43
11.9 ALL_LANES_EPARAM_LINK_INFO Struct Reference	43
11.9.1 Detailed Description	44
11.10commonSetup Struct Reference	44
11.10.1 Detailed Description	44
11.10.2 Member Data Documentation	45
11.10.2.1 ddrTxFactor	45
11.10.2.2 debugJumper	45
11.10.2.3 maxAddrMem	45
11.10.2.4 options	45
11.10.2.5 serialDebugMsgLvl	45
11.10.2.6 SocketConfig	46
11.11CPU_POLICY_HOB Struct Reference	46
11.11.1 Detailed Description	47
11.11.2 Member Data Documentation	48
11.11.2.1 flexRatioNext	48
11.12ddrChannelSetup Struct Reference	48
11.12.1 Detailed Description	48
11.13ddrDimmSetup Struct Reference	49
11.13.1 Detailed Description	49

11.14	ddrSocketSetup Struct Reference	49
11.14.1	Detailed Description	50
11.15	DMI_HW_WIDTH_CONTROL Struct Reference	50
11.15.1	Detailed Description	50
11.16	EFI_HOB_CPU Struct Reference	51
11.16.1	Detailed Description	51
11.16.2	Member Data Documentation	51
11.16.2.1	Header	51
11.17	EFI_HOB_FIRMWARE_VOLUME Struct Reference	52
11.17.1	Detailed Description	52
11.17.2	Member Data Documentation	52
11.17.2.1	Header	52
11.18	EFI_HOB_FIRMWARE_VOLUME2 Struct Reference	52
11.18.1	Detailed Description	53
11.18.2	Member Data Documentation	53
11.18.2.1	Header	53
11.19	EFI_HOB_FIRMWARE_VOLUME3 Struct Reference	54
11.19.1	Detailed Description	54
11.19.2	Member Data Documentation	54
11.19.2.1	ExtractedFv	54
11.19.2.2	FileName	55
11.19.2.3	FvName	55
11.19.2.4	Header	55
11.20	EFI_HOB_GENERIC_HEADER Struct Reference	55
11.20.1	Detailed Description	55
11.21	EFI_HOB_GUID_TYPE Struct Reference	56
11.21.1	Detailed Description	56
11.21.2	Member Data Documentation	56
11.21.2.1	Header	56
11.22	EFI_HOB_HANDOFF_INFO_TABLE Struct Reference	56
11.22.1	Detailed Description	57
11.22.2	Member Data Documentation	57
11.22.2.1	EfiMemoryTop	57
11.22.2.2	Header	58
11.22.2.3	Version	58
11.23	EFI_HOB_MEMORY_ALLOCATION Struct Reference	58
11.23.1	Detailed Description	58
11.23.2	Member Data Documentation	59
11.23.2.1	Header	59
11.24	EFI_HOB_MEMORY_ALLOCATION_BSP_STORE Struct Reference	59

11.24.1 Detailed Description	59
11.24.2 Member Data Documentation	60
11.24.2.1 Header	60
11.25EFI_HOB_MEMORY_ALLOCATION_HEADER Struct Reference	60
11.25.1 Detailed Description	61
11.25.2 Member Data Documentation	61
11.25.2.1 MemoryBaseAddress	61
11.25.2.2 MemoryType	61
11.25.2.3 Name	61
11.26EFI_HOB_MEMORY_ALLOCATION_MODULE Struct Reference	61
11.26.1 Detailed Description	62
11.26.2 Member Data Documentation	62
11.26.2.1 Header	62
11.27EFI_HOB_MEMORY_ALLOCATION_STACK Struct Reference	62
11.27.1 Detailed Description	63
11.27.2 Member Data Documentation	63
11.27.2.1 Header	63
11.28EFI_HOB_MEMORY_POOL Struct Reference	63
11.28.1 Detailed Description	64
11.28.2 Member Data Documentation	64
11.28.2.1 Header	64
11.29EFI_HOB_RESOURCE_DESCRIPTOR Struct Reference	64
11.29.1 Detailed Description	65
11.29.2 Member Data Documentation	65
11.29.2.1 Header	65
11.29.2.2 Owner	65
11.30EFI_HOB_UEFI_CAPSULE Struct Reference	66
11.30.1 Detailed Description	66
11.30.2 Member Data Documentation	66
11.30.2.1 BaseAddress	66
11.31EFI_IP_ADDRESS Union Reference	67
11.31.1 Detailed Description	67
11.32EFI_MAC_ADDRESS Struct Reference	67
11.32.1 Detailed Description	67
11.33EFI_MMRAM_DESCRIPTOR Struct Reference	68
11.33.1 Detailed Description	68
11.33.2 Member Data Documentation	68
11.33.2.1 CpuStart	68
11.33.2.2 PhysicalStart	68
11.33.2.3 RegionState	68

11.34EFI_PEI_HOB_POINTERS Union Reference	69
11.34.1 Detailed Description	69
11.35EFI_TIME Struct Reference	69
11.35.1 Detailed Description	69
11.36FSPM_CONFIG Struct Reference	70
11.36.1 Detailed Description	74
11.36.2 Member Data Documentation	74
11.36.2.1 BoardTypeBitmask	74
11.36.2.2 BusRatio	75
11.36.2.3 D2KCreditConfig	75
11.36.2.4 DdrtQosMode	75
11.36.2.5 DebugPrintLevel	75
11.36.2.6 DeEmphasis	75
11.36.2.7 Degrade4SPreference	75
11.36.2.8 DegradePrecedence	75
11.36.2.9 DfxDnTxPreset	75
11.36.2.10DfxRxPreset	76
11.36.2.11DfxUpTxPreset	76
11.36.2.12IOPcieMaxPayload	76
11.36.2.13IOPciePortLinkSpeed	76
11.36.2.14oDcMode	76
11.36.2.15rqThreshold	76
11.36.2.16sKtiNvramDataReady	76
11.36.2.17KtiCpuSktHotPlugTopology	77
11.36.2.18KtiCrcMode	77
11.36.2.19KtiFailoverEn	77
11.36.2.20KtiLinkL0pEn	77
11.36.2.21KtiLinkL1En	77
11.36.2.22KtiLinkSpeed	77
11.36.2.23KtiLinkSpeedMode	77
11.36.2.24LegacyVgaSoc	77
11.36.2.25LegacyVgaStack	78
11.36.2.26MbeBwCal	78
11.36.2.27mmCfgBase	78
11.36.2.28mmCfgSize	78
11.36.2.29mmiohBase	78
11.36.2.30mmiohSize	78
11.36.2.31NtbBarSizeEmBarSZ1	78
11.36.2.32NtbBarSizeEmBarSZ2	79
11.36.2.33NtbBarSizeEmBarSZ2_0	79

11.36.2.34	NtbBarSizeEmBarSZ2_1	79
11.36.2.35	NtbBarSizeImBar1	79
11.36.2.36	NtbBarSizeImBar2	79
11.36.2.37	NtbBarSizeImBar2_0	79
11.36.2.38	NtbBarSizeImBar2_1	79
11.36.2.39	NtbPpd	80
11.36.2.40	NtbXlinkCtlOverride	80
11.36.2.41	PchPciePIISsc	80
11.36.2.42	PchSirqMode	80
11.36.2.43	PcieCommonClock	80
11.36.2.44	SerialIoUartDebugEnable	80
11.36.2.45	SerialIoUartDebugIoBase	80
11.36.2.46	SnoopAllCores	81
11.36.2.47	SnoopThrottleConfig	81
11.36.2.48	SplitLock	81
11.36.2.49	StaleAtoSOptEn	81
11.36.2.50	ThermalDeviceEnable	81
11.36.2.51	TorThresLoctoremEmpty	81
11.36.2.52	TorThresLoctoremNorm	81
11.36.2.53	TscSyncEn	81
11.36.2.54	UmaClustering	82
11.36.2.55	WaitTimeForPSBP	82
11.36.2.56	XptPrefetchEn	82
11.37	FSPM_UPD Struct Reference	82
11.37.1	Detailed Description	83
11.38	FSPTS_CONFIG Struct Reference	83
11.38.1	Detailed Description	84
11.38.2	Member Data Documentation	84
11.38.2.1	BifurcationPcie0	84
11.38.2.2	BifurcationPcie1	84
11.38.2.3	EnableGbE	84
11.39	FSPTS_UPD Struct Reference	85
11.39.1	Detailed Description	85
11.40	FSPTS_CONFIG Struct Reference	85
11.40.1	Detailed Description	86
11.41	FSPTS_CORE_UPD Struct Reference	86
11.41.1	Detailed Description	86
11.42	FSPTS_UPD Struct Reference	86
11.42.1	Detailed Description	87
11.43	GUID Struct Reference	87

11.43.1 Detailed Description	87
11.44 IPv4_ADDRESS Struct Reference	88
11.44.1 Detailed Description	88
11.45 IPv6_ADDRESS Struct Reference	88
11.45.1 Detailed Description	88
11.46 KTI_HOST_IN Struct Reference	88
11.46.1 Detailed Description	91
11.46.2 Member Data Documentation	91
11.46.2.1 BoardTypeBitmask	91
11.46.2.2 BusRatio	92
11.46.2.3 CFRImagePtr	92
11.46.2.4 ColdResetRequestEnd	92
11.46.2.5 ColdResetRequestStart	92
11.46.2.6 highGap	92
11.46.2.7 lowGap	93
11.46.2.8 OemCheckCpuPartsChangeSwap	93
11.46.2.9 OemGetAdaptedEqSettings	93
11.46.2.10 SplitLock	93
11.47 memSetup Struct Reference	93
11.47.1 Detailed Description	100
11.47.2 Member Data Documentation	101
11.47.2.1 ADRDataSaveMode	101
11.47.2.2 ADREn	101
11.47.2.3 AdvMemTestCondPause	101
11.47.2.4 AdvMemTestCondTrefi	101
11.47.2.5 AdvMemTestCondTwr	101
11.47.2.6 AdvMemTestRankListNumEntries	101
11.47.2.7 AdvMemTestResetList	102
11.47.2.8 AepNotSupportedException	102
11.47.2.9 ApdEn	102
11.47.2.10 AppDirectMemoryHole	102
11.47.2.11 Blockgnt2cmd1cyc	102
11.47.2.12 CacheMemType	102
11.47.2.13 check_platform_detect	103
11.47.2.14 check_pm_sts	103
11.47.2.15 chInter	103
11.47.2.16 CkIdleTimer	103
11.47.2.17 CkeProgramming	103
11.47.2.18 CkeThrottling	104
11.47.2.19 CkMode	104

11.47.2.20cmdSetupPercentOffset	104
11.47.2.21CmiInitOption	104
11.47.2.22CmsEnableDramPm	104
11.47.2.23DataBufferDfe	105
11.47.2.24DataDIIOff	105
11.47.2.25DcpmmAveragePowerLimit	105
11.47.2.26DcpmmAveragePowerTimeConstant	105
11.47.2.27DcpmmMbbAveragePowerTimeConstant	105
11.47.2.28DcpmmMbbFeature	105
11.47.2.29DcpmmMbbMaxPowerLimit	106
11.47.2.30DdrCacheSize	106
11.47.2.31ddrFreqLimit	106
11.47.2.32DdrCkeEn	106
11.47.2.33dimmTypeSupport	106
11.47.2.34DisableDirForAppDirect	106
11.47.2.35Disddrtopprd	107
11.47.2.36DramRaplEnable	107
11.47.2.37dramraplRefreshBase	107
11.47.2.38EliminateDirectoryInFarMemory	107
11.47.2.39EnforcePopulationPor	108
11.47.2.40enforcePOR	108
11.47.2.41ExtendedADDDCEn	108
11.47.2.42ExtendedType17	108
11.47.2.43FastGoConfig	108
11.47.2.44ForcePxclnit	109
11.47.2.45mcBclk	109
11.47.2.46LatchSystemShutdownState	109
11.47.2.47LegacyADRMdEn	109
11.47.2.48LsxImplementation	110
11.47.2.49MdIIOffEn	110
11.47.2.50memFlows	110
11.47.2.51memFlowsExt	111
11.47.2.52MemHotOuputAssertThreshold	111
11.47.2.53MemoryConnectorType	112
11.47.2.54MemoryTopology	112
11.47.2.55MinNormalMemSize	112
11.47.2.56NfitPublishMailboxStructsDisable	112
11.47.2.57normOppIntvl	112
11.47.2.58NvDimmEnergyPolicy	113
11.47.2.59NvdimmSmbusMaxAccessTime	113

11.47.2.60NvdimmSmbusReleaseDelay	113
11.47.2.61NvmdimmPerfConfig	113
11.47.2.62NvmdimmPowerCyclePolicy	113
11.47.2.63NvmMediaStatusException	113
11.47.2.64NvmQos	114
11.47.2.65oltPeakBWLIMITPercent	114
11.47.2.66OppSrefEn	114
11.47.2.67options	114
11.47.2.68optionsExt	115
11.47.2.69optionsNgn	116
11.47.2.70PanicWm	116
11.47.2.71partialmirrorpercent	116
11.47.2.72partialmirrorsad0	117
11.47.2.73partialmirrorsts	117
11.47.2.74partialMirrorUEFI	117
11.47.2.75patrolScrubAddrMode	117
11.47.2.76PdaModeX16	118
11.47.2.77PeriodicRcomp	118
11.47.2.78PeriodicRcompInterval	118
11.47.2.79PkgcSrefEn	118
11.47.2.80PpdEn	118
11.47.2.81pprAddrSetup	119
11.47.2.82pprType	119
11.47.2.83readPreamble	119
11.47.2.84RxDfeEn	119
11.47.2.85setSecureEraseAllDIMMs	119
11.47.2.86setSecureEraseSktCh	120
11.47.2.87smartTestKey	120
11.47.2.88spareErrTh	120
11.47.2.89SpdPrintEn	120
11.47.2.90SpdPrintLength	120
11.47.2.91SpdSmbSpeed	121
11.47.2.92SrefProgramming	121
11.47.2.93TempRefreshOption	121
11.47.2.94thermalThrottlingOptions	121
11.47.2.95ThrottlingMidOnTempLo	121
11.47.2.96TrainingCompOptions	122
11.47.2.97trainingResultOffsetFunctionEnable	122
11.47.2.98TxRiseFallSlewRate	122
11.47.2.99UseSmbusForMrwEarly	122

11.47.2.100	VirtualNumaEnable	122
11.47.2.101	BlMemMode	123
11.47.2.102	WritePreamble	123
11.48	memTiming Struct Reference	123
11.48.1	Detailed Description	124
11.48.2	Member Data Documentation	124
11.48.2.1	nCL	124
11.48.2.2	nCMDRate	124
11.48.2.3	nFAW	125
11.48.2.4	nRAS	125
11.48.2.5	nRC	125
11.48.2.6	nRCD	125
11.48.2.7	nRFC	125
11.48.2.8	nRP	125
11.48.2.9	nRRD	125
11.48.2.10	nRTP	126
11.48.2.11	nWR	126
11.48.2.12	nWTR	126
11.49	PCH_DCI_CONFIG Struct Reference	126
11.49.1	Detailed Description	126
11.49.2	Member Data Documentation	127
11.49.2.1	DciAutoDetect	127
11.49.2.2	DciEn	127
11.50	PCH_DEVICE_INTERRUPT_CONFIG Struct Reference	127
11.50.1	Detailed Description	127
11.51	PCH_DMI_CONFIG Struct Reference	127
11.51.1	Detailed Description	128
11.51.2	Member Data Documentation	128
11.51.2.1	DmiAspm	128
11.52	PCH_FLASH_PROTECTION_CONFIG Struct Reference	128
11.52.1	Detailed Description	129
11.53	PCH_GBL2HOST_EN Union Reference	129
11.53.1	Detailed Description	129
11.54	PCH_GENERAL_CONFIG Struct Reference	129
11.54.1	Detailed Description	129
11.54.2	Member Data Documentation	130
11.54.2.1	Crid	130
11.54.2.2	SubSystemVendorId	130
11.55	PCH_HDAUDIO_CONFIG Struct Reference	130
11.55.1	Detailed Description	131

11.55.2 Member Data Documentation	131
11.55.2.1 DspEndpointDmic	131
11.55.2.2 DspPpModuleMask	131
11.55.2.3 Enable	131
11.56PCH_HPET_CONFIG Struct Reference	132
11.56.1 Detailed Description	132
11.56.2 Member Data Documentation	132
11.56.2.1 Enable	132
11.57PCH_HSIO_PCIE_CONFIG Struct Reference	132
11.57.1 Detailed Description	133
11.58PCH_HSIO_PCIE_LANE_CONFIG Struct Reference	133
11.58.1 Detailed Description	134
11.58.2 Member Data Documentation	135
11.58.2.1 RsvdBits2	135
11.58.2.2 RsvdBits3	135
11.59PCH_HSIO_PCIE_WM20_CONFIG Struct Reference	135
11.59.1 Detailed Description	135
11.60PCH_HSIO_SATA_CONFIG Struct Reference	136
11.60.1 Detailed Description	136
11.61PCH_HSIO_SATA_PORT_LANE Struct Reference	136
11.61.1 Detailed Description	138
11.62PCH_INTERRUPT_CONFIG Struct Reference	138
11.62.1 Detailed Description	139
11.63PCH_IOAPIC_CONFIG Struct Reference	139
11.63.1 Detailed Description	139
11.64PCH_LAN_CONFIG Struct Reference	140
11.64.1 Detailed Description	140
11.64.2 Member Data Documentation	140
11.64.2.1 Enable	140
11.65PCH_LOCK_DOWN_CONFIG Struct Reference	140
11.65.1 Detailed Description	141
11.65.2 Member Data Documentation	141
11.65.2.1 BiosInterface	141
11.65.2.2 BiosLock	141
11.65.2.3 GlobalSmi	142
11.65.2.4 GpioLockDown	142
11.65.2.5 RtcLock	142
11.65.2.6 SpiEiss	142
11.65.2.7 TcoLock	142
11.66PCH_LPC_CONFIG Struct Reference	142

11.66.1 Detailed Description	143
11.66.2 Member Data Documentation	143
11.66.2.1 EnhancePort8xhDecoding	143
11.67PCH_LPC_SIRQ_CONFIG Struct Reference	143
11.67.1 Detailed Description	143
11.68PCH_MEMORY_THROTTLING Struct Reference	144
11.68.1 Detailed Description	144
11.68.2 Member Data Documentation	144
11.68.2.1 Enable	144
11.68.2.2 TsGpioPinSetting	144
11.69PCH_P2SB_CONFIG Struct Reference	145
11.69.1 Detailed Description	145
11.69.2 Member Data Documentation	145
11.69.2.1 P2SbReveal	145
11.69.2.2 PsfUnlock	145
11.69.2.3 SbiUnlock	145
11.70PCH_PCIE_CONFIG Struct Reference	145
11.70.1 Detailed Description	147
11.70.2 Member Data Documentation	147
11.70.2.1 AllowNoLtrIccPIIShutdown	147
11.70.2.2 ComplianceTestMode	147
11.70.2.3 DetectTimeoutMs	147
11.70.2.4 DisableRootPortClockGating	147
11.70.2.5 EnablePeerMemoryWrite	148
11.70.2.6 EnablePort8xhDecode	148
11.70.2.7 EqPh3LaneParam	148
11.70.2.8 RpFunctionSwap	148
11.71PCH_PCIE_CONFIG2 Struct Reference	148
11.71.1 Detailed Description	149
11.72PCH_PCIE_EQ_LANE_PARAM Struct Reference	149
11.72.1 Detailed Description	149
11.73PCH_PCIE_ROOT_PORT_CONFIG Struct Reference	150
11.73.1 Detailed Description	152
11.73.2 Member Data Documentation	152
11.73.2.1 ClkReqDetect	152
11.73.2.2 ClkReqNumber	152
11.73.2.3 DeviceResetPad	152
11.73.2.4 Gen3EqPh3Method	152
11.73.2.5 HsioRxSetCtle	152
11.73.2.6 HsioRxSetCtleEnable	153

11.73.2.7 PcieSpeed	153
11.73.2.8 SlotImplemented	153
11.74 PCH_PM_CONFIG Struct Reference	153
11.74.1 Detailed Description	154
11.74.2 Member Data Documentation	155
11.74.2.1 CapsuleResetType	155
11.74.2.2 DisableDsxAcPresentPulldown	155
11.74.2.3 DisableEnergyReport	155
11.74.2.4 DisableNativePowerButton	155
11.74.2.5 PchPwrCycDur	155
11.74.2.6 PciClockRun	156
11.74.2.7 PciePIISsc	156
11.74.2.8 PmcReadDisable	156
11.74.2.9 PowerResetStatusClear	156
11.74.2.10 PwrBtnOverridePeriod	156
11.74.2.11 RsvdBits0	156
11.74.2.12 SlpLanLowDc	156
11.74.2.13 SlpS0Enable	157
11.75 PCH_PORT61H_SMM_CONFIG Struct Reference	157
11.75.1 Detailed Description	157
11.76 PCH_POWER_RESET_STATUS Struct Reference	157
11.76.1 Detailed Description	158
11.77 PCH_RST_PCIE_STORAGE_CONFIG Struct Reference	158
11.77.1 Detailed Description	158
11.77.2 Member Data Documentation	159
11.77.2.1 DeviceResetDelay	159
11.77.2.2 Enable	159
11.77.2.3 RstPcieStoragePort	159
11.78 PCH_SATA_CONFIG Struct Reference	159
11.78.1 Detailed Description	160
11.78.2 Member Data Documentation	160
11.78.2.1 Enable	160
11.78.2.2 eSATA SpeedLimit	160
11.78.2.3 SataMode	161
11.79 PCH_SATA_PORT_CONFIG Struct Reference	161
11.79.1 Detailed Description	162
11.79.2 Member Data Documentation	162
11.79.2.1 Enable	162
11.79.2.2 HsioRxEqBoostMagAd	162
11.79.2.3 HsioRxEqBoostMagAdEnable	162

11.79.2.4 HsioTxGen1DownscaleAmp	162
11.79.2.5 HsioTxGen1DownscaleAmpEnable	162
11.79.2.6 HsioTxGen2DownscaleAmp	162
11.79.2.7 HsioTxGen2DownscaleAmpEnable	162
11.79.2.8 ZpOdd	163
11.80PCH_SATA_RST_CONFIG Struct Reference	163
11.80.1 Detailed Description	164
11.81PCH_SKYCAM_CIO2_FLS_CONFIG Struct Reference	164
11.81.1 Detailed Description	165
11.82PCH_SMBUS_CONFIG Struct Reference	165
11.82.1 Detailed Description	165
11.82.2 Member Data Documentation	165
11.82.2.1 Enable	165
11.83PCH_SPI_CONFIG Struct Reference	166
11.83.1 Detailed Description	166
11.83.2 Member Data Documentation	166
11.83.2.1 ShowSpiController	166
11.84PCH_SSIC_CONFIG Struct Reference	166
11.84.1 Detailed Description	167
11.85PCH_THERMAL_CONFIG Struct Reference	167
11.85.1 Detailed Description	168
11.85.2 Member Data Documentation	168
11.85.2.1 PchHotLevel	168
11.85.2.2 ThermalDeviceEnable	168
11.85.2.3 ThermalThrottling	168
11.86PCH_THERMAL_THROTTLING Struct Reference	168
11.86.1 Detailed Description	169
11.87PCH_TRACE_HUB_CONFIG Struct Reference	169
11.87.1 Detailed Description	169
11.88PCH_USB20_PORT_CONFIG Struct Reference	169
11.88.1 Detailed Description	170
11.88.2 Member Data Documentation	170
11.88.2.1 OverCurrentPin	170
11.89PCH_USB30_PORT_CONFIG Struct Reference	170
11.89.1 Detailed Description	171
11.89.2 Member Data Documentation	171
11.89.2.1 OverCurrentPin	171
11.90PCH_USB_CONFIG Struct Reference	171
11.90.1 Detailed Description	172
11.90.2 Member Data Documentation	172

11.90.2.1 DisableComplianceMode	172
11.90.2.2 PortUsb20	172
11.90.2.3 UsbPrecondition	172
11.91 PCH_WAKE_CONFIG Struct Reference	173
11.91.1 Detailed Description	173
11.91.2 Member Data Documentation	173
11.91.2.1 Gp27WakeFromDeepSx	173
11.91.2.2 PmeB0S5Dis	174
11.92 PCH_WDT_CONFIG Struct Reference	174
11.92.1 Detailed Description	174
11.93 PCH_XDCI_CONFIG Struct Reference	174
11.93.1 Detailed Description	174
11.93.2 Member Data Documentation	175
11.93.2.1 Enable	175
11.94 PCH_XHCI_SSIC_PORT Struct Reference	175
11.94.1 Detailed Description	175
11.95 PER_LANE_EPARAM_LINK_INFO Struct Reference	175
11.95.1 Detailed Description	176
11.96 PPR_ADDR Struct Reference	176
11.96.1 Detailed Description	176
11.97 PPR_ADDR_MRC_SETUP Struct Reference	176
11.97.1 Detailed Description	176
11.98 PROTECTED_RANGE Struct Reference	177
11.98.1 Detailed Description	177
11.99 PSMI_POLICY_DATA_HOB Struct Reference	177
11.99.1 Detailed Description	178
11.100 RAS_RC_POLICY_PPI Struct Reference	178
11.100.1 Detailed Description	179
11.100.2 Member Data Documentation	179
11.100.2.1 CrashLogClear	179
11.100.2.2 CrashLogReArm	179
11.101 SATA_THERMAL_THROTTLE Struct Reference	179
11.101.1 Detailed Description	180
11.102 SECURITY_POLICY Struct Reference	180
11.102.1 Detailed Description	181
11.102.2 Member Data Documentation	181
11.102.2.1 SgxDebugMode	181
11.102.2.2 SgxSinitDataFromTpm	181
11.102.2.3 SgxSinitNvsData	182
11.103 SysSetup Struct Reference	182

11.103.1	Detailed Description	183
11.103.2	Member Data Documentation	183
11.103.2.1	WFRWAEEnable	183
11.104	THERMAL_THROTTLE_LEVELS Struct Reference	183
11.104.1	Detailed Description	184
11.104.2	Member Data Documentation	184
11.104.2.1	PchCrossThrottling	184
11.104.2.2	TTLock	184
11.104.2.3	TState13Enable	184
11.105	RACE_INFO Struct Reference	184
11.105.1	Detailed Description	185
11.106	S_GPIO_PIN_SETTING Struct Reference	185
11.106.1	Detailed Description	185
11.106.2	Member Data Documentation	185
11.106.2.1	PmsyncEnable	185
12	File Documentation	187
12.1	Base.h File Reference	187
12.1.1	Detailed Description	191
12.1.2	Macro Definition Documentation	192
12.1.2.1	_BASE_INT_SIZE_OF	192
12.1.2.2	_INT_SIZE_OF	192
12.1.2.3	ABS	192
12.1.2.4	ALIGN_POINTER	192
12.1.2.5	ALIGN_VALUE	193
12.1.2.6	ALIGN_VARIABLE	193
12.1.2.7	ANALYZER_NORETURN	193
12.1.2.8	ANALYZER_UNREACHABLE	193
12.1.2.9	ARRAY_SIZE	194
12.1.2.10	BASE_ARG	195
12.1.2.11	BASE_CR	195
12.1.2.12	ENCODE_ERROR	195
12.1.2.13	ENCODE_WARNING	196
12.1.2.14	FALSE	196
12.1.2.15	MAX	196
12.1.2.16	MIN	196
12.1.2.17	NORETURN	197
12.1.2.18	OFFSET_OF	197
12.1.2.19	RETURN_ADDRESS	197
12.1.2.20	RETURN_BUFFER_TOO_SMALL	197

12.1.2.21 RETURN_ERROR	198
12.1.2.22 RETURNS_TWICE	198
12.1.2.23 SIGNATURE_16	198
12.1.2.24 SIGNATURE_32	198
12.1.2.25 SIGNATURE_64	199
12.1.2.26 STATIC_ASSERT	199
12.1.2.27 TRUE	199
12.1.2.28 UNREACHABLE	199
12.1.2.29 VA_ARG	199
12.1.2.30 VA_COPY	200
12.1.2.31 VA_END	200
12.1.2.32 VA_START	200
12.1.3 Typedef Documentation	201
12.1.3.1 BASE_LIST	201
12.1.3.2 VA_LIST	201
12.2 CpuPolicyHob.h File Reference	201
12.2.1 Detailed Description	202
12.3 FspFixedPcds.h File Reference	202
12.3.1 Detailed Description	202
12.4 FspmUpd.h File Reference	202
12.4.1 Detailed Description	203
12.5 FspUpd.h File Reference	204
12.5.1 Detailed Description	204
12.6 FsptUpd.h File Reference	205
12.6.1 Detailed Description	205
12.7 FspUpd.h File Reference	206
12.7.1 Detailed Description	206
12.8 KtiHost.h File Reference	207
12.8.1 Detailed Description	207
12.9 MemoryPolicyPpi.h File Reference	208
12.9.1 Detailed Description	209
12.10 MpServices.h File Reference	209
12.10.1 Detailed Description	210
12.10.2 Typedef Documentation	210
12.10.2.1 EFI_PEI_MP_SERVICES_ENABLEDISABLEAP	210
12.10.2.2 EFI_PEI_MP_SERVICES_GET_NUMBER_OF_PROCESSORS	211
12.10.2.3 EFI_PEI_MP_SERVICES_GET_PROCESSOR_INFO	211
12.10.2.4 EFI_PEI_MP_SERVICES_STARTUP_ALL_APS	212
12.10.2.5 EFI_PEI_MP_SERVICES_STARTUP_THIS_AP	212
12.10.2.6 EFI_PEI_MP_SERVICES_SWITCH_BSP	213

12.10.2.7 EFI_PEI_MP_SERVICES_WHOAMI	214
12.11 PchPolicyCommon.h File Reference	214
12.11.1 Detailed Description	217
12.11.2 Enumeration Type Documentation	217
12.11.2.1 PCH_HDAUDIO_IO_BUFFER_OWNERSHIP	217
12.11.2.2 PCH_INT_PIN	218
12.11.2.3 PCH_PCIE_EQ_METHOD	218
12.11.2.4 PCH_RESERVED_PAGE_ROUTE	218
12.11.2.5 PCH_SLP_S4_MIN_ASSERT	218
12.11.2.6 PCH_USB_PORT_LOCATION	218
12.12 PiHob.h File Reference	218
12.12.1 Detailed Description	220
12.13 PiMultiPhase.h File Reference	220
12.13.1 Detailed Description	221
12.13.2 Macro Definition Documentation	221
12.13.2.1 DXE_ERROR	221
12.13.2.2 EFI_AUTH_STATUS_PLATFORM_OVERRIDE	222
12.13.2.3 PI_ENCODE_ERROR	222
12.13.2.4 PI_ENCODE_WARNING	222
12.13.3 Typedef Documentation	222
12.13.3.1 EFI_AP_PROCEDURE	222
12.13.3.2 EFI_AP_PROCEDURE2	222
12.14 PsmiPolicyHob.h File Reference	223
12.14.1 Detailed Description	223
12.15 RasImcS3Data.h File Reference	223
12.15.1 Detailed Description	224
12.15.2 Typedef Documentation	224
12.15.2.1 RAS_IMC_S3_DATA_PPI_GET_IMC_S3_RAS_DATA	224
12.16 RasRcPolicyPpi.h File Reference	226
12.16.1 Detailed Description	226
12.17 SecurityPolicy.h File Reference	226
12.17.1 Detailed Description	227
12.18 UefiBaseType.h File Reference	227
12.18.1 Detailed Description	230
12.18.2 Macro Definition Documentation	230
12.18.2.1 EFI_PAGES_TO_SIZE	230
12.18.2.2 EFI_SIZE_TO_PAGES	230
12.18.3 Typedef Documentation	230
12.18.3.1 EFI_IPv4_ADDRESS	230
12.18.3.2 EFI_IPv6_ADDRESS	230

12.19UpiPolicyPpi.h File Reference	231
12.19.1 Detailed Description	231

Index	233
--------------	------------

Chapter 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 **Cedar Island** platforms with the **CopperLake** processor and **Lewisburg** 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 v1.7* located at <http://www.uefi.org/specifications>
- *Intel® Firmware Support Package: External Architecture Specification (EAS) v2.2* located at <https://cdrdv2.intel.com/v1/dl/getContent/627153>
- *Boot Setting File Specification (BSF) v1.0* https://firmware.intel.com/sites/default/files/BSF_1_0.pdf
- *Binary Configuration Tool for Intel® Firmware Support Package* available at <http://www.intel.com/fsp>

1.4 Acronyms and Terminology

Acronym	Definition
BCT	Binary Configuration Tool
BDSM	Base Data Of Stolen Memory
BSF	Boot Setting File
BSP	Boot Strap Processor
BWG	BIOS Writer's Guide

CAR	Cache As Ram
CRB	Customer Reference Board
DPR	DMA Protected Range
FIT	Firmware Interface Table
FSP	Firmware Support Package
FSP API	Firmware Support Package Interface
FW	Firmware
GTT	Graphics Translation Table
IED	Intel Enhanced Debug
IFWI	Integrated Firmware Image
IOT	Internal Observation Trace
MRC	Memory Reference Code (Memory Init code encapsulated by FSP-M)
MOT	Memory Observation Trace
PCH	Platform Controller Hub
PMC	Power Management Controller
PMRR	Protected Memory Range Reporting
REMAP	Remapped Memory Area
RVP	Reference and Validation Platform
SBSP	System BSP
SMI	System Management Interrupt
SMM	System Management Mode
SMRAM	System Management Mode RAM
SPI	Serial Peripheral Interface
TOLUD	Top of Low Usable Memory
TOUUD	Top of Upper Usable Memory
TSEG	Memory Reserved at the Top of Memory to be used as SMRAM
UPD	Updatable Product Data

Chapter 2

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 into many different boot loaders, such as coreboot, EDKII MinPlatform, Intel® Slim Bootloader, etc. and also into an 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 tools provided by Intel. This step is optional as configuration data may also be provided at runtime.
- **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 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 Integration Guide
 - BSF Configuration File
 - Data Structure Header Files
- The FSP configuration utility called BCT is available as a separate package. It can be downloaded from link mentioned in [Section 1.3](#).

2.2.1 Package Layout

- **Docs (Auto generated)**
 - CedarIsland_FSP_Integration_Guide.pdf
 - CedarIsland_FSP_Integration_Guide.chm
 - **Include**
 - [FspUpd.h](#), [FspmUpd.h](#) and [FspSUpd.h](#) (FSP UPD structure and related definitions)
 - **Library**
 - FspPcdListLibNull (Empty library class used to build the boot loader PCD database for FSP dispatch mode)
 - **UefiDrivers**
 - FvLateSilicon.FV
 - Additional UEFI PI drivers for use by EDK II based boot loaders
 - FspBinPkg.dec (EDKII declaration file for package)
 - DynamicExPcd.dsc (List of PCDs used by the FSP in dispatch mode)
 - Fsp.bsf (BSF file for configuring the data using BCT tool)
 - Fsp.fd (FSP Binary)
-

Chapter 3

FSP Integration

3.1 Assumptions Used in this Document

The FSP for this platform is built with a preferred base address given by [PcdFspAreaBaseAddress](#) 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) or `SplitFsp↔Bin.py` before integrating to the boot loader.

For other assumptions and conventions, please refer to Chapter 8 and 9 of the FSP External Architecture Specification version 2.2.

3.2 Boot Flow

Please refer Chapter 7 in the FSP External Architecture Specification version 2.2 for Boot flow chart. The FSP for this platform supports dispatch mode, see Chapter 7 and 9 of the FSP External Architecture Specification version 2.2 for a description of dispatch mode.

3.3 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 2.2 with a HeaderRevision of 5.

3.4 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 AP↔I parameters could change over different FSP IDs and revisions. All the FSP FV segments (FSP-T, FSP-M and FSP-S) must have same FSP Image ID and revision number, using FV segments with different revision numbers in a single FSP image is not valid. 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 **\$CPX-SP\$** and the ImageRevision field is **0x02020033 (2.2.0.33)**.

3.5 FSP Global Data

FSP uses some amount of TempRam area to store FSP global data which contains some critical data like pointers to FSP information headers and UPD configuration regions, FSP/Bootloader stack pointers required for stack switching etc. HPET Timer register(2) [PcdGlobalDataPointerAddress](#) is reserved to store address of this global data, and hence boot loader should not use this register for any other purpose. If TempRAM initialization is done by boot loader, then HPET has to be initialized to the base so that access to the register will work fine.

3.6 Additional FSP Temp RAM Usage

The FSP-M for this platform reserves a region of TempRam for its exclusive use during the pre-memory phase. This region starts at physical address 0xFE800000 and ends at 0xFE92FFFF. These addresses are hardcoded and cannot be changed by the boot loader. The boot loader must configure TempRam such that these addresses fall within the range of temporary memory and are available for use by the FSP. If the boot loader installs a page table, then this region of memory must be identity mapped during the pre-memory phase. After FSP-M is complete and main memory is available, this region is no longer used and can be safely allocated for other purposes.

This region is used both in API mode and Dispatch mode. In API mode, this region is in addition to the region provided by `FSPM_ARCH_UPD.StackBase` and `FSPM_ARCH_UPD.StackSize`.

3.7 FSP CMOS Usage

The FSP for this platform uses 3 bytes of CMOS memory for internal data storage. The boot loader and the operating system must not write to these locations in CMOS to ensure that any data written by the FSP is retained. These 3 bytes are located at the following offsets in CMOS memory:

- 0x2A
- 0x46
- 0x47

3.8 FvLateSilicon

An additional FvLateSilicon firmware volume (FV) is provided for use with UEFI PI (aka EDK II) based boot loaders. This FV is located in the UefiDrivers subdirectory of the FSP Distribution Package. FvLateSilicon contains several DXE and SMM drivers that run later in the boot flow and provide additional silicon initialization to UEFI PI boot loaders which is not present in the FSP. Several features of the Xeon Scalable processor require the execution of the drivers contained in this FV and are therefore not available to non UEFI PI bootloaders. It is recommended that UEFI PI boot loaders use FSP Dispatch mode since it allows the policy options for these added features to be set to values other than their default values. The policy options for these features are available in the policy data structures exposed to FSP Dispatch mode, and are not exposed via FSP UPDs.

3.9 Memory Map

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

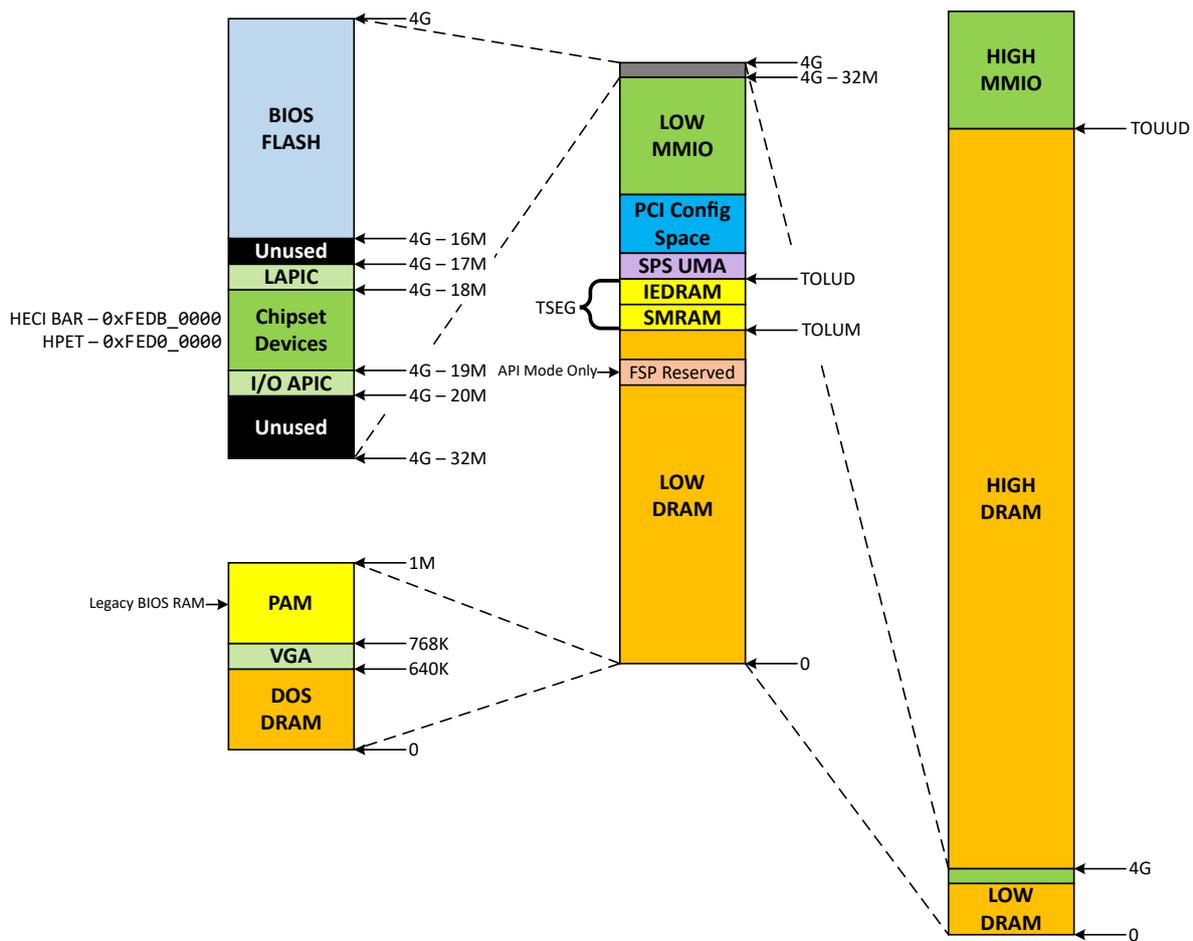


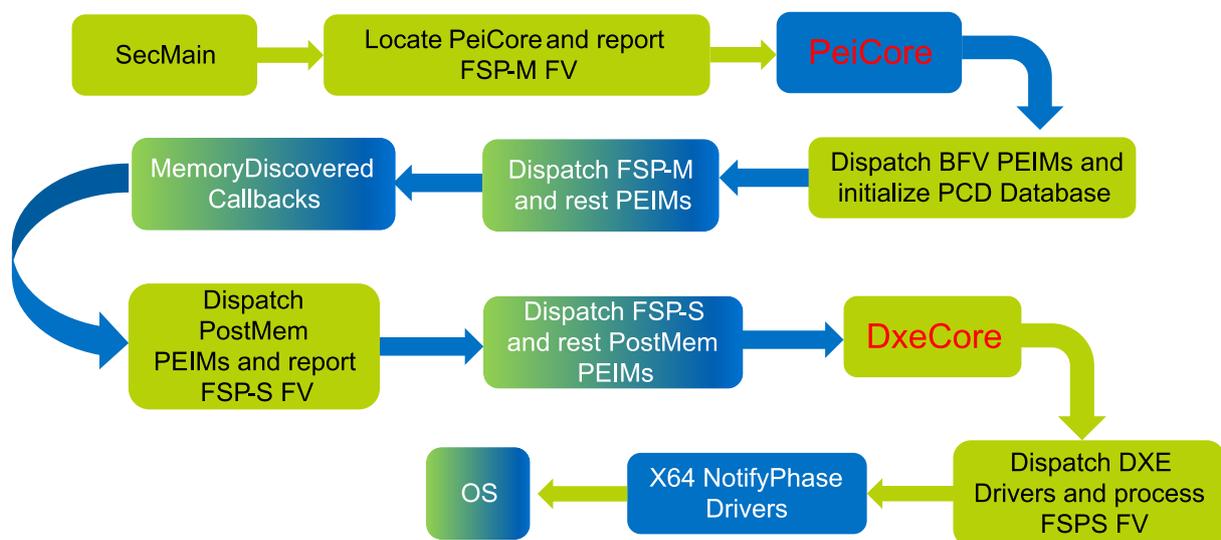
Figure 3.1: System Memory Map

Chapter 4

FSP Dispatch Mode

Overview

The FSP for this platform supports dispatch mode. Support for dispatch mode can be detected by checking if `FSP->_INFO_HEADER->ImageAttribute[BIT1] == 1`. Dispatch mode is intended to enable FSP to integrate well in to UEFI bootloader implementations. Dispatch mode implements a boot flow that is as close to a standard UEFI boot flow as possible. In dispatch mode, the FSP is exposed as standard Firmware Volumes (FVs) directly to the bootloader. The PEIMs in these FVs are executed in the same PEI environment as the boot loader. In dispatch mode, the PPI database, PCD database, and HOB list are shared between the boot loader and the FSP.



Blue blocks are from the FSP binary and green blocks are from the bootloader. Blocks with mixed colors indicate that both bootloader and FSP modules are dispatched during that phase of the boot flow. In dispatch mode, the `NotifyPhase()` API is not used. Instead, FSP-S contains DXE drivers that implement the native callbacks on equivalent events for each of the `NotifyPhase()` invocations.

In dispatch mode, the the PPI database and PCD database are used for providing policy data from the bootloader to FSP. Because these mechanisms provide a great deal of flexibility, dispatch mode does not constrain the method for passing policy data as strongly as API mode. The following sections describe the dispatch mode policy initialization flow used specifically for this platform.

4.1 Dispatch Mode Policy Init

Before FSP-M, the bootloader must ensure that all of the policy data structures described in [section 4.2.1](#) have been created and installed.

All these policy values must be correct. The bootloader must ensure this is done before any PEIMs in FSP-M are dispatched. One method to ensure policy data is initialized at the correct time is to not install the `FV_INFO_PPI` for FSP-M until policy data initialization is complete.

Before FSP-S, the bootloader must ensure that all of the policy data structures described in [section 4.2.3](#) have been created and installed. All these policy values must be correct. The bootloader must ensure this is done before any PEIMs in FSP-S are dispatched. One method to ensure policy data is initialized at the correct time is to not install the `FV_INFO_PPI` for FSP-S until policy data initialization is complete.

4.2 Dispatch Mode Policy Data Structures

Overview

Policy data for this platform is stored using PPIs, HOBs, and DynamicEx PCDs.

See the sections below for details.

4.2.1 Policy Data Structures used by FSP-M

The following policy data structures are consumed by FSP-M in dispatch mode:

Name	Data Structure Type
MEMORY_POLICY_PPI	PPI
UPI_POLICY_PPI	PPI
PCH_POLICY_PPI	PPI
RAS_RC_POLICY_PPI	PPI
RAS_IMC_S3_DATA_PPI	PPI
CPU_POLICY_HOB	HOB
PSMI_POLICY_DATA_HOB	HOB
SECURITY_POLICY	HOB
PcdEvMode	DynamicEx PCD

4.2.2 RAS_IMC_S3_DATA_PPI

An SMM driver in FvLateSilicon can under certain conditions write to a UEFI variable during OS runtime. This data is used by the MRC contained in FSP-M to enable fast MRC and S3 resume. The [RAS_IMC_S3_DATA_PPI](#) provides the mechanism for this data to be read by FSP-M. This PPI contains 1 function, [GetImcS3RasData\(\)](#) which retrieves the data from this UEFI variable and returns it. This PPI must be implemented by the boot loader when using FSP Dispatch mode. The variable that this function must retrieve has the name `L"ImcRasS3SaveData"` and the GUID `{0xe626f9ca, 0xfd71, 0x458c, {0xb9, 0x26, 0xbf, 0x40, 0x80, 0x62, 0x42, 0xa9}}`.

4.2.3 Policy Data Structures used by FSP-S

The following policy data structures are consumed by FSP-S in dispatch mode:

Name	Data Structure Type
PCH_POLICY_PPI	PPI

4.3 FSP Error Information

In the case of a fatal error occurring during the execution of the FSP, it may not be possible for the FSP to continue.

If a fatal error that prevents the successful completion of the FSP occurs, the FSP may use FSP_ERROR_INFO to report this error to the bootloader. During PEI phase, (*PeiServices)->ReportStatusCode() shall be used to transmit this error notification to the bootloader. During DXE phase, EFI_STATUS_CODE_PROTOCOL.ReportStatusCode() shall be used to transmit this error notification to the bootloader. The bootloader must ensure that ReportStatusCode() services are available before FSP-M begins execution.

```
typedef struct {
    EFI_STATUS_CODE_DATA    DataHeader;
    EFI_GUID                ErrorType;
    EFI_STATUS              Status;
} FSP_ERROR_INFO;
```

FSP_ERROR_INFO is provided as the optional EFI_STATUS_CODE_DATA parameter to ReportStatusCode(). EFI_STATUS_CODE_DATA provides a CallerId GUID, this CallerId combined with the ErrorType GUID describes the error to the bootloader.

When the FSP calls ReportStatusCode(), the Type parameter's EFI_STATUS_CODE_TYPE_MASK must be EFI_ERROR_CODE with the EFI_STATUS_CODE_SEVERITY_MASK >= EFI_ERROR_UNRECOVERED. The Value and Instance parameters must be 0.

The bootloader must register a listener for this status code. This listener should check if DataHeader.Type == STATUS_CODE_DATA_TYPE_FSP_ERROR_GUID to detect an FSP_ERROR_INFO notification. If an FSP_ERROR_INFO notification is encountered, the bootloader should assume that normal operation is no longer possible. In debug scenarios, this notification should be considered an ASSERT.

4.4 Dispatch Mode Integration

Dispatch Mode Integration Notes:

1. The FSP for this platform contains PEIMs compiled for the IA32 architecture. The boot loader therefore must utilize a PEI Foundation compiled for the IA32 architecture.
2. Since the FSP binary can be integrated into flash at any address, the boot loader has to report FSP FVs to the PEI and DXE dispatcher using PI specification defined mechanisms so PEIMs and DXE drivers inside the FSP Binary can be dispatched. FspmWrapperPeim and FspWrapperPeim from IntelFsp2WrapperPkg can aid in implementing this.
3. For this platform, FSP-T, FSP-M, and FSP-S contain 1 FV each.
4. The FSP distribution package will include a DSC file which contains all DynamicEx PCDs consumed by the FSP binary. The boot loader should include the DSC during its build process so that any PCDs defined by this DSC file are included in the boot loader's PCD database. This enables the boot loader and FSP to share a single PCD database.
 - A NULL library (FspPcdListLibNull.inf) is included in the FSP distribution package. This library should be included in one of the boot loader's PEIMs. This ensures all DynamicEx PCDs used by the FSP are included in the boot loader's PCD database. One can fulfill this requirement by including the following code snippet in *BoardPkg.dsc:

```
IntelFsp2WrapperPkg/FspmWrapperPeim/FspmWrapperPeim.inf {
<LibraryClasses>
!if gIntelFsp2WrapperTokenSpaceGuid.PcdFspModeSelection == 0
#
# In FSP Dispatch mode below dummy library should be linked to bootloader PEIM
# to build all DynamicEx PCDs that FSP consumes into bootloader PCD database.
```

```
    #  
    NULL|$(PLATFORM_FSP_BIN_PACKAGE)/Library/FspPcdListLib/FspPcdListLibNull.inf  
  !endif  
}
```

5. The boot loader must provide at minimum 256KB of stack and 512KB of HOB heap to execute FSP on this platform.
6. In dispatch mode, the boot loader should not use FSP API calls described in chapter 5 of this document or chapter 8 of the FSP External Architecture Specification version 2.2. The TempRamInit API is the only exception, it is supported in both API mode and dispatch mode. All other APIs (MemoryInit, SiliconInit, etc.) should not be invoked.
7. For dispatch mode, FSP contains x64 DXE drivers to replace the NotifyPhase API. This eliminates thunking from 64bit to 32bit when using FSP dispatch mode. The boot loader should remove S3EndOfPeiNotify and FspWrapperNotifyDxe since they are no longer used in dispatch mode.
8. EFI_PEI_CORE_FV_LOCATION_PPI should be installed by the boot loader's SEC phase. EFI_PEI_CORE_FV_LOCATION_PPI.PeiCoreFvLocation should point to the first Firmware Volume (FV) in FSP-M so the PeiCore inside FSP will be invoked. If EFI_PEI_CORE_FV_LOCATION_PPI is not installed or PeiCore cannot be found at the address specified by EFI_PEI_CORE_FV_LOCATION_PPI.PeiCoreFvLocation, the PeiCore from the Boot Firmware Volume (BFV) will be invoked instead.
9. FSP-S requires multi-threaded code to complete silicon initialization on this platform. The boot loader must include the UefiCpuPkg/CpuMpPei/CpuMpPei.inf PEIM to provide the MP_SERVICES implementation necessary for this. FSP-M can perform an INIT-SIPI-SIPI during the pre-memory phase. The boot loader should anticipate this and expect that an INIT-SIPI-SIPI will be needed after reaching post-memory.
10. FSPM_ARCH_CONFIG_PPI->NvsBufferPtr is required by this FSP implementation. To enable the fast MRC training flow, the boot loader must to install this PPI to restore the previous MRC training data.

Chapter 5

FSP API Mode

Overview

This release of the FSP implements the all APIs required by the FSP External Architecture Specification version 2.2. These APIs are only used when running the FSP in API mode. In Dispatch mode, these APIs are not used (with the exception of TempRamInit.) The FSP information header contains the address offset for these APIs. Register usage is described in the FSP External Architecture Specification version 2.2. Any usage not described by the specification is described in the individual sections below.

FSP API mode for this platform offers a reduced feature set compared to dispatch mode. For example, dispatch mode offers thousands of policy options that are not available in API mode. To access the full spectrum of platform features, use of FSP dispatch mode is recommended.

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

5.1 FSP APIs

5.1.1 TempRamInit API

Please refer Chapter 8.6 in the FSP External Architecture Specification version 2.2 for complete details including the prototype, parameters and return value details for this API.

TempRamInit is mandatory for this platform.

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 total temporary ram currently available is given by [PcdTemporaryRamSize](#) starting from the base address of [PcdTemporaryRamBase](#). Out of the total temporary memory available, the last [PcdFspReservedBufferSize](#) bytes of space are reserved by the FSP for TempRamInit if temporary RAM initialization is done by the FSP. Any remaining space from **TemporaryRamBase**(ECX) to **TemporaryRamBase+TemporaryRamSize-FspReservedBufferSize** (EDX) is available for both bootloader and FSP use.

TempRamInit** also sets up the code caching of the region passed in through CodeCacheBase and CodeCacheLength, which are input parameters to TempRamInitApi. if 0 is passed in for CodeCacheBase, the base used will be (4 GB - 1 - CodeCacheLength).

Note

: When programming MTRRs CodeCacheLength will be reduced, if the LLC size on the current processor is smaller than the requested size.

It is a requirement for Firmware to have a Firmware Interface Table (FIT). The FIT contains pointers to each microcode update. The microcode update is loaded by some of the logical processors before executing the reset

vector. If more than one microcode update for the CPU is present, the microcode update with the latest revision is loaded.

`FSPT_UPD.MicrocodeRegionBase**` and `FSPT_UPD.MicrocodeRegionLength` are input parameters to `TempRamInit` API. These values are required to be set to a valid microcode region to complete the `TempRamInit()` function successfully.

5.1.2 FspMemoryInit API

Please refer to Chapter 8.7 in the FSP External Architecture Specification version 2.2 for the prototype, parameters and return value details for this API.

The variable `FspmUpdPtr` is a pointer to the `FSPM_UPD` structure which is described in the header file `FspmUpd.h`.

The bootloader must pass valid a CAR region for FSP through `FSPM_UPD.FspmArchUpd.StackBase` and `FSPM_UPD.FspmArchUpd.StackSize` UPDs.

The FSP for this platform will run `FspMemoryInit` top of the stack provided by the bootloader instead of establishing a separate stack as described by the FSP External Architecture Specification version 2.1/2.2. The memory region provided by the `FSPM_UPD.FspmArchUpd.StackBase` and `FSPM_UPD.FspmArchUpd.StackSize` UPDs is used to establish a HOB heap. The names `StackBase` and `StackSize` can be confusing since they are **NOT** used for stack. These names were retained for backwards compatibility with FSP v2.0.

Below are the heap and stack requirements for FSP on this platform:

HOB Heap requirement:

HOB Heap	UPD	Setting
Base	<code>FSPM_UPD.FspmArchUpd.StackBase</code>	Any non-conflict CAR region (0xFE17F00 as default)
Size	<code>FSPM_UPD.FspmArchUpd.StackSize</code>	at least 512KB

Stack requirement: FSP's stack usage starts from the current stack pointer. The minimum stack size requirement for FSP-M is 256KB.

The bootloader must ensure that sufficient stack space is available to fulfill the FSP-M minimum stack size requirement at the point in execution where `FspMemoryInit()` is called. The stack allocated by the bootloader must be large enough for both FSP-M as well as any other parent function calls that are still on the stack at the point when `FspMemoryInit()` is called.

After `FspMemoryInit()` is completed, permanent memory is available. After this point, the memory pressure experienced early in boot is eliminated. Accordingly, right before `FspMemoryInit()` exits, any data that needs to be retained for later use by `FspSiliconInit()` will be copied to permanent memory. `FspSiliconInit()` will then execute on a second stack.

The base address of HECI device (Bus 0, Device 22, Function 0) is required to be initialized prior to calling `FspMemoryInit()`. The default address is programmed to 0xFEDB0000.

`FspMemoryInit()` will program the TSEG (SMRAM) memory size to 128MB, which is the maximum allowed by the processor design.

`FspMemoryInit()` will calculate the memory map by taking into account the size of several memory regions: TSEG, IED, ME stolen, Uncore PMRR, IOT, MOT, DPR, REMAP, TOLUD, TOUUD. These memory regions may not be initialized by `FspMemoryInit()`, but space will be reserved for them.

5.1.3 TempRamExit API

Please refer to Chapter 8.8 in the FSP external Architecture Specification version 2.2 for the prototype, parameters and return value details for this API.

The FSP for this platform doesn't have any input parameters for this API. The value of `TempRamExitParamPtr` should be NULL.

At the end of *TempRamExit* the original code and data cache are disabled. FSP will reconfigure all MTRRs as described in the table below. These MTRR values optimize performance in most scenarios. If the boot loader wishes to configure the MTRRs differently, they can be reprogrammed immediately after this API call.

Memory range	Cache Attribute
0xFF000000 - 0xFFFFFFFF (Flash region)	Write protect
0x00000000 - 0x0009FFFF	Write back
0x000C0000 - Top of Low Memory	Write back
xxxx - xxxx	x *Note1
0x100000000 - Top of High Memory	Write back *Note2

Stack requirement: 4KB of free stack space should be provided to execute **TempRamExit**.

Note1: Certain silicon features require specific cache types for specific memory ranges. These ranges will be configured by FSP when such features are enabled.

Note2: In some cases MTRRs might not be enough to cover all desired regions, in this case memory regions need to be adjusted for better alignment (e.g., adjust *MmioSize* or *MmioSizeAdjustment* UPD) Covering flash region and above 4GB memory is another case which may consume more MTRRs, when there is not enough MTRR available FSP will only cover above 4GB memory partially. In this case the boot loader can optimize MTRRs to remove flash from the cached regions after all needed data is loaded from flash and before booting the OS.

5.1.4 FspSiliconInit API

Please refer to Chapter 8.9 in the FSP external Architecture Specification version 2.2 for the prototype, parameters and return value details for this API.

The variable *FspUpdPtr* is a pointer to the **FSPS_UPD** structure which is described in the header file [FspUpd.h](#).

It is expected that the boot loader will adjust MTRRs for SBSP if needed after **TempRamExit** but before entering **FspSiliconInit**. If the MTRRs are not programmed properly, boot performance can be impacted.

The region of 0x5_8000 - 0x5_8FFF is used by *FspSiliconInit* for starting APs. If this data is important to bootloader, then bootloader needs to preserve it before calling *FspSiliconInit*.

It is a requirement for the bootloader to have a Firmware Interface Table (FIT), which contains pointers to each microcode. The microcode is loaded for all cores before reset vector. If more than one microcode update for the CPU is present, the latest revision is loaded.

Stack requirement: 4KB of free stack space should be provided to execute *FspSiliconInit*.

5.1.5 FspMultiPhaseSilnit API

Please refer Chapter 8.10 in the FSP External Architecture Specification version 2.2 for the prototype, parameters and return value details for this API.

This platform does not support the *FspMultiPhaseSilnit* API. The bootloader should not attempt to call *FspMultiPhaseSilnit*.

5.1.6 NotifyPhase API

Please refer Chapter 8.11 in the FSP External Architecture Specification version 2.2 for the prototype, parameters and return value details for this API.

Stack requirement: 4KB of free stack space should be provided to execute *NotifyPhase*.

5.1.6.1 PostPciEnumeration Notification

This phase *EnumInitPhaseAfterPciEnumeration* is to be called after PCI enumeration but before execution of third party code such as option ROMs.

5.1.6.2 ReadyToBoot Notification

This phase *EnumInitPhaseReadyToBoot* is to be called before giving control to the operating system. It includes some final initialization steps recommended by the BWG, including power management settings.

5.1.6.3 EndOfFirmware Notification

This phase *EnumInitEndOfFirmware* is to be called before the firmware/preboot environment transfers management of all system resources to the OS or next level execution environment. It includes final locking of chipset registers.

5.1.7 FSP Events API

Please refer Chapter 8.5 in the FSP External Architecture Specification version 2.2 for the prototype, parameters and return value details for these APIs.

This platform does not support FSP Events.

5.2 Reset Return Codes

As per FSP External Architecture Specification version 2.0/2.1/2.2, any reset required in the FSP flow will be reported by returning one of the `FSP_STATUS_RESET_REQUIRED*` return codes.

It is the boot loader's responsibility to reset the system according to the reset type requested.

Below table specifies the return status returned by FSP API and the requested reset type.

FSP_STATUS_RESET_REQUIRED Code	Reset Type requested
0x40000001	Cold Reset
0x40000002	Warm Reset
0x40000003	Global Reset - Puts the system through a Global Reset through HECI or a Full Reset through PCH
0x40000004	Reserved
0x40000005	Reserved
0x40000006	Reserved
0x40000007	Reserved
0x40000008	Reserved

Chapter 6

Porting Recommendations

Here are some notes and recommendations for adapting an existing boot loader to FSP.

6.1 Locking SMI register

It is recommended that the global SMI bit is locked before any third party code (e.g. OpROM) execution. SMM initialization flows may vary depending on boot loader implementation details. Accordingly, FSP will not lock it by default. The boot loader is responsible for locking the following registers after SMM configuration is complete. Set `AcpiBase + 0x30[0]` to 1b to enable global SMI. Set `PMC PCI offset A0h[4]` = 1b to lock SMI.

Chapter 7

Todo List

Member [PCH_PM_CONFIG::RsvdBits0](#)
ADD DESCRIPTION

Chapter 8

Deprecated List

Member [KTI_HOST_IN::ColdResetRequestEnd](#)

Reserved.

Member [KTI_HOST_IN::ColdResetRequestStart](#)

Reserved.

Member [KTI_HOST_IN::highGap](#)

Reserved.

Member [KTI_HOST_IN::lowGap](#)

Reserved.

Member [KTI_HOST_IN::OemCheckCpuPartsChangeSwap](#)

Reserved, must be set to 0.

Member [KTI_HOST_IN::OemGetAdaptedEqSettings](#)

Reserved, must be set to 0.

Member [KTI_HOST_IN::SplitLock](#)

Reserved, must be set to 0.

Member [PCH_PCIE_ROOT_PORT_CONFIG::HsioRxSetCtle](#)

, please use HsioRxSetCtle from [PCH_HSIO_PCIE_LANE_CONFIG](#)

Member [PCH_PCIE_ROOT_PORT_CONFIG::HsioRxSetCtleEnable](#)

, please use HsioRxSetCtleEnable from [PCH_HSIO_PCIE_LANE_CONFIG](#)

Member [PCH_SATA_PORT_CONFIG::HsioRxEqBoostMagAd](#)

, please use HsioRxGen3EqBoostMag

Member [PCH_SATA_PORT_CONFIG::HsioRxEqBoostMagAdEnable](#)

, please use HsioRxGen3EqBoostMagEnable

Member [PCH_SATA_PORT_CONFIG::HsioTxGen1DownscaleAmp](#)

, please use HsioTxGen1DownscaleAmp in [PCH_HSIO_SATA_PORT_LANE](#)

Member [PCH_SATA_PORT_CONFIG::HsioTxGen1DownscaleAmpEnable](#)

, please use HsioTxGen1DownscaleAmpEnable in [PCH_HSIO_SATA_PORT_LANE](#)

Member [PCH_SATA_PORT_CONFIG::HsioTxGen2DownscaleAmp](#)

, please use HsioTxGen2DownscaleAmp in [PCH_HSIO_SATA_PORT_LANE](#)

Member [PCH_SATA_PORT_CONFIG::HsioTxGen2DownscaleAmpEnable](#)

, please use HsioTxGen2DownscaleAmpEnable in [PCH_HSIO_SATA_PORT_LANE](#)

Member [PCH_WAKE_CONFIG::Gp27WakeFromDeepSx](#)

Member [RAS_RC_POLICY_PPI::CrashLogClear](#)

Member [RAS_RC_POLICY_PPI::CrashLogReArm](#)

Member [SECURITY_POLICY::SgxDebugMode](#)

Member [SECURITY_POLICY::SgxSinitDataFromTpm](#)

SGX SVN data from TPM; 0: when SGX is disabled or TPM is not present or no data is present in TPM.

Member [SECURITY_POLICY::SgxSinitNvsData](#)

SGX NVS data from Flash passed during previous boot using CPU_INFO_PROTOCOL.SGX_INFO; Pass value of zero if there is not data saved or when SGX is disabled.

Chapter 9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_EFI_MM_RESERVED_MMRAM_REGION	Structure describing a MMRAM region which cannot be used for the MMRAM heap	31
_EFI_PEI_MP_SERVICES_PPI	This PPI is installed by some platform or chipset-specific PEIM that abstracts handling multiprocessor support	32
_LIST_ENTRY	_LIST_ENTRY structure definition	32
_MEMORY_POLICY_PPI	Memory Policy PPI Definition	33
_PCH_POLICY	The PCH Policy allows the platform code to publish a set of configuration information that the PCH drivers will use to configure the PCH hardware	33
_RAS_IMC_S3_DATA_PPI	RAS IMC S3 Data PPI	41
_UPI_POLICY_PPI	UPI Policy Structure	42
AdvMemTestRankData	Define AdvMemTest Rank List item The input format is defined as follows: Rank number in bits[3:0] DIMM number in bits[7:4] Channel number in the MC in bits[11:8] MC number in bits[15:12] Socket number in bits [19:16] bits [31:20] are reserved For example: To test M↔C 0, CH 1, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x100 To test MC 1, CH 0, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x1000	43
ALL_LANES_EPARAM_LINK_INFO	All Lanes PHY Configuration	43
commonSetup	Common Platform Settings of MRC	44
CPU_POLICY_HOB	CPU Initialization Policy Options	46
ddrChannelSetup	Channel setup structure declaration	48
ddrDimmSetup	DIMM enable/disable information	49
ddrSocketSetup	Socket setup structure declaration	49
DMI_HW_WIDTH_CONTROL	This structure allows to customize DMI HW Autonomous Width Control for Thermal and Mechanical spec design	50

EFI_HOB_CPU	Describes processor information, such as address space and I/O space capabilities	51
EFI_HOB_FIRMWARE_VOLUME	Details the location of firmware volumes that contain firmware files	52
EFI_HOB_FIRMWARE_VOLUME2	Details the location of a firmware volume that was extracted from a file within another firmware volume	52
EFI_HOB_FIRMWARE_VOLUME3	Details the location of a firmware volume that was extracted from a file within another firmware volume	54
EFI_HOB_GENERIC_HEADER	Describes the format and size of the data inside the HOB	55
EFI_HOB_GUID_TYPE	Allows writers of executable content in the HOB producer phase to maintain and manage HOBs with specific GUID	56
EFI_HOB_HANDOFF_INFO_TABLE	Contains general state information used by the HOB producer phase	56
EFI_HOB_MEMORY_ALLOCATION	Describes all memory ranges used during the HOB producer phase that exist outside the HOB list	58
EFI_HOB_MEMORY_ALLOCATION_BSP_STORE	Defines the location of the boot-strap processor (BSP) BSPStore ("Backing Store Pointer Store")	59
EFI_HOB_MEMORY_ALLOCATION_HEADER	EFI_HOB_MEMORY_ALLOCATION_HEADER describes the various attributes of the logical memory allocation	60
EFI_HOB_MEMORY_ALLOCATION_MODULE	Defines the location and entry point of the HOB consumer phase	61
EFI_HOB_MEMORY_ALLOCATION_STACK	Describes the memory stack that is produced by the HOB producer phase and upon which all post-memory-installed executable content in the HOB producer phase is executing	62
EFI_HOB_MEMORY_POOL	Describes pool memory allocations	63
EFI_HOB_RESOURCE_DESCRIPTOR	Describes the resource properties of all fixed, nonrelocatable resource ranges found on the processor host bus during the HOB producer phase	64
EFI_HOB_UEFI_CAPSULE	Each UEFI capsule HOB details the location of a UEFI capsule	66
EFI_IP_ADDRESS	16-byte buffer aligned on a 4-byte boundary	67
EFI_MAC_ADDRESS	32-byte buffer containing a network Media Access Control address	67
EFI_MMRAM_DESCRIPTOR	Structure describing a MMRAM region and its accessibility attributes	68
EFI_PEI_HOB_POINTERS	Union of all the possible HOB Types	69
EFI_TIME	EFI Time Abstraction: Year: 1900 - 9999 Month: 1 - 12 Day: 1 - 31 Hour: 0 - 23 Minute: 0 - 59 Second: 0 - 59 Nanosecond: 0 - 999,999,999 TimeZone: -1440 to 1440 or 2047	69
FSPM_CONFIG	FSP-M Configuration	70
FSPM_UPD	Fsp M UPD Configuration	82
FSPS_CONFIG	FSP-S Configuration	83
FSPS_UPD	Fsp S UPD Configuration	85
FSPT_CONFIG	FSP-T Configuration	85

FSPT_CORE_UPD	
FSP-T Core UPD	86
FSPT_UPD	
Fsp T UPD Configuration	86
GUID	
128 bit buffer containing a unique identifier value	87
IPv4_ADDRESS	
4-byte buffer	88
IPv6_ADDRESS	
16-byte buffer	88
KTI_HOST_IN	
KTIRC input structure	88
memSetup	
Host memory setup structure declaration	93
memTiming	
Memory Timings Settings	123
PCH_DCI_CONFIG	
This structure contains the policies which are related to Direct Connection Interface (DCI)	126
PCH_DEVICE_INTERRUPT_CONFIG	
The PCH_DEVICE_INTERRUPT_CONFIG block describes interrupt pin, IRQ and interrupt mode for PCH device	127
PCH_DMI_CONFIG	
The PCH_DMI_CONFIG block describes the expected configuration of the PCH for DMI	127
PCH_FLASH_PROTECTION_CONFIG	
PCH Flash Protection Configuration	128
PCH_GBL2HOST_EN	
This PCH_GBL2HOST_EN specifies enable bits related to the "Convert Global Resets to Host Resets" (G2H) feature	129
PCH_GENERAL_CONFIG	
PCH General Configuration	129
PCH_HDAUDIO_CONFIG	
This structure contains the policies which are related to HD Audio device (cAVS)	130
PCH_HPET_CONFIG	
The PCH_HPET_CONFIG block passes the bus/device/function value for HPET	132
PCH_HSIO_PCIE_CONFIG	
The PCH_HSIO_PCIE_CONFIG block describes the configuration of the HSIO for PCIe lanes	132
PCH_HSIO_PCIE_LANE_CONFIG	
The PCH_HSIO_PCIE_LANE_CONFIG describes HSIO settings for PCIe lane	133
PCH_HSIO_PCIE_WM20_CONFIG	
The PCH_HSIO_PCIE_WM20_CONFIG block describes the configuration of the HSIO for PCIe lanes	135
PCH_HSIO_SATA_CONFIG	
The PCH_HSIO_SATA_CONFIG block describes the HSIO configuration of the SATA controller	136
PCH_HSIO_SATA_PORT_LANE	
The PCH_HSIO_SATA_PORT_LANE describes HSIO settings for SATA Port lane	136
PCH_INTERRUPT_CONFIG	
The PCH_INTERRUPT_CONFIG block describes interrupt settings for PCH	138
PCH_IOAPIC_CONFIG	
The PCH_IOAPIC_CONFIG block describes the expected configuration of the PCH IO APIC, it's optional and PCH code would ignore it if the BdfValid bit is not TRUE	139
PCH_LAN_CONFIG	
PCH intergrated LAN controller configuration settings	140
PCH_LOCK_DOWN_CONFIG	
The PCH_LOCK_DOWN_CONFIG block describes the expected configuration of the PCH for security requirement	140
PCH_LPC_CONFIG	
This structure contains the policies which are related to LPC	142

PCH_LPC_SIRQ_CONFIG	
The PCH_LPC_SIRQ_CONFIG block describes the expected configuration of the PCH for Serial IRQ	143
PCH_MEMORY_THROTTLING	
This structure supports an external memory thermal sensor (TS-on-DIMM or TS-on-Board)	144
PCH_P2SB_CONFIG	
This structure contains the policies which are related to P2SB device	145
PCH_PCIE_CONFIG	
The PCH_PCIE_CONFIG block describes the expected configuration of the PCH PCI Express controllers	145
PCH_PCIE_CONFIG2	
The PCH_PCIE_CONFIG2 block describes the additional configuration of the PCH PCI Express controllers	148
PCH_PCIE_EQ_LANE_PARAM	
Represent lane specific PCIe Gen3 equalization parameters	149
PCH_PCIE_ROOT_PORT_CONFIG	
The PCH_PCIE_ROOT_PORT_CONFIG describe the feature and capability of each PCH PCIe root port	150
PCH_PM_CONFIG	
The PCH_PM_CONFIG block describes expected miscellaneous power management settings	153
PCH_PORT61H_SMM_CONFIG	
This structure is used for the emulation feature for Port61h read	157
PCH_POWER_RESET_STATUS	
This PCH_POWER_RESET_STATUS Specifies which Power/Reset bits need to be cleared by the PCH Init Driver	157
PCH_RST_PCIE_STORAGE_CONFIG	
This structure describes the details of Intel RST for PCIe Storage remapping Note: In order to use this feature, Intel RST Driver is required	158
PCH_SATA_CONFIG	
The PCH_SATA_CONFIG block describes the expected configuration of the SATA controllers	159
PCH_SATA_PORT_CONFIG	
This structure configures the features, property, and capability for each SATA port	161
PCH_SATA_RST_CONFIG	
Rapid Storage Technology settings	163
PCH_SKYCAM_CIO2_FLS_CONFIG	
The PCH_SKYCAM_CIO2_FLS_CONFIG block describes SkyCam CIO2 FLS registers configuration	164
PCH_SMBUS_CONFIG	
The SMBUS_CONFIG block lists the reserved addresses for non-ARP capable devices in the platform	165
PCH_SPI_CONFIG	
This structure contains the policies which are related to SPI	166
PCH_SSI_CONFIG	
These members describe some settings which are related to the SSI ports	166
PCH_THERMAL_CONFIG	
The PCH_THERMAL_CONFIG block describes the expected configuration of the PCH for Thermal	167
PCH_THERMAL_THROTTLING	
This structure decides the settings of PCH Thermal throttling	168
PCH_TRACE_HUB_CONFIG	
The PCH_TRACE_HUB_CONFIG block describes TraceHub settings for PCH	169
PCH_USB20_PORT_CONFIG	
This structure configures per USB2 port physical settings	169
PCH_USB30_PORT_CONFIG	
This structure describes whether the USB3 Port N of PCH is enabled by platform modules	170
PCH_USB_CONFIG	
This member describes the expected configuration of the PCH USB controllers, Platform modules may need to refer Setup options, schematic, BIOS specification to update this field	171

PCH_WAKE_CONFIG	This structure allows to customize PCH wake up capability from S5 or DeepSx by WOL, LAN, PCIE wake events	173
PCH_WDT_CONFIG	This policy clears status bits and disable watchdog, then lock the WDT registers	174
PCH_XDCI_CONFIG	The PCH_XDCI_CONFIG block describes the configurations of the xDCI Usb Device controller	174
PCH_XHCI_SSI_PORT	These members describe some settings which are related to the SSIC ports	175
PER_LANE_EPARAM_LINK_INFO	Per Lane PHY Configuration	175
PPR_ADDR	PPR DRAM Address	176
PPR_ADDR_MRC_SETUP	PPR Address, buffer to hold DRAM Address that need to be repaired. 176	
PROTECTED_RANGE	The PCH provides a method for blocking writes and reads to specific ranges in the SPI flash when the Protected Ranges are enabled	177
PSMI_POLICY_DATA_HOB	PSMI policy	177
RAS_RC_POLICY_PPI	RAS policy being requested of RC	178
SATA_THERMAL_THROTTLE	This structure lists PCH supported SATA thermal throttling register setting for customization . . .	179
SECURITY_POLICY	Security Policy	180
sysSetup	Platform Setting for MRC	182
THERMAL_THROTTLE_LEVELS	This structure lists PCH supported throttling register setting for customization	183
TRACE_INFO	Trace Info	184
TS_GPIO_PIN_SETTING	This structure configures PCH memory throttling thermal sensor GPIO PIN settings	185

Chapter 10

File Index

10.1 File List

Here is a list of all documented files with brief descriptions:

Base.h	Root include file for Mde Package Base type modules	187
CpuPolicyHob.h	CPU Policy HOB	201
FspFixedPcds.h	This file lists all FixedAtBuild PCDs referenced in FSP integration guide	202
FspmUpd.h	Copyright (c) 2021, Intel Corporation	202
FspUpd.h	Copyright (c) 2021, Intel Corporation	204
FspUpd.h	Copyright (c) 2021, Intel Corporation	205
FspUpd.h	Copyright (c) 2021, Intel Corporation	206
KtiHost.h	207
MemoryPolicyPpi.h	Header file defining MEMORY_POLICY_PPI, which is for platform code to set platform specific configurations of memory reference code	208
MpServices.h	This file declares UEFI PI Multi-processor PPI	209
PchPolicyCommon.h	PCH configuration based on PCH policy	214
PiHob.h	HOB related definitions in PI	218
PiMultiPhase.h	Include file matches things in PI for multiple module types	220
PsmiPolicyHob.h	PSMI Policy HOB	223
RasImcS3Data.h	RAS IMC S3 Data Load PPI	223
RasRcPolicyPpi.h	RAS Policy PPI header file	226
SecurityPolicy.h	Provides data structure information used by ServerSecurity features in Mtkme etc	226
UefiBaseType.h	Defines data types and constants introduced in UEFI	227
UpiPolicyPpi.h	Silicon Policy PPI is used for specifying platform related Intel silicon information and policy setting	231

Chapter 11

Class Documentation

11.1 `_EFI_MM_RESERVED_MMRAM_REGION` Struct Reference

Structure describing a MMRAM region which cannot be used for the MMRAM heap.

```
#include <PiMultiPhase.h>
```

Public Attributes

- [EFI_PHYSICAL_ADDRESS MmramReservedStart](#)
Starting address of the reserved MMRAM area, as it appears while MMRAM is open.
- [UINT64 MmramReservedSize](#)
Number of bytes occupied by the reserved MMRAM area.

11.1.1 Detailed Description

Structure describing a MMRAM region which cannot be used for the MMRAM heap.

Definition at line 139 of file PiMultiPhase.h.

11.1.2 Member Data Documentation

11.1.2.1 `UINT64 _EFI_MM_RESERVED_MMRAM_REGION::MmramReservedSize`

Number of bytes occupied by the reserved MMRAM area.

A size of zero indicates the last MMRAM area.

Definition at line 149 of file PiMultiPhase.h.

11.1.2.2 `EFI_PHYSICAL_ADDRESS _EFI_MM_RESERVED_MMRAM_REGION::MmramReservedStart`

Starting address of the reserved MMRAM area, as it appears while MMRAM is open.

Ignored if `MmramReservedSize` is 0.

Definition at line 144 of file PiMultiPhase.h.

The documentation for this struct was generated from the following file:

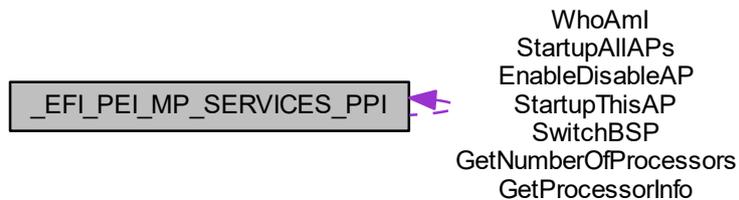
- [PiMultiPhase.h](#)

11.2 `_EFI_PEI_MP_SERVICES_PPI` Struct Reference

This PPI is installed by some platform or chipset-specific PEIM that abstracts handling multiprocessor support.

```
#include <MpServices.h>
```

Collaboration diagram for `_EFI_PEI_MP_SERVICES_PPI`:



11.2.1 Detailed Description

This PPI is installed by some platform or chipset-specific PEIM that abstracts handling multiprocessor support.

Definition at line 265 of file `MpServices.h`.

The documentation for this struct was generated from the following file:

- [MpServices.h](#)

11.3 `_LIST_ENTRY` Struct Reference

[_LIST_ENTRY](#) structure definition.

```
#include <Base.h>
```

Collaboration diagram for `_LIST_ENTRY`:



11.3.1 Detailed Description

[_LIST_ENTRY](#) structure definition.

Definition at line 256 of file `Base.h`.

The documentation for this struct was generated from the following file:

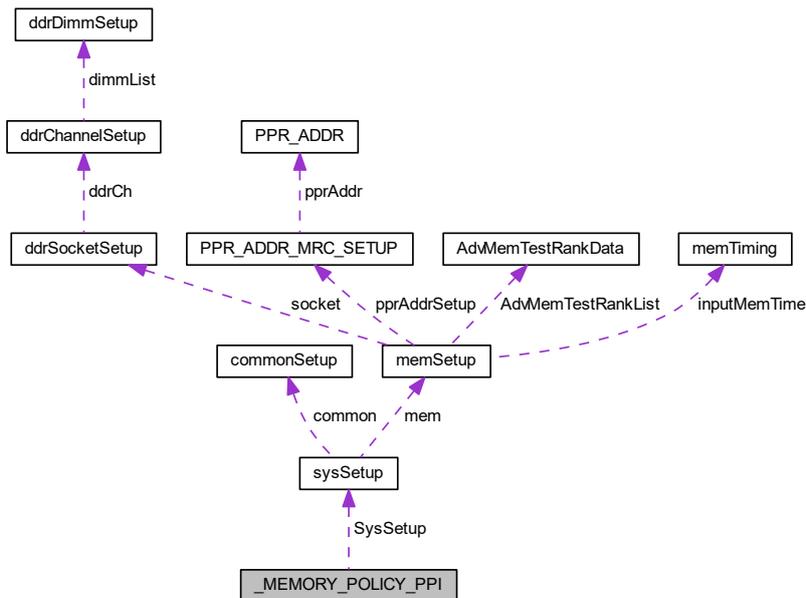
- [Base.h](#)

11.4 _MEMORY_POLICY_PPI Struct Reference

Memory Policy PPI Definition.

```
#include <MemoryPolicyPpi.h>
```

Collaboration diagram for _MEMORY_POLICY_PPI:



Public Attributes

- `UINT32` [Revision](#)
Revision of this PPI.
- `SYS_SETUP` * [SysSetup](#)
This data structure contains all platform level configuration for MRC.

11.4.1 Detailed Description

Memory Policy PPI Definition.

Definition at line 2097 of file `MemoryPolicyPpi.h`.

The documentation for this struct was generated from the following file:

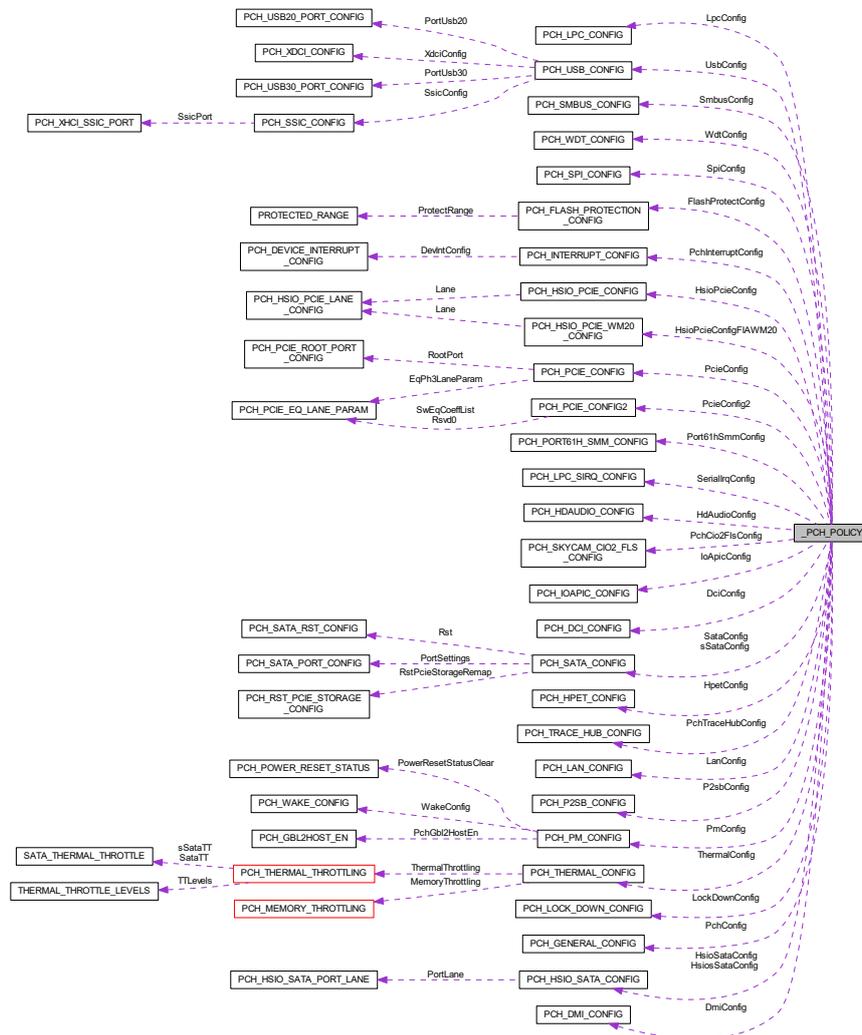
- [MemoryPolicyPpi.h](#)

11.5 _PCH_POLICY Struct Reference

The PCH Policy allows the platform code to publish a set of configuration information that the PCH drivers will use to configure the PCH hardware.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for `_PCH_POLICY`:



Public Attributes

- `UINT8 Revision`
This member specifies the revision of the PCH policy PPI.
- `UINT8 Port80Route`
Control where the Port 80h cycles are sent, **0: LPC**; **1: PCI**.
- `UINT16 AcpiBase`
Power management I/O base address. Default is **0x1800**.
- `PCH_GENERAL_CONFIG PchConfig`
PCH General configuration.
- `PCH_PCIE_CONFIG PcieConfig`
This member describes PCI Express controller's related configuration.
- `PCH_SATA_CONFIG SataConfig`
SATA controller's related configuration.
- `PCH_USB_CONFIG UsbConfig`

- This member describes USB controller's related configuration.*

 - [PCH_IOAPIC_CONFIG IoApicConfig](#)

This member describes IOAPIC related configuration.
 - [PCH_HPET_CONFIG HpetConfig](#)

This member describes HPET related configuration.
 - [PCH_HDAUDIO_CONFIG HdAudioConfig](#)

This member describes the Intel HD Audio (Azalia) related configuration.
 - [PCH_LAN_CONFIG LanConfig](#)

LAN controller settings.
 - [PCH_SMBUS_CONFIG SmbusConfig](#)

This member describes SMBus related configuration.
 - [PCH_LOCK_DOWN_CONFIG LockDownConfig](#)

This member describes LockDown related configuration.
 - [PCH_THERMAL_CONFIG ThermalConfig](#)

This member describes Thermal related configuration.
 - [PCH_PM_CONFIG PmConfig](#)

This member describes miscellaneous platform power management configurations.
 - [PCH_DMI_CONFIG DmiConfig](#)

This member describes DMI related configuration.
 - [PCH_LPC_SIRQ_CONFIG SerialIrqConfig](#)

This member describes the expected configuration of the PCH for Serial IRQ.
 - [PCH_INTERRUPT_CONFIG PchInterruptConfig](#)

This member describes interrupt settings for PCH.
 - [PCH_TRACE_HUB_CONFIG PchTraceHubConfig](#)

This member describes TraceHub settings for PCH.
 - [PCH_PORT61H_SMM_CONFIG Port61hSmmConfig](#)

This member describes the enabling of emulation for port 61h.
 - [PCH_FLASH_PROTECTION_CONFIG FlashProtectConfig](#)

This member describes the Flash Protection related configuration.
 - [PCH_SATA_CONFIG sSataConfig](#)

This member describes the sSata related configuration.
 - [PCH_WDT_CONFIG WdtConfig](#)

This member contains WDT enable configuration.
 - [PCH_P2SB_CONFIG P2sbConfig](#)

This member contains P2SB configuration.
 - [PCH_DCI_CONFIG DciConfig](#)

This member contains DCI configuration.
 - [UINT8 TempPciBusMin](#)

Platform specific common policies that used by several silicon components.
 - [UINT32 TempMemBaseAddr](#)

Temporary Memory Base Address for PCI devices to be used to initialize MMIO registers.
 - [PCH_LPC_CONFIG LpcConfig](#)

This member contains LPC configuration.
 - [PCH_SKYCAM_CIO2_FLS_CONFIG PchCio2FlsConfig](#)

This member describes SkyCam CIO2 FLS registers configuration.
 - [PCH_SPI_CONFIG SpiConfig](#)

This member contains SPI configuration.
 - [PCH_HSIO_SATA_CONFIG HsioSataConfig](#)

This member describes HSIO settings for SATA controller.
 - [PCH_HSIO_SATA_CONFIG HsiosSataConfig](#)

This member describes HSIO settings for second SATA controller.
-

- [PCH_HSIO_PCIE_CONFIG HsioPcieConfig](#)
This member describes HSIO settings for PCIe controller.
- [PCH_HSIO_PCIE_WM20_CONFIG HsioPcieConfigFIAWM20](#)
This member describes HSIO settings for FIA WM20 PCIe.
- [PCH_PCIE_CONFIG2 PcieConfig2](#)
This is the extension of PCIE CONFIG.

11.5.1 Detailed Description

The PCH Policy allows the platform code to publish a set of configuration information that the PCH drivers will use to configure the PCH hardware.

The Revision field is used to accommodate backward compatible changes to the PPI/protocol. The Revision should be initialized to PCH_POLICY_REVISION_X by the PPI producer. The BusNumber field is used for platform to assign Bus number with multiple instances.

All reserved/unused fields must be initialized with zeros.

Definition at line 1814 of file PchPolicyCommon.h.

11.5.2 Member Data Documentation

11.5.2.1 PCH_IOAPIC_CONFIG_PCH_POLICY::IoApicConfig

This member describes IOAPIC related configuration.

Determines IO APIC ID and IO APIC Range.

Definition at line 2046 of file PchPolicyCommon.h.

11.5.2.2 UINT8_PCH_POLICY::Revision

This member specifies the revision of the PCH policy PPI.

This field is used to indicate backwards compatible changes to the protocol. Platform code that produces this PPI must fill with the correct revision value for the PCH reference code to correctly interpret the content of the PPI fields.

Revision 1: Original version

- Add DciAutoDetect policy in [PCH_GENERAL_CONFIG](#).
- Add SbiUnlock policy in [PCH_P2SB_CONFIG](#).
- Add the following policies in PCH_ISH_CONFIG:
 - SpiGpioAssign
 - Uart0GpioAssign
 - Uart1GpioAssign
 - I2c0GpioAssign
 - I2c1GpioAssign
 - I2c2GpioAssign
 - Gp0GpioAssign
 - Gp1GpioAssign
 - Gp2GpioAssign
 - Gp3GpioAssign
 - Gp4GpioAssign

- Gp5GpioAssign
- Gp6GpioAssign
- Gp7GpioAssign
- Add ClkReqSupported and ClkReqDetect in [PCH_PCIE_ROOT_PORT_CONFIG](#).
- Add the following in [PCH_SKYCAM_CIO2_CONFIG](#)
 - SkyCamPortATermOvrEnable
 - SkyCamPortBTermOvrEnable
 - SkyCamPortCTermOvrEnable
 - SkyCamPortDTermOvrEnable
- Add UartHwFlowCtrl in [PCH_SERIAL_IO](#)
- Move DciEn and DciAutoDetect to [PCH_DCI_CONFIG](#)

Revision 2: Updated version

- Add Enable policy in [PCH_SSI_CONFIG](#)
- Deprecated Target Debugger option of EnableMode in [PCH_TRACE_HUB_CONFIG](#)
- Deprecated the following policies in [PCH_TRACE_HUB_CONFIG](#)
 - MemReg0WrapEnable
 - MemReg1WrapEnable
 - TraceDestination
 - PtiMode
 - PtiSpeed
 - PtiTraining
- Deprecated the Usb3PinsTermination and ManualModeUsb30PerPinEnable in [PCH_XHCI_CONFIG](#)
- Redefine the Enable policy in [PCH_HPET_CONFIG](#)
- Add EnhancePort8xhDecoding in [PCH_LPC_CONFIG](#)
- Add PsfUnlock in [PCH_P2SB_CONFIG](#)
- Add AllowNoLtrIccPIIShutdown in [PCH_PCIE_CONFIG](#)
- Add PdtUnlock in [PCH_ISH_CONFIG](#)
- Remove PwrMBase from policy since the base address is predefined.
- Add DspEndpointDmic, DspEndpointBluetooth, DspEndpointI2s in [PCH_HDAUDIO_CONFIG](#)
- Add Gen3EqPh3Method and EqPh3LaneParam in [PCH_PCIE_ROOT_PORT_CONFIG/PCH_PCIE_CONFIG](#)
- Remove SlotImplemented and PmeInterrupt from [PCH_PCIE_ROOT_PORT_CONFIG](#)

Revision 3: Updated version

- Add PwrBtnOverridePeriod policy in [PCH_PM_CONFIG](#)
- Add USB20_AFE in [PCH_USB20_PORT_CONFIG](#)
- Add ClkReqSupported in [PCH_LAN_CONFIG](#)

Revision 4: Updated version

- Add DeviceResetPad and DeviceResetPadActiveHigh in [PCH_PCIE_ROOT_PORT_CONFIG](#)

Revision 5: Updated version

- Deprecated ScsSdioMode in PCH_SCS_CONFIG
- Deprecated PchScsSdioMode (PCH_SCS_DEV_SD_MODE enum) for ScsSdSwitch in PCH_SCS_CONFIG
- Add HSIO RX and TX EQ policy for PCIe and SATA
- Add ComplianceTestMode in [PCH_PCIE_CONFIG](#)

Revision 6: Updated version

- Add DisableEnergyReport in [PCH_PM_CONFIG](#)

Revision 7: Updated version

- Deprecated Enabled as Acpi device option of DeviceEnable in PCH_SKYCAM_CIO2_CONFIG
 - Add [PCH_SKYCAM_CIO2_FLS_CONFIG](#) with the following elements:
 - PortACtleEnable
 - PortBCtleEnable
 - PortCCtleEnable
 - PortDCtleEnable
 - PortACtleCapValue
 - PortBCtleCapValue
 - PortCCtleCapValue
 - PortDCtleCapValue
 - PortACtleResValue
 - PortBCtleResValue
 - PortCCtleResValue
 - PortDCtleResValue
 - PortATrimEnable
 - PortBTrimEnable
 - PortCTrimEnable
 - PortDTrimEnable
 - PortADataTrimValue
 - PortBDataTrimValue
 - PortCDataTrimValue
 - PortDDataTrimValue
 - PortAClkTrimValue
 - PortBClkTrimValue
 - PortCClkTrimValue
 - PortDCIkTrimValue
 - Rename and reorder the policies for better understanding.
 - HsioTxOutDownscaleAmpAd3GbsEnable to HsioTxGen1DownscaleAmpEnable
 - HsioTxOutDownscaleAmpAd6GbsEnable to HsioTxGen2DownscaleAmpEnable
 - HsioTxOutDownscaleAmpAd3Gbs to HsioTxGen2DownscaleAmp
-

- HsioTxOutDownscaleAmpAd6Gbs to HsioTxGen2DownscaleAmp
- Update SerialIo DevMode default to PCI mode.

Revision 8: Updated version

- Deprecate GP27WakeFromDeepSx and add LanWakeFromDeepSx to align EDS naming
- Add ShowSpiController policy and [PCH_SPI_CONFIG](#).
- Add DspUaaCompliance in [PCH_HDAUDIO_CONFIG](#)
- Add PchPcieEqHardware support in [PCH_PCIE_EQ_METHOD](#)

Revision 9: Updated version

- Add DebugUartNumber and EnableDebugUartAfterPost in [PCH_SERIAL_IO_CONFIG](#)
- Add DetectTimeoutMs in [PCH_PCIE_CONFIG](#)
- Add PciePllSsc in [PCH_PM_CONFIG](#)

Revision 10: Updated version

- Add HsioTxDeEmph in [PCH_USB30_PORT_CONFIG](#)
 - Add HsioTxDownscaleAmp in [PCH_USB30_PORT_CONFIG](#)
 - Add HsioTxDeEmphEnable in [PCH_USB30_PORT_CONFIG](#)
 - Add HsioTxDownscaleAmpEnable in [PCH_USB30_PORT_CONFIG](#)
 - Deprecated [PCH_SATA_PORT_CONFIG.HsioRxEqBoostMagAdEnable](#)
 - Deprecated [PCH_SATA_PORT_CONFIG.HsioRxEqBoostMagAd](#)
 - Deprecated [PCH_SATA_PORT_CONFIG.HsioTxGen1DownscaleAmpEnable](#)
 - Deprecated [PCH_SATA_PORT_CONFIG.HsioTxGen1DownscaleAmp](#)
 - Deprecated [PCH_SATA_PORT_CONFIG.HsioTxGen2DownscaleAmpEnable](#)
 - Deprecated [PCH_SATA_PORT_CONFIG.HsioTxGen2DownscaleAmp](#)
 - Add [PCH_HSIO_SATA_CONFIG](#) HsioSataConfig in [PCH_POLICY](#)
 - Add HsioRxGen1EqBoostMagEnable in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioRxGen1EqBoostMag in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioRxGen2EqBoostMagEnable in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioRxGen2EqBoostMag in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen1DeEmphEnable in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen1DeEmph in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen2DeEmphEnable in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen2DeEmph in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen3DeEmphEnable in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen3DeEmph in [PCH_HSIO_SATA_PORT_LANE](#)
 - Add HsioTxGen3DownscaleAmpEnable in [PCH_HSIO_SATA_PORT_LANE](#)
-

- Add HsioTxGen3DownscaleAmp in [PCH_HSIO_SATA_PORT_LANE](#)
- Add [PCH_HSIO_PCIE_CONFIG](#) HsioPcieConfig in PCH_POLICY
- Deprecated [PCH_PCIE_ROOT_PORT_CONFIG.HsioRxSetCtleEnable](#)
- Deprecated [PCH_PCIE_ROOT_PORT_CONFIG.HsioRxSetCtle](#)
- Add HsioRxSetCtleEnable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioRxSetCtle in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen1DownscaleAmpEnable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen1DownscaleAmp in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen2DownscaleAmpEnable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen2DownscaleAmp in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen3DownscaleAmpEnable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen3DownscaleAmp in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen1DeEmphEnable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen1DeEmph in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen2DeEmph3p5Enable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen2DeEmph3p5 in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen2DeEmph6p0Enable in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add HsioTxGen2DeEmph6p0 in [PCH_HSIO_PCIE_LANE_CONFIG](#)
- Add DisableDsxAcpresentPulldown in [PCH_PM_CONFIG](#)
- Add DynamicPowerGating in [PCH_SMBUS_CONFIG](#)
- Add ZpOdd in [PCH_SATA_PORT_CONFIG](#)
- Add Uptp and Dptp in [PCH_PCIE_ROOT_PORT_CONFIG](#)
- Add [PCH_PCIE_CONFIG2](#) PcieConfig2 in PCH_POLICY

Revision 11: Updated version

- Add DisableComplianceMode in [PCH_USB_CONFIG](#)

Revision 12: Updated version

- Add PmcReadDisable in [PCH_PM_CONFIG](#)
- Add CapsuleResetType in [PCH_PM_CONFIG](#)
- Add RpFunctionSwap in [PCH_PCIE_CONFIG](#)

Revision 13: Update version

- Add DisableNativePowerButton in [PCH_PM_CONFIG](#)
 - Add MaxPayload in [PCH_PCIE_ROOT_PORT_CONFIG](#)
 - Add IDispCodecDisconnect in [PCH_HDAUDIO_CONFIG](#) Revision 13a: Server updates
 - Add HsioIcfgAdjLimitLoEnable
-

- Add HsioIcfgAdjLimitLo
- Add HsioSampOffstEvenErrSpEnable
- Add HsioSampOffstEvenErrSp
- Add HsioRemainingSamplerOffEnable
- Add HsioRemainingSamplerOff
- Add HsioVgaGainCal in [PCH_HSIO_PCIE_LANE_CONFIG](#)

Definition at line 2019 of file PchPolicyCommon.h.

11.5.2.3 PCH_SATA_CONFIG_PCH_POLICY::SataConfig

SATA controller's related configuration.

SATA configuration that decides which Mode the SATA controller should operate in and whether PCH SATA TEST mode is enabled.

Definition at line 2037 of file PchPolicyCommon.h.

11.5.2.4 UINT32_PCH_POLICY::TempMemBaseAddr

Temporary Memory Base Address for PCI devices to be used to initialize MMIO registers.

Minimum size is 2MB bytes

Definition at line 2129 of file PchPolicyCommon.h.

11.5.2.5 UINT8_PCH_POLICY::TempPciBusMin

Platform specific common policies that used by several silicon components.

Temp Bus Number range available to be assigned to each root port and its downstream devices for initialization of these devices before PCI Bus enumeration.

Definition at line 2123 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.6 _RAS_IMC_S3_DATA_PPI Struct Reference

RAS IMC S3 Data PPI.

```
#include <RasImcS3Data.h>
```

Collaboration diagram for _RAS_IMC_S3_DATA_PPI:



Public Attributes

- [RAS_IMC_S3_DATA_PPI_GET_IMC_S3_RAS_DATA](#) `GetImcS3RasData`
Retrieves data for S3 saved memory RAS features from non-volatile storage.

11.6.1 Detailed Description

RAS IMC S3 Data PPI.

Definition at line 50 of file `RasImcS3Data.h`.

The documentation for this struct was generated from the following file:

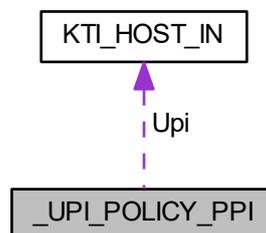
- [RasImcS3Data.h](#)

11.7 `_UPI_POLICY_PPI` Struct Reference

UPI Policy Structure.

```
#include <UpiPolicyPpi.h>
```

Collaboration diagram for `_UPI_POLICY_PPI`:



Public Attributes

- `UINT32` [Revision](#)
This member specifies the revision of the `_UPI_POLICY_PPI`.
- [KTI_HOST_IN](#) `Upi`
KTIRC input structure.

11.7.1 Detailed Description

UPI Policy Structure.

Definition at line 30 of file `UpiPolicyPpi.h`.

11.7.2 Member Data Documentation

11.7.2.1 UINT32_UPI_POLICY_PPI::Revision

This member specifies the revision of the UPI_POLICY_PPI.

This field is used to indicate backwards compatible changes to the INTERFACE. Platform code that produces this INTERFACE must fill with the correct revision value for UPI code to correctly interpret the content of the INTERF↵ACE fields.

Definition at line 37 of file UpiPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [UpiPolicyPpi.h](#)

11.8 AdvMemTestRankData Union Reference

Define AdvMemTest Rank List item The input format is defined as follows: Rank number in bits[3:0] DIMM number in bits[7:4] Channel number in the MC in bits[11:8] MC number in bits[15:12] Socket number in bits [19:16] bits [31:20] are reserved For example: To test MC 0, CH 1, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x100 To test MC 1, CH 0, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x1000.

```
#include <MemoryPolicyPpi.h>
```

11.8.1 Detailed Description

Define AdvMemTest Rank List item The input format is defined as follows: Rank number in bits[3:0] DIMM number in bits[7:4] Channel number in the MC in bits[11:8] MC number in bits[15:12] Socket number in bits [19:16] bits [31:20] are reserved For example: To test MC 0, CH 1, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x100 To test MC 1, CH 0, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x1000.

Definition at line 407 of file MemoryPolicyPpi.h.

The documentation for this union was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.9 ALL_LANES_EPARAM_LINK_INFO Struct Reference

All Lanes PHY Configuration.

```
#include <KtiHost.h>
```

Public Attributes

- [UINT8 SocketID](#)
Socket ID.
- [UINT8 Freq](#)
The Link Speed these TXEQ settings should be used for.
- [UINT32 Link](#)
Port Number.
- [UINT32 AllLanesTXEQ](#)
TXEQ Setting.
- [UINT8 CTLEPEAK](#)
CTLE Peaking Setting.

11.9.1 Detailed Description

All Lanes PHY Configuration.

This is for full speed mode, all lanes have the same TXEQ setting

Definition at line 121 of file KtiHost.h.

The documentation for this struct was generated from the following file:

- [KtiHost.h](#)

11.10 commonSetup Struct Reference

Common Platform Settings of MRC.

```
#include <MemoryPolicyPpi.h>
```

Public Attributes

- UUINT32 [options](#)
Flags for common platform settings.
- UUINT8 [debugJumper](#)
MRC debug feature.
- UUINT32 [serialDebugMsgLvl](#)
Specifies what level of debug messages will be sent to serial port.
- UUINT8 [serialBufEnable](#)
MRC debug feature: Enable/Disable serial port buffer.
- UUINT8 [serialPipeEnable](#)
MRC debug feature: Enable/Disable serial port pipe.
- UUINT8 [serialPipeCompress](#)
MRC debug feature: Enable/Disable serial pipe compress.
- UUINT32 [maxAddrMem](#)
Maximum addressable memory supported by the platform.
- UUINT16 [debugPort](#)
User configurable IO port for post code which is traditionally located at 0x80.
- UUINT32 [nvramptr](#)
32-bit pointer to an optional OEM NVRAM image to be copied into the host NVRAM structure.
- UUINT32 [sysHostBufferPtr](#)
32-bit pointer to an optional OEM provided Host structure.
- UUINT8 [ddrtXactor](#)
Disable/Enable DDRT Transcator.
- UUINT8 [SocketConfig](#)
Socket configuration supported by platform.

11.10.1 Detailed Description

Common Platform Settings of MRC.

Definition at line 1943 of file MemoryPolicyPpi.h.

11.10.2 Member Data Documentation

11.10.2.1 UINT8 commonSetup::ddrtXactor

Disable/Enable DDRT Transcator.

0 - Disable;
1 - Enable;

Definition at line 2026 of file MemoryPolicyPpi.h.

11.10.2.2 UINT8 commonSetup::debugJumper

MRC debug feature.

It indicates if debug jumper is set.

0 - Debug jumper is not set.
1 - Debug jumper is set.

Definition at line 1963 of file MemoryPolicyPpi.h.

11.10.2.3 UINT32 commonSetup::maxAddrMem

Maximum addressable memory supported by the platform.

Skylake Processor supports up to 46-bit addressing. This input should be the total number of addressable bytes in 256MB units. (0x40000 for 46-bit and 0x1000 for 40-bit).

Definition at line 2002 of file MemoryPolicyPpi.h.

11.10.2.4 UINT32 commonSetup::options

Flags for common platform settings.

PROMOTE_WARN_EN BIT0 Enables warnings to be treated as fatal error.

PROMOTE_MRC_WARN_EN BIT1 Enables MRC warnings to be treated as fatal error.

HALT_ON_ERROR_EN BIT2 Enables errors to loop forever.

HALT_ON_ERROR_AUTO BIT3 Auto reset with Maximum Serial port debug message level when fatal error is encountered.

Definition at line 1954 of file MemoryPolicyPpi.h.

11.10.2.5 UINT32 commonSetup::serialDebugMsgLvl

Specifies what level of debug messages will be sent to serial port.

Available options are a bitfield where:

SDBG_MIN BIT0;

SDBG_MAX BIT1;
 SDBG_TRACE BIT2;
 SDBG_MEM_TRAIN BIT3 + SDBG_MAX;
 SDBG_CPGC BIT5;
 SDBG_MINMAX SDBG_MIN + SDBG_MAX.

Definition at line 1977 of file MemoryPolicyPpi.h.

11.10.2.6 UINT8 commonSetup::SocketConfig

Socket configuration supported by platform.

0 - SOCKET_UNDEFINED 1 - SOCKET_4S 2 - SOCKET_HEDT High End Desktop 3 - SOCKET_1S 4 - SOCKET_1SWS 1 Socket Work Station 5 - SOCKET_8S 6 - SOCKET_2S

Definition at line 2044 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.11 CPU_POLICY_HOB Struct Reference

CPU Initialization Policy Options.

```
#include <CpuPolicyHob.h>
```

Public Attributes

- UINT8 [dcuModeSelect](#)
0: 32KB 8-way (hardware default). Non-zero: 16KB 4-way with ECC (CPU MSR 031h)
- UINT8 [EnableGv](#)
GV3 Enable.
- UINT8 [flexRatioEn](#)
FLEX_RATIO Override Enable.
- UINT8 [flexRatioNext](#)
FLEX_RATIO, common for all CPU sockets 0=Don't change flex ratio (default) 0xff = Max Non-turbo ratio.
- UINT8 [IssTdpLevel](#)
0 - 2: 0 = Normal; 1 = Level 1; 2 = Level 2
- UINT8 [DynamicIss](#)
0/1 Disable/Enable Dynamic ISS
- UINT8 [ActivePbf](#)
1: Active PBF if capable
- UINT8 [ConfigTdpLevel](#)
0, 3 - 4: 0 = Base; 3 = Level 3; 4 = Level 4
- UINT16 [NumberOfCores2Disable](#) [MAX_SOCKET]
Number of processor cores to disable for each CPU socket.
- UINT64 [CoreDisableMask](#) [MAX_SOCKET]
CoreOffMask value for each CPU socket (64bits)
- UINT8 [smtEnable](#)
0/1 Disable/Enable SMT(HT). common for all CPU sockets
- UINT8 [vtEnable](#)
0/1 Disable/Enable VMX. Common for all CPU sockets

- UINT8 [IotEn](#) [MAX_SOCKET]
IOT/OCLA Config Disable/Enable,.
- UINT8 [OclaTorEntry](#) [MAX_SOCKET]
IOT/OCLA MaxTorEntry.
- UINT8 [OclaWay](#) [MAX_SOCKET]
IOT/OCLA LLC Ways.
- UINT8 [AllowMixedPowerOnCpuRatio](#)
Keep CPU ratios at power-on default without forcing common ratio among CPU sockets.
- UINT8 [CheckCpuBist](#)
Check BIST result and disable failed cores when enabled. Otherwise, ignore BIST result.
- UINT8 [CoreFailover](#)
Enable spare core(s) in place of core(s) that fail BIST.
- UINT64 [DfxBistFailureEmulation](#)
Emulate core BIST failure to test core sparing.
- UINT8 [debugInterfaceEn](#)
1: Enable Debug Interface for DFX
- UINT8 [WFRWAAEnable](#)
WFRWAAEnable.
- UINT8 [UncoreFreqRapLimit](#)
UncoreFreqRapLimit.
- UINT8 [UncoreFreqScaling](#)
UncoreFreqScaling.
- UINT8 [InputUncoreFreq](#)
InputUncoreFreq.
- UINT8 [PmaxDisable](#)
PmaxDisable.
- UINT8 [RdtCatOpportunisticTuning](#)
RdtCatOpportunisticTuning.
- UINT8 [EarlyC1eEnable](#)
EarlyC1eEnable.
- UINT8 [LlcPrefetchEnable](#)
LlcPrefetchEnable.
- UINT8 [ProcessorMsrLockControl](#)
ProcessorMsrLockControl.
- UINT8 [ProcessorMsrPkgCstConfigControlLock](#)
ProcessorMsrPkgCstConfigControlLock.
- UINT8 [FadrSupport](#)
FadrSupport.
- UINT8 [TscResetEnable](#)
TscResetEnable.

11.11.1 Detailed Description

CPU Initialization Policy Options.

Definition at line 18 of file CpuPolicyHob.h.

11.11.2 Member Data Documentation

11.11.2.1 UINT8 CPU_POLICY_HOB::flexRatioNext

FLEX_RATIO, common for all CPU sockets 0=Don't change flex ratio (default) 0xff = Max Non-turbo ratio.

Other values defines target flex ratio

Definition at line 22 of file CpuPolicyHob.h.

The documentation for this struct was generated from the following file:

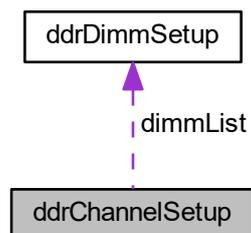
- [CpuPolicyHob.h](#)

11.12 ddrChannelSetup Struct Reference

Channel setup structure declaration.

```
#include <MemoryPolicyPpi.h>
```

Collaboration diagram for ddrChannelSetup:



Public Attributes

- UINT8 [numDimmSlots](#)
Channel enable switch.
- UINT8 [batterybacked](#)
Number of DIMM slots per channel.
- UINT8 [rankmask](#)
ADR Battery backed or not.
- struct [ddrDimmSetup dimmlist](#) [MAX_DIMM]
Rank mask. 0 = disable; 1 = enable.

11.12.1 Detailed Description

Channel setup structure declaration.

Definition at line 336 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.13 ddrDimmSetup Struct Reference

DIMM enable/disable information.

```
#include <MemoryPolicyPpi.h>
```

Public Attributes

- UINT8 [mapOut](#) [MAX_RANK_DIMM]
Setting for each DIMM to be mapped out.

11.13.1 Detailed Description

DIMM enable/disable information.

Definition at line 325 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

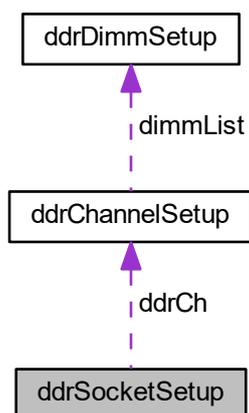
- [MemoryPolicyPpi.h](#)

11.14 ddrSocketSetup Struct Reference

Socket setup structure declaration.

```
#include <MemoryPolicyPpi.h>
```

Collaboration diagram for ddrSocketSetup:



Public Attributes

- UINT8 [enabled](#)
iMC enable/disable switch.
- UINT8 [options](#)

- *Bit-mapped options per socket.*
- struct [ddrChannelSetup ddrCh](#) [MAX_CH]
 - *Platform configuration for each channel.*
- UINT8 [imcEnabled](#) [MAX_IMC]
 - *Enable/Disable memory controller.*

11.14.1 Detailed Description

Socket setup structure declaration.

Definition at line 371 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.15 DMI_HW_WIDTH_CONTROL Struct Reference

This structure allows to customize DMI HW Autonomous Width Control for Thermal and Mechanical spec design.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [DmiTsawEn](#): 1
 - *DMI Thermal Sensor Autonomous Width Enable.*
- UINT32 [SuggestedSetting](#): 1
 - *0: Disable; 1: **Enable** suggested representative values*
- UINT32 [RsvdBits0](#): 6
 - *Reserved bits.*
- UINT32 [TS0TW](#): 2
 - *Thermal Sensor 0 Target Width.*
- UINT32 [TS1TW](#): 2
 - *Thermal Sensor 1 Target Width.*
- UINT32 [TS2TW](#): 2
 - *Thermal Sensor 2 Target Width.*
- UINT32 [TS3TW](#): 2
 - *Thermal Sensor 3 Target Width.*
- UINT32 [RsvdBits1](#): 16
 - *Reserved bits.*

11.15.1 Detailed Description

This structure allows to customize DMI HW Autonomous Width Control for Thermal and Mechanical spec design.

When the SuggestedSetting is enabled, the customized values are ignored.

Definition at line 984 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

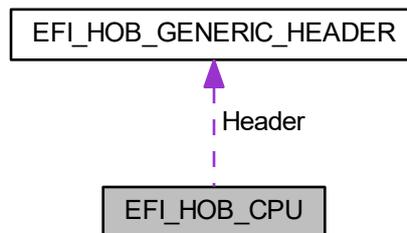
- [PchPolicyCommon.h](#)

11.16 EFI_HOB_CPU Struct Reference

Describes processor information, such as address space and I/O space capabilities.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_CPU:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)
The HOB generic header.
- [UINT8 SizeOfMemorySpace](#)
Identifies the maximum physical memory addressability of the processor.
- [UINT8 SizeOfIoSpace](#)
Identifies the maximum physical I/O addressability of the processor.
- [UINT8 Reserved \[6\]](#)
This field will always be set to zero.

11.16.1 Detailed Description

Describes processor information, such as address space and I/O space capabilities.

Definition at line 438 of file PiHob.h.

11.16.2 Member Data Documentation

11.16.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_CPU::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_CPU.

Definition at line 442 of file PiHob.h.

The documentation for this struct was generated from the following file:

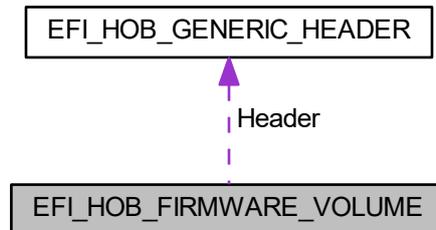
- [PiHob.h](#)

11.17 EFI_HOB_FIRMWARE_VOLUME Struct Reference

Details the location of firmware volumes that contain firmware files.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_FIRMWARE_VOLUME:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)
The HOB generic header.
- [EFI_PHYSICAL_ADDRESS BaseAddress](#)
The physical memory-mapped base address of the firmware volume.
- [UINT64 Length](#)
The length in bytes of the firmware volume.

11.17.1 Detailed Description

Details the location of firmware volumes that contain firmware files.

Definition at line 355 of file PiHob.h.

11.17.2 Member Data Documentation

11.17.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_FIRMWARE_VOLUME::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_FV.

Definition at line 359 of file PiHob.h.

The documentation for this struct was generated from the following file:

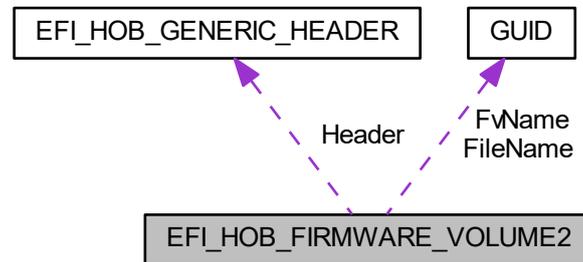
- [PiHob.h](#)

11.18 EFI_HOB_FIRMWARE_VOLUME2 Struct Reference

Details the location of a firmware volume that was extracted from a file within another firmware volume.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_FIRMWARE_VOLUME2:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)
The HOB generic header.
- [EFI_PHYSICAL_ADDRESS BaseAddress](#)
The physical memory-mapped base address of the firmware volume.
- [UINT64 Length](#)
The length in bytes of the firmware volume.
- [EFI_GUID FvName](#)
The name of the firmware volume.
- [EFI_GUID FileName](#)
The name of the firmware file that contained this firmware volume.

11.18.1 Detailed Description

Details the location of a firmware volume that was extracted from a file within another firmware volume.

Definition at line 374 of file PiHob.h.

11.18.2 Member Data Documentation

11.18.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_FIRMWARE_VOLUME2::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_FV2.

Definition at line 378 of file PiHob.h.

The documentation for this struct was generated from the following file:

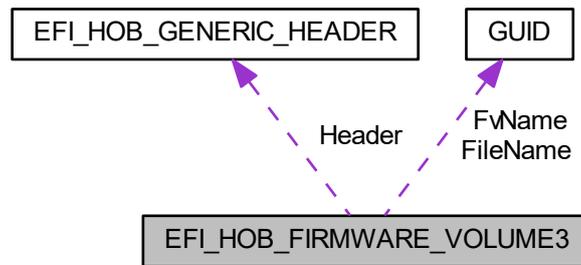
- [PiHob.h](#)

11.19 EFI_HOB_FIRMWARE_VOLUME3 Struct Reference

Details the location of a firmware volume that was extracted from a file within another firmware volume.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_FIRMWARE_VOLUME3:



Public Attributes

- [EFI_HOB_GENERIC_HEADER](#) **Header**
The HOB generic header.
- [EFI_PHYSICAL_ADDRESS](#) **BaseAddress**
The physical memory-mapped base address of the firmware volume.
- [UINT64](#) **Length**
The length in bytes of the firmware volume.
- [UINT32](#) **AuthenticationStatus**
The authentication status.
- [BOOLEAN](#) **ExtractedFv**
TRUE if the FV was extracted as a file within another firmware volume.
- [EFI_GUID](#) **FvName**
The name of the firmware volume.
- [EFI_GUID](#) **FileName**
The name of the firmware file that contained this firmware volume.

11.19.1 Detailed Description

Details the location of a firmware volume that was extracted from a file within another firmware volume.

Definition at line 401 of file PiHob.h.

11.19.2 Member Data Documentation

11.19.2.1 BOOLEAN EFI_HOB_FIRMWARE_VOLUME3::ExtractedFv

TRUE if the FV was extracted as a file within another firmware volume.

FALSE otherwise.

Definition at line 422 of file PiHob.h.

11.19.2.2 EFI_GUID EFI_HOB_FIRMWARE_VOLUME3::FileName

The name of the firmware file that contained this firmware volume.

Valid only if IsExtractedFv is TRUE.

Definition at line 432 of file PiHob.h.

11.19.2.3 EFI_GUID EFI_HOB_FIRMWARE_VOLUME3::FvName

The name of the firmware volume.

Valid only if IsExtractedFv is TRUE.

Definition at line 427 of file PiHob.h.

11.19.2.4 EFI_HOB_GENERIC_HEADER EFI_HOB_FIRMWARE_VOLUME3::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_FV3.

Definition at line 405 of file PiHob.h.

The documentation for this struct was generated from the following file:

- [PiHob.h](#)

11.20 EFI_HOB_GENERIC_HEADER Struct Reference

Describes the format and size of the data inside the HOB.

```
#include <PiHob.h>
```

Public Attributes

- UUINT16 [HobType](#)
Identifies the HOB data structure type.
- UUINT16 [HobLength](#)
The length in bytes of the HOB.
- UUINT32 [Reserved](#)
This field must always be set to zero.

11.20.1 Detailed Description

Describes the format and size of the data inside the HOB.

All HOBs must contain this generic HOB header.

Definition at line 36 of file PiHob.h.

The documentation for this struct was generated from the following file:

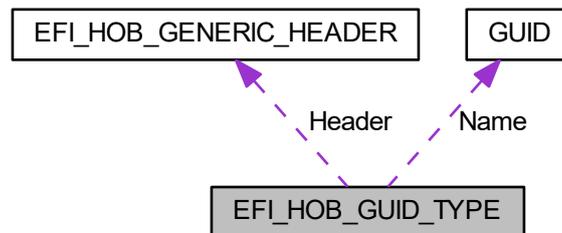
- [PiHob.h](#)
-

11.21 EFI_HOB_GUID_TYPE Struct Reference

Allows writers of executable content in the HOB producer phase to maintain and manage HOBs with specific [GUID](#).

```
#include <PiHob.h>
```

Collaboration diagram for `EFI_HOB_GUID_TYPE`:



Public Attributes

- [EFI_HOB_GENERIC_HEADER](#) Header
The HOB generic header.
- [EFI_GUID](#) Name
A [GUID](#) that defines the contents of this HOB.

11.21.1 Detailed Description

Allows writers of executable content in the HOB producer phase to maintain and manage HOBs with specific [GUID](#).

Definition at line 338 of file `PiHob.h`.

11.21.2 Member Data Documentation

11.21.2.1 `EFI_HOB_GENERIC_HEADER` `EFI_HOB_GUID_TYPE::Header`

The HOB generic header.

`Header.HobType = EFI_HOB_TYPE_GUID_EXTENSION.`

Definition at line 342 of file `PiHob.h`.

The documentation for this struct was generated from the following file:

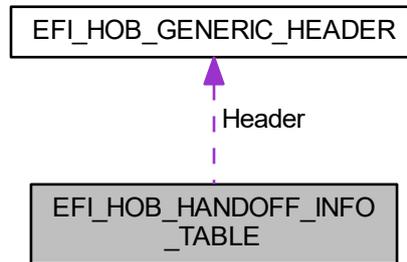
- [PiHob.h](#)

11.22 EFI_HOB_HANDOFF_INFO_TABLE Struct Reference

Contains general state information used by the HOB producer phase.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_HANDOFF_INFO_TABLE:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)
The HOB generic header.
- [UINT32 Version](#)
The version number pertaining to the PHIT HOB definition.
- [EFI_BOOT_MODE BootMode](#)
The system boot mode as determined during the HOB producer phase.
- [EFI_PHYSICAL_ADDRESS EfiMemoryTop](#)
The highest address location of memory that is allocated for use by the HOB producer phase.
- [EFI_PHYSICAL_ADDRESS EfiMemoryBottom](#)
The lowest address location of memory that is allocated for use by the HOB producer phase.
- [EFI_PHYSICAL_ADDRESS EfiFreeMemoryTop](#)
The highest address location of free memory that is currently available for use by the HOB producer phase.
- [EFI_PHYSICAL_ADDRESS EfiFreeMemoryBottom](#)
The lowest address location of free memory that is available for use by the HOB producer phase.
- [EFI_PHYSICAL_ADDRESS EfiEndOfHobList](#)
The end of the HOB list.

11.22.1 Detailed Description

Contains general state information used by the HOB producer phase.

This HOB must be the first one in the HOB list.

Definition at line 61 of file PiHob.h.

11.22.2 Member Data Documentation

11.22.2.1 [EFI_PHYSICAL_ADDRESS EFI_HOB_HANDOFF_INFO_TABLE::EfiMemoryTop](#)

The highest address location of memory that is allocated for use by the HOB producer phase.

This address must be 4-KB aligned to meet page restrictions of UEFI.

Definition at line 80 of file PiHob.h.

11.22.2.2 EFI_HOB_GENERIC_HEADER EFI_HOB_HANDOFF_INFO_TABLE::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_HANDOFF.

Definition at line 65 of file PiHob.h.

11.22.2.3 UINT32 EFI_HOB_HANDOFF_INFO_TABLE::Version

The version number pertaining to the PHIT HOB definition.

This value is four bytes in length to provide an 8-byte aligned entry when it is combined with the 4-byte BootMode.

Definition at line 71 of file PiHob.h.

The documentation for this struct was generated from the following file:

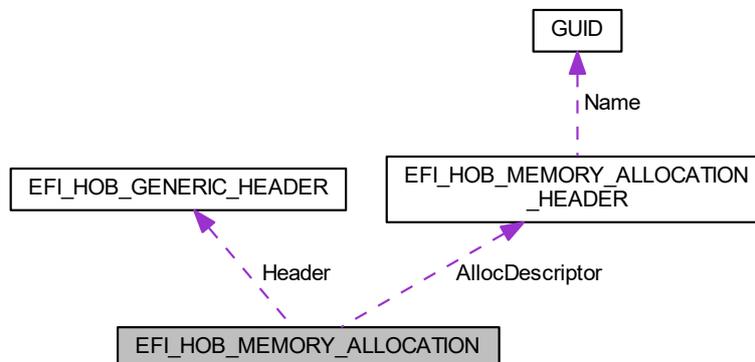
- [PiHob.h](#)

11.23 EFI_HOB_MEMORY_ALLOCATION Struct Reference

Describes all memory ranges used during the HOB producer phase that exist outside the HOB list.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_MEMORY_ALLOCATION:



Public Attributes

- [EFI_HOB_GENERIC_HEADER](#) Header
The HOB generic header.
- [EFI_HOB_MEMORY_ALLOCATION_HEADER](#) AllocDescriptor
An instance of the [EFI_HOB_MEMORY_ALLOCATION_HEADER](#) that describes the various attributes of the logical memory allocation.

11.23.1 Detailed Description

Describes all memory ranges used during the HOB producer phase that exist outside the HOB list.

This HOB type describes how memory is used, not the physical attributes of memory.

Definition at line 145 of file PiHob.h.

11.23.2 Member Data Documentation

11.23.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_MEMORY_ALLOCATION::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_MEMORY_ALLOCATION.

Definition at line 149 of file PiHob.h.

The documentation for this struct was generated from the following file:

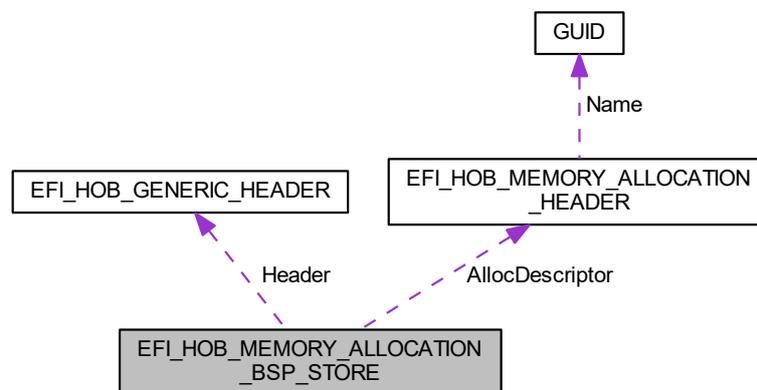
- [PiHob.h](#)

11.24 EFI_HOB_MEMORY_ALLOCATION_BSP_STORE Struct Reference

Defines the location of the boot-strap processor (BSP) BSPStore ("Backing Store Pointer Store").

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_MEMORY_ALLOCATION_BSP_STORE:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)

The HOB generic header.

- [EFI_HOB_MEMORY_ALLOCATION_HEADER AllocDescriptor](#)

An instance of the [EFI_HOB_MEMORY_ALLOCATION_HEADER](#) that describes the various attributes of the logical memory allocation.

11.24.1 Detailed Description

Defines the location of the boot-strap processor (BSP) BSPStore ("Backing Store Pointer Store").

This HOB is valid for the Itanium processor family only register overflow store.

Definition at line 185 of file PiHob.h.

11.24.2 Member Data Documentation

11.24.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_MEMORY_ALLOCATION_BSP_STORE::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_MEMORY_ALLOCATION.

Definition at line 189 of file PiHob.h.

The documentation for this struct was generated from the following file:

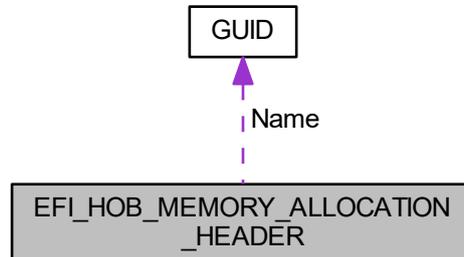
- [PiHob.h](#)

11.25 EFI_HOB_MEMORY_ALLOCATION_HEADER Struct Reference

[EFI_HOB_MEMORY_ALLOCATION_HEADER](#) describes the various attributes of the logical memory allocation.

```
#include <PiHob.h>
```

Collaboration diagram for [EFI_HOB_MEMORY_ALLOCATION_HEADER](#):



Public Attributes

- [EFI_GUID](#) **Name**
A *GUID* that defines the memory allocation region's type and purpose, as well as other fields within the memory allocation HOB.
- [EFI_PHYSICAL_ADDRESS](#) **MemoryBaseAddress**
The base address of memory allocated by this HOB.
- [UINT64](#) **MemoryLength**
The length in bytes of memory allocated by this HOB.
- [EFI_MEMORY_TYPE](#) **MemoryType**
Defines the type of memory allocated by this HOB.
- [UINT8](#) **Reserved** [4]
Padding for Itanium processor family.

11.25.1 Detailed Description

[EFI_HOB_MEMORY_ALLOCATION_HEADER](#) describes the various attributes of the logical memory allocation.

The type field will be used for subsequent inclusion in the UEFI memory map.

Definition at line 105 of file PiHob.h.

11.25.2 Member Data Documentation

11.25.2.1 EFI_PHYSICAL_ADDRESS EFI_HOB_MEMORY_ALLOCATION_HEADER::MemoryBaseAddress

The base address of memory allocated by this HOB.

Type `EFI_PHYSICAL_ADDRESS` is defined in `AllocatePages()` in the UEFI 2.0 specification.

Definition at line 120 of file PiHob.h.

11.25.2.2 EFI_MEMORY_TYPE EFI_HOB_MEMORY_ALLOCATION_HEADER::MemoryType

Defines the type of memory allocated by this HOB.

The memory type definition follows the `EFI_MEMORY_TYPE` definition. Type `EFI_MEMORY_TYPE` is defined in `AllocatePages()` in the UEFI 2.0 specification.

Definition at line 132 of file PiHob.h.

11.25.2.3 EFI_GUID EFI_HOB_MEMORY_ALLOCATION_HEADER::Name

A [GUID](#) that defines the memory allocation region's type and purpose, as well as other fields within the memory allocation HOB.

This [GUID](#) is used to define the additional data within the HOB that may be present for the memory allocation HOB. Type `EFI_GUID` is defined in `InstallProtocolInterface()` in the UEFI 2.0 specification.

Definition at line 113 of file PiHob.h.

The documentation for this struct was generated from the following file:

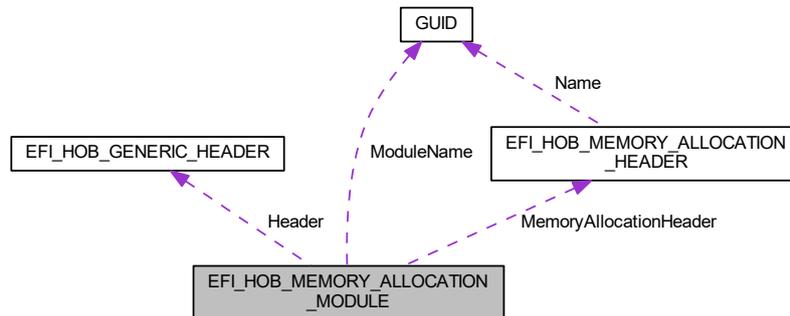
- [PiHob.h](#)

11.26 EFI_HOB_MEMORY_ALLOCATION_MODULE Struct Reference

Defines the location and entry point of the HOB consumer phase.

```
#include <PiHob.h>
```

Collaboration diagram for `EFI_HOB_MEMORY_ALLOCATION_MODULE`:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)
The HOB generic header.
- [EFI_HOB_MEMORY_ALLOCATION_HEADER MemoryAllocationHeader](#)
An instance of the [EFI_HOB_MEMORY_ALLOCATION_HEADER](#) that describes the various attributes of the logical memory allocation.
- [EFI_GUID ModuleName](#)
The [GUID](#) specifying the values of the firmware file system name that contains the HOB consumer phase component.
- [EFI_PHYSICAL_ADDRESS EntryPoint](#)
The address of the memory-mapped firmware volume that contains the HOB consumer phase firmware file.

11.26.1 Detailed Description

Defines the location and entry point of the HOB consumer phase.

Definition at line 200 of file PiHob.h.

11.26.2 Member Data Documentation

11.26.2.1 `EFI_HOB_GENERIC_HEADER` `EFI_HOB_MEMORY_ALLOCATION_MODULE::Header`

The HOB generic header.

`Header.HobType = EFI_HOB_TYPE_MEMORY_ALLOCATION.`

Definition at line 204 of file PiHob.h.

The documentation for this struct was generated from the following file:

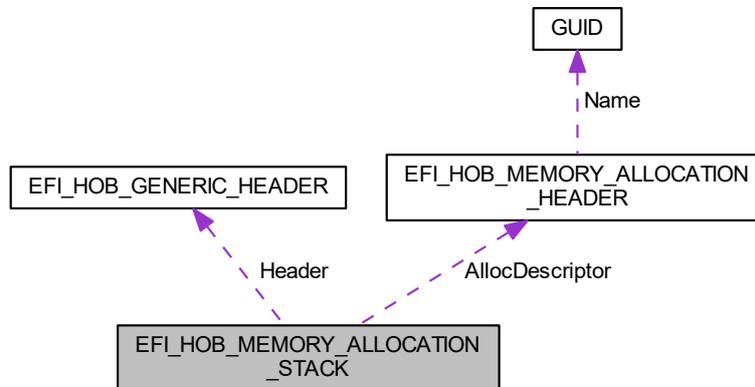
- [PiHob.h](#)

11.27 `EFI_HOB_MEMORY_ALLOCATION_STACK` Struct Reference

Describes the memory stack that is produced by the HOB producer phase and upon which all post-memory-installed executable content in the HOB producer phase is executing.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_MEMORY_ALLOCATION_STACK:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)

The HOB generic header.

- [EFI_HOB_MEMORY_ALLOCATION_HEADER AllocDescriptor](#)

An instance of the [EFI_HOB_MEMORY_ALLOCATION_HEADER](#) that describes the various attributes of the logical memory allocation.

11.27.1 Detailed Description

Describes the memory stack that is produced by the HOB producer phase and upon which all post-memory-installed executable content in the HOB producer phase is executing.

Definition at line 167 of file PiHob.h.

11.27.2 Member Data Documentation

11.27.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_MEMORY_ALLOCATION_STACK::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_MEMORY_ALLOCATION.

Definition at line 171 of file PiHob.h.

The documentation for this struct was generated from the following file:

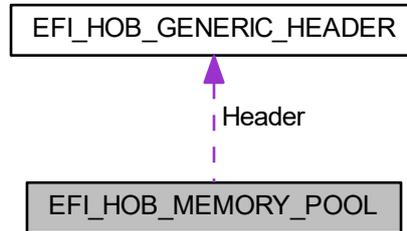
- [PiHob.h](#)

11.28 EFI_HOB_MEMORY_POOL Struct Reference

Describes pool memory allocations.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_MEMORY_POOL:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)

The HOB generic header.

11.28.1 Detailed Description

Describes pool memory allocations.

Definition at line 461 of file PiHob.h.

11.28.2 Member Data Documentation

11.28.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_MEMORY_POOL::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_MEMORY_POOL.

Definition at line 465 of file PiHob.h.

The documentation for this struct was generated from the following file:

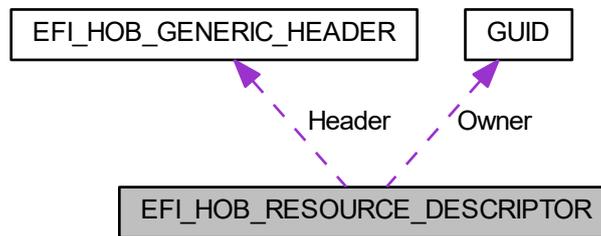
- [PiHob.h](#)

11.29 EFI_HOB_RESOURCE_DESCRIPTOR Struct Reference

Describes the resource properties of all fixed, nonrelocatable resource ranges found on the processor host bus during the HOB producer phase.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_RESOURCE_DESCRIPTOR:



Public Attributes

- [EFI_HOB_GENERIC_HEADER](#) Header
The HOB generic header.
- [EFI_GUID](#) Owner
A GUID representing the owner of the resource.
- [EFI_RESOURCE_TYPE](#) ResourceType
The resource type enumeration as defined by EFI_RESOURCE_TYPE.
- [EFI_RESOURCE_ATTRIBUTE_TYPE](#) ResourceAttribute
Resource attributes as defined by EFI_RESOURCE_ATTRIBUTE_TYPE.
- [EFI_PHYSICAL_ADDRESS](#) PhysicalStart
The physical start address of the resource region.
- [UINT64](#) ResourceLength
The number of bytes of the resource region.

11.29.1 Detailed Description

Describes the resource properties of all fixed, nonrelocatable resource ranges found on the processor host bus during the HOB producer phase.

Definition at line 306 of file PiHob.h.

11.29.2 Member Data Documentation

11.29.2.1 EFI_HOB_GENERIC_HEADER EFI_HOB_RESOURCE_DESCRIPTOR::Header

The HOB generic header.

Header.HobType = EFI_HOB_TYPE_RESOURCE_DESCRIPTOR.

Definition at line 310 of file PiHob.h.

11.29.2.2 EFI_GUID EFI_HOB_RESOURCE_DESCRIPTOR::Owner

A GUID representing the owner of the resource.

This GUID is used by HOB consumer phase components to correlate device ownership of a resource.

Definition at line 315 of file PiHob.h.

The documentation for this struct was generated from the following file:

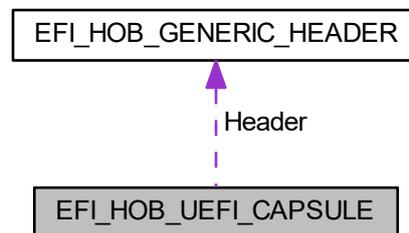
- [PiHob.h](#)

11.30 EFI_HOB_UEFI_CAPSULE Struct Reference

Each UEFI capsule HOB details the location of a UEFI capsule.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_HOB_UEFI_CAPSULE:



Public Attributes

- [EFI_HOB_GENERIC_HEADER Header](#)
The HOB generic header where Header.HobType = EFI_HOB_TYPE_UEFI_CAPSULE.
- [EFI_PHYSICAL_ADDRESS BaseAddress](#)
The physical memory-mapped base address of an UEFI capsule.

11.30.1 Detailed Description

Each UEFI capsule HOB details the location of a UEFI capsule.

It includes a base address and length which is based upon memory blocks with a `EFI_CAPSULE_HEADER` and the associated `CapsuleImageSize`-based payloads. These HOB's shall be created by the PEI PI firmware sometime after the UEFI UpdateCapsule service invocation with the `CAPSULE_FLAGS_POPULATE_SYSTEM_TABLE` flag set in the `EFI_CAPSULE_HEADER`.

Definition at line 475 of file PiHob.h.

11.30.2 Member Data Documentation

11.30.2.1 `EFI_PHYSICAL_ADDRESS` `EFI_HOB_UEFI_CAPSULE::BaseAddress`

The physical memory-mapped base address of an UEFI capsule.

This value is set to point to the base of the contiguous memory of the UEFI capsule. The length of the contiguous memory in bytes.

Definition at line 486 of file PiHob.h.

The documentation for this struct was generated from the following file:

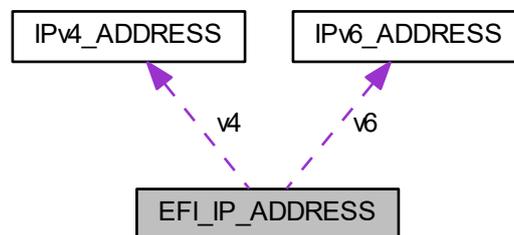
- [PiHob.h](#)

11.31 EFI_IP_ADDRESS Union Reference

16-byte buffer aligned on a 4-byte boundary.

```
#include <UefiBaseType.h>
```

Collaboration diagram for EFI_IP_ADDRESS:



11.31.1 Detailed Description

16-byte buffer aligned on a 4-byte boundary.

An IPv4 or IPv6 internet protocol address.

Definition at line 103 of file UefiBaseType.h.

The documentation for this union was generated from the following file:

- [UefiBaseType.h](#)

11.32 EFI_MAC_ADDRESS Struct Reference

32-byte buffer containing a network Media Access Control address.

```
#include <UefiBaseType.h>
```

11.32.1 Detailed Description

32-byte buffer containing a network Media Access Control address.

Definition at line 95 of file UefiBaseType.h.

The documentation for this struct was generated from the following file:

- [UefiBaseType.h](#)
-

11.33 EFI_MMRAM_DESCRIPTOR Struct Reference

Structure describing a MMRAM region and its accessibility attributes.

```
#include <PiMultiPhase.h>
```

Public Attributes

- [EFI_PHYSICAL_ADDRESS PhysicalStart](#)
Designates the physical address of the MMRAM in memory.
- [EFI_PHYSICAL_ADDRESS CpuStart](#)
Designates the address of the MMRAM, as seen by software executing on the processors.
- [UINT64 PhysicalSize](#)
Describes the number of bytes in the MMRAM region.
- [UINT64 RegionState](#)
Describes the accessibility attributes of the MMRAM.

11.33.1 Detailed Description

Structure describing a MMRAM region and its accessibility attributes.

Definition at line 109 of file PiMultiPhase.h.

11.33.2 Member Data Documentation

11.33.2.1 EFI_PHYSICAL_ADDRESS EFI_MMRAM_DESCRIPTOR::CpuStart

Designates the address of the MMRAM, as seen by software executing on the processors.

This address may or may not match PhysicalStart.

Definition at line 120 of file PiMultiPhase.h.

11.33.2.2 EFI_PHYSICAL_ADDRESS EFI_MMRAM_DESCRIPTOR::PhysicalStart

Designates the physical address of the MMRAM in memory.

This view of memory is the same as seen by I/O-based agents, for example, but it may not be the address seen by the processors.

Definition at line 115 of file PiMultiPhase.h.

11.33.2.3 UINT64 EFI_MMRAM_DESCRIPTOR::RegionState

Describes the accessibility attributes of the MMRAM.

These attributes include the hardware state (e.g., Open/Closed/Locked), capability (e.g., cacheable), logical allocation (e.g., allocated), and pre-use initialization (e.g., needs testing/ECC initialization).

Definition at line 131 of file PiMultiPhase.h.

The documentation for this struct was generated from the following file:

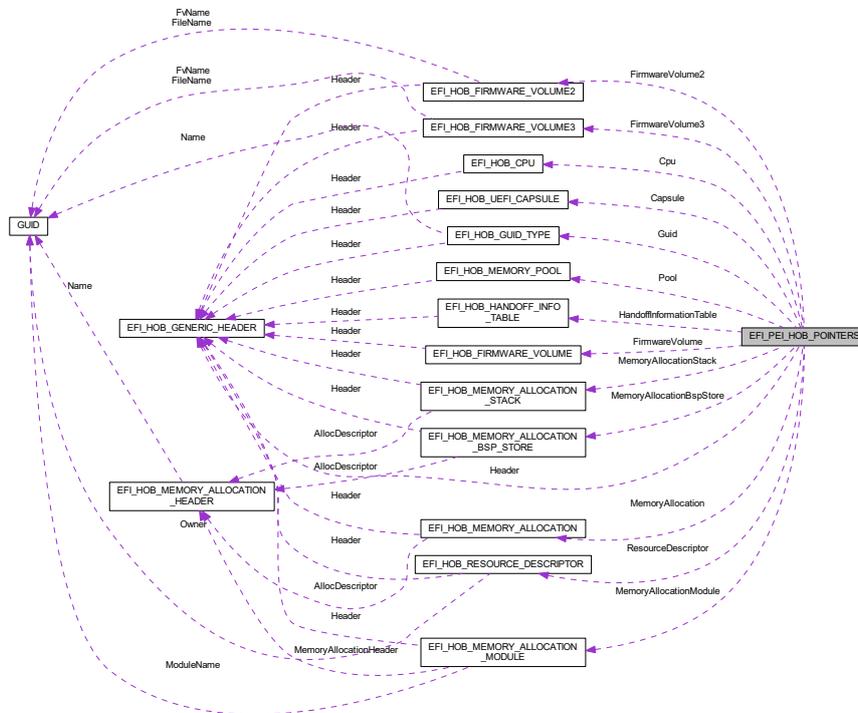
- [PiMultiPhase.h](#)
-

11.34 EFI_PEI_HOB_POINTERS Union Reference

Union of all the possible HOB Types.

```
#include <PiHob.h>
```

Collaboration diagram for EFI_PEI_HOB_POINTERS:



11.34.1 Detailed Description

Union of all the possible HOB Types.

Definition at line 493 of file PiHob.h.

The documentation for this union was generated from the following file:

- [PiHob.h](#)

11.35 EFI_TIME Struct Reference

EFI Time Abstraction: Year: 1900 - 9999 Month: 1 - 12 Day: 1 - 31 Hour: 0 - 23 Minute: 0 - 59 Second: 0 - 59 Nanosecond: 0 - 999,999,999 TimeZone: -1440 to 1440 or 2047.

```
#include <UefiBaseType.h>
```

11.35.1 Detailed Description

EFI Time Abstraction: Year: 1900 - 9999 Month: 1 - 12 Day: 1 - 31 Hour: 0 - 23 Minute: 0 - 59 Second: 0 - 59 Nanosecond: 0 - 999,999,999 TimeZone: -1440 to 1440 or 2047.

Definition at line 67 of file UefiBaseType.h.

The documentation for this struct was generated from the following file:

- [UefiBaseType.h](#)

11.36 FSPM_CONFIG Struct Reference

FSP-M Configuration.

```
#include <FspmUpd.h>
```

Public Attributes

- UINT8 [CustomerRevision](#) [32]
Offset 0x0040 - Customer Revision The Customer can set this revision string for their own purpose.
- UINT8 [BusRatio](#) [8]
Offset 0x0060 - Bus Ratio Indicates the ratio of Bus/MMIOL/IO resource to be allocated for each CPU's IIO.
- UINT8 [D2KCreditConfig](#)
Offset 0x0068 - D2K Credit Config Set the D2K Credit Config.
- UINT8 [SnoopThrottleConfig](#)
Offset 0x0069 - Snoop Throttle Config Set the Snoop Throttle Config.
- UINT8 [SnoopAllCores](#)
Offset 0x006A - Snoop Throttle Config Set the Snoop All Core Config.
- UINT8 [LegacyVgaSoc](#)
Offset 0x006B - Legacy VGA Socket Socket that claims the legacy VGA range.
- UINT8 [LegacyVgaStack](#)
Offset 0x006C - Legacy VGA Stack Stack that claims the legacy VGA range.
- UINT8 [P2pRelaxedOrdering](#)
*Offset 0x006D - Pcie P2P Performance Mode Enable: Enable PCIe P2P Performance Mode, **Disable(Default)**↔ : Disable PCIe P2P Performance Mode \$EN_DIS.*
- UINT8 [DebugPrintLevel](#)
Offset 0x006E - UPI Debug Print Level UPI Debug Print Level Bitmask.
- UINT8 [SncEn](#)
*Offset 0x006F - SNC **Enable(Default)** or Disable SNC \$EN_DIS.*
- UINT8 [UmaClustering](#)
Offset 0x0070 - UMA Clustering Set number of enabled UMA Clusters.
- UINT8 [IoDcMode](#)
Offset 0x0071 - IODC Mode IODC Mode.
- UINT8 [DegradPrecedence](#)
Offset 0x0072 - Degrad Precedence Degrad Precedence.
- UINT8 [Degrad4SPreference](#)
Offset 0x0073 - Degrad 4 Socket Preference Degrad 4 Socket Preference.
- UINT8 [DirectoryModeEn](#)
*Offset 0x0074 - Directory Mode **Enable(Default)** or Disable Directory Mode \$EN_DIS.*
- UINT8 [XptPrefetchEn](#)
Offset 0x0075 - XPT Prefetch Enable XPT Prefetch.
- UINT8 [KtiPrefetchEn](#)
*Offset 0x0076 - KTI Prefetch Enable **Enable(Default)** or Disable KTI Prefetch \$EN_DIS.*
- UINT8 [XptRemotePrefetchEn](#)
*Offset 0x0077 - XPT Remote Prefetch Enable Enable or **Disable(Default)** XPT Remote Prefetch \$EN_DIS.*
- UINT8 [KtiFpgaEnable](#) [8]
Offset 0x0078 - KTI FPGA Enable or Disable KTI FPGA, Default: 0x1 (Enable) \$EN_DIS.

- UINT8 [DdrtQoSMode](#)
Offset 0x0080 - DDRT QoS Mode DDRT QoS.
 - UINT8 [KtiLinkSpeedMode](#)
Offset 0x0081 - KTI Link Speed Mode KTI Link Speed Mode.
 - UINT8 [KtiLinkSpeed](#)
Offset 0x0082 - KTI Link Speed KTI Link Speed.
 - UINT8 [KtiLinkL0pEn](#)
Offset 0x0083 - KTI Link L0p KTI Link L0p.
 - UINT8 [KtiLinkL1En](#)
Offset 0x0084 - KTI Link L1 KTI Link L1.
 - UINT8 [KtiFailoverEn](#)
Offset 0x0085 - KTI Failover KTI Failover.
 - UINT8 [KtiLbEn](#)
*Offset 0x0086 - KTI LB Enable Enable or **Disable(Default)** KTI LB \$EN_DIS.*
 - UINT8 [KtiCrcMode](#)
Offset 0x0087 - KTI CRC Mode KTI CRC Mode.
 - UINT8 [KtiCpuSktHotPlugEn](#)
*Offset 0x0088 - KTI CPU Socket Hotplug Enable or **Disable(Default)** KTI CPU Socket Hotplug \$EN_DIS.*
 - UINT8 [KtiCpuSktHotPlugTopology](#)
Offset 0x0089 - KTI CPU Socket HotPlug Topology KTI CPU Socket HotPlug Topology.
 - UINT8 [KtiSkuMismatchCheck](#)
*Offset 0x008A - KTI SKU Mismatch Check **Enable(Default)** or Disable KTI SKU Mismatch Check \$EN_DIS.*
 - UINT8 [IrqThreshold](#)
Offset 0x008B - IRQ Threshold IRQ Threshold.
 - UINT8 [TorThresLoctoremNorm](#)
Offset 0x008C - TOR threshold - Loctorem threshold normal TOR threshold - Loctorem threshold normal.
 - UINT8 [TorThresLoctoremEmpty](#)
Offset 0x008D - TOR threshold - Loctorem threshold empty TOR threshold - Loctorem threshold empty.
 - UINT8 [MbeBwCal](#)
Offset 0x008E - MBA BW Calibration MBA BW Calibration setting.
 - UINT8 [TscSyncEn](#)
Offset 0x008F - TSC Sync in Sockets TSC Sync in Sockets.
 - UINT8 [StaleAtoSOptEn](#)
Offset 0x0090 - HA A to S directory optimization HA A to S directory optimization.
 - UINT8 [LLCDeadlineAlloc](#)
*Offset 0x0091 - LLC Deadline Allocation **Enable(Default)** or Disable LLC Deadline Allocation \$EN_DIS.*
 - UINT8 [SplitLock](#)
Offset 0x0092 - Split Lock Split Lock.
 - UINT8 [mmCfgBase](#)
Offset 0x0093 - MMCFG Base Address MMCFG Base Address.
 - UINT8 [mmCfgSize](#)
Offset 0x0094 - MMCFG Size Select MMCFG Size.
 - UINT8 [UnusedUpdSpace0](#) [3]
Offset 0x0095.
 - UINT32 [mmiohBase](#)
Offset 0x0098 - MMIO High Base Address MMIO High Base Address, a hex number for Bit[51:32].
 - UINT8 [CpuPaLimit](#)
*Offset 0x009C - CPU Physical Address Limit **Enable(Default)** or Disable CPU Physical Address Limit 0:Disable, 1:Enable.*
 - UINT8 [highGap](#)
*Offset 0x009D - High Gap Enable or **Disable(Default)** High Gap \$EN_DIS.*
-

- UINT16 [mmiohSize](#)
Offset 0x009E - MMIO High Size MMIO High Size, Number of 1GB contiguous regions to be assigned for MMIOH space per CPU.
 - UINT8 [isocEn](#)
*Offset 0x00A0 - ISOC **Enable(Default)** or Disable ISOC \$EN_DIS.*
 - UINT8 [dcaEn](#)
*Offset 0x00A1 - DCA Enable or **Disable(Default)** DCA \$EN_DIS.*
 - UINT8 [UnusedUpdSpace1](#) [2]
Offset 0x00A2.
 - UINT32 [BoardTypeBitmask](#)
Offset 0x00A4 - BoardTypeBitmask Board Type Bitmask.
 - UINT32 [AllLanesPtr](#)
Offset 0x00A8 - AllLanesPtr Pointer to array of [ALL_LANES_EPARAM_LINK_INFO](#).
 - UINT32 [PerLanePtr](#)
Offset 0x00AC - PerLanePtr Pointer to array of [PER_LANE_EPARAM_LINK_INFO](#).
 - UINT32 [AllLanesSizeOfTable](#)
Offset 0x00B0 - AllLanesSizeOfTable Number of elements in AllLanesPtr array.
 - UINT32 [PerLaneSizeOfTable](#)
Offset 0x00B4 - PerLaneSizeOfTable Number of elements in PerLanePtr array.
 - UINT32 [WaitTimeForPSBP](#)
Offset 0x00B8 - WaitTimeForPSBP Number of milliseconds to wait for remote CPUs to initialize.
 - UINT8 [IsKtiNvramDataReady](#)
Offset 0x00BC - IsKtiNvramDataReady DEPRECATED.
 - UINT8 [BoardId](#)
Offset 0x00BD - BoardId Board ID.
 - UINT8 [WaSerializationEn](#)
*Offset 0x00BE - WaSerializationEn **Enable(Default)** or Disable BIOS serialization WA \$EN_DIS.*
 - UINT8 [KtiInEnableMktme](#)
Offset 0x00BF - KtiInEnableMktme Enable(Default) or Disable Mktme status decides D2Kti feature state \$EN_DIS.
 - UINT8 [VmxEnable](#)
Offset 0x00C0 - Processor VmxEnable Function Enable(Default) or Disable Processor VmxEnable Function \$EN_↔DIS.
 - UINT8 [X2apic](#)
Offset 0x00C1 - Processor X2apic Function Enable(Default) or Disable Processor X2apic Function \$EN_DIS.
 - UINT8 [DdrFreqLimit](#)
Offset 0x00C2 - DDR frequency limit Enable(Default) or Disable Processor X2apic Function.
 - UINT8 [serialDebugMsgLvl](#)
*Offset 0x00C3 - Memory Serial Debug Message Level Select Memory Serial Debug Message Level, 0: Disable, 1: Minimum, 2: Normal, **3: Maximum(Default)**, 4: Auto 0:Disable, 1:Minimum, 2:Normal, 3:Maximum, 4:Auto.*
 - UINT8 [lioConfigIOU0](#) [8]
*Offset 0x00C4 - IIO ConfigIOU0 ConfigIOU[MAX_SOCKET][0]: MAX_SOCKET=8, 0x00:x4x4x4x4, 0x01:x4x4xxx8, 0x02:xxx8x4x4, 0x03:xxx8xxx8, 0x04:xxxxxx16, **0xFF:AUTO(Default)***
 - UINT8 [lioConfigIOU1](#) [8]
*Offset 0x00CC - IIO ConfigIOU1 ConfigIOU[MAX_SOCKET][1]: MAX_SOCKET=8, 0x00:x4x4x4x4, 0x01:x4x4xxx8, 0x02:xxx8x4x4, 0x03:xxx8xxx8, 0x04:xxxxxx16, **0xFF:AUTO(Default)***
 - UINT8 [lioConfigIOU2](#) [8]
*Offset 0x00D4 - IIO ConfigIOU2 ConfigIOU[MAX_SOCKET][2]: MAX_SOCKET=8, 0x00:x4x4x4x4, 0x01:x4x4xxx8, 0x02:xxx8x4x4, 0x03:xxx8xxx8, 0x04:xxxxxx16, **0xFF:AUTO(Default)***
 - UINT8 [lioConfigIOU3](#) [8]
*Offset 0x00DC - IIO ConfigIOU3 ConfigIOU[MAX_SOCKET][3]: MAX_SOCKET=8, 0x00:x4x4x4x4, 0x01:x4x4xxx8, 0x02:xxx8x4x4, 0x03:xxx8xxx8, 0x04:xxxxxx16, **0xFF:AUTO(Default)***
 - UINT8 [lioConfigIOU4](#) [8]
-

- Offset 0x00E4 - IIO ConfigIOU4 ConfigIOU[MAX_SOCKET][4]: MAX_SOCKET=8, 0x00:x4x4x4x4, 0x01:x4x4xxx8, 0x02:xxx8x4x4, 0x03:xxx8xxx8, 0x04:xxxxxx16, **0xFF:AUTO(Default)**
- UUINT32 [lioPcieConfigTablePtr](#)
Offset 0x00EC - IIO PCIE Config Table Ptr Pointer to array of UPD_IIO_PCIE_PORT_CONFIG.
 - UUINT32 [lioPcieConfigTableNumber](#)
Offset 0x00F0 - IIO PCIE Config Table Number Number of elements in lioPcieConfigTablePtr array.
 - UUINT8 [IIOpcieRootPortEnable](#)
Offset 0x00F4 - IIO PCIE Root Port Enable **Enable(Default)** or Disable IIO PCH rootport.
 - UUINT8 [DeEmphasis](#)
Offset 0x00F5 - IIO DeEmphasis IIO DeEmphasis.
 - UUINT8 [IIOpciePortLinkSpeed](#)
Offset 0x00F6 - IIO PCIe Root Port Link Speed IIO PCIe Root Port Link Speed.
 - UUINT8 [IIOpcieMaxPayload](#)
Offset 0x00F7 - IIO PCIe Root Port Max Payload IIO PCIe Root Port Max Payload.
 - UUINT8 [DfxDnTxPreset](#)
Offset 0x00F8 - IIO DfxDnTxPreset IIO Downstream Transmitter Preset.
 - UUINT8 [DfxRxPreset](#)
Offset 0x00F9 - IIO DfxRxPreset IIO Downstream Receiver Preset.
 - UUINT8 [DfxUpTxPreset](#)
Offset 0x00FA - IIO DfxUpTxPreset IIO Upstream Transmitter Preset.
 - UUINT8 [PcieCommonClock](#)
Offset 0x00FB - IIO PCIe Common Clock IIO PCIe Common Clock.
 - UUINT8 [NtbPpd](#)
Offset 0x00FC - IIO Non-Transparent Port Definition IIO Non-Transparent Port Definition.
 - UUINT8 [NtbBarSizeOverride](#)
Offset 0x00FD - IIO Non-Transparent Bridge BAR Size Override Enable or **Disable(Default)** IIO Non-Transparent Bridge BAR Size Override.
 - UUINT8 [NtbSplitBar](#)
Offset 0x00FE - IIO Non-Transparent Bridge Split BAR Mode Enable or **Disable(Default)** IIO Non-Transparent Bridge Split BAR Mode.
 - UUINT8 [NtbBarSizeImBar1](#)
Offset 0x00FF - IIO NtbBarSizeImBar1 IIO NtbBarSizeImBar1.
 - UUINT8 [NtbBarSizeImBar2](#)
Offset 0x0100 - IIO NtbBarSizeImBar2 IIO PNtbBarSizeImBar2.
 - UUINT8 [NtbBarSizeImBar2_0](#)
Offset 0x0101 - IIO NtbBarSizeImBar2_0 IIO PNtbBarSizeImBar2_0.
 - UUINT8 [NtbBarSizeImBar2_1](#)
Offset 0x0102 - IIO NtbBarSizeImBar2_1 IIO NtbBarSizeImBar2_1.
 - UUINT8 [NtbBarSizeEmBarSZ1](#)
Offset 0x0103 - IIO NtbBarSizeEmBarSZ1 IIO NtbBarSizeEmBarSZ1.
 - UUINT8 [NtbBarSizeEmBarSZ2](#)
Offset 0x0104 - IIO NtbBarSizeEmBarSZ2 IIO NtbBarSizeEmBarSZ2.
 - UUINT8 [NtbBarSizeEmBarSZ2_0](#)
Offset 0x0105 - IIO NtbBarSizeEmBarSZ2_0 IIO NtbBarSizeEmBarSZ2_0.
 - UUINT8 [NtbBarSizeEmBarSZ2_1](#)
Offset 0x0106 - IIO NtbBarSizeEmBarSZ2_1 IIO NtbBarSizeEmBarSZ2_1.
 - UUINT8 [NtbXlinkCtlOverride](#)
Offset 0x0107 - IIO Non-Transparent Cross Link Override IIO Non-Transparent Cross Link Override.
 - UUINT8 [VtdSupport](#)
Offset 0x0108 - VT-d Support Enable or **Disable(Default)** VT-d Support \$EN_DIS.
 - UUINT8 [PEXP HIDE](#)
-

- Offset 0x0109 - IIO PCIe Port Hide Hide or visible for IIO Pcie Port, 1 : Hide, 0 : Visible.*

 - UINT8 [HidePEXPMenu](#)
- Offset 0x010A - IIO Pcie Port Menu Hide Hide or visible for IIO PCIe Port Menu, 1 : Hide, 0 : Visible.*

 - UINT8 [PchSirqMode](#)
- Offset 0x010B - PchSirqMode PchSirqMode.*

 - UINT8 [PchAdrEn](#)
- Offset 0x010C - PchAdrEn PchAdr 0:PLATFORM POR, 1:FORCE ENABLE(Default), 2:FORCE DISABLE 0:PLA↔TFORM POR, 1:FORCE ENABLE, 2:FORCE DISABLE.*

 - UINT8 [ThermalDeviceEnable](#)
- Offset 0x010D - ThermalDeviceEnable Thermal Device Mode.*

 - UINT8 [PchPcieRootPortFunctionSwap](#)
- Offset 0x010E - PchPcieRootPortFunctionSwap Root port swapping based on device connection status : 1↔: Enable(Default), 0: Disable \$EN_DIS.*

 - UINT8 [PchPciePllSsc](#)
- Offset 0x010F - PCH PCIE PLL Ssc Valid spread range : 0x00-0x14 (A value of 0 is SSC of 0.0%.*

 - UINT8 [PchPciePortIndex](#) [20]
- Offset 0x0110 - PCH PCIE Root Port Index Index assigned to every PCH PCIE Root Port.*

 - UINT8 [PchPcieForceEnable](#) [20]
- Offset 0x0124 - PCH PCIE Root Port Enable or Disable 0-19: PCH rootport, if port is enabled(Default), the value is 0x01, if the port is disabled, the value is 0x00.*

 - UINT8 [PchPciePortLinkSpeed](#) [20]
- Offset 0x0138 - PCH PCIE Root Port Link Speed 0-19: PCH rootport, 0x00 : Pcie Auto Speed(Default), 0x01 : Pcie Gen1 Speed, 0x02 : Pcie Gen2 Speed, 0x03 : Pcie Gen3 Speed.*

 - UINT8 [PchDciEn](#)
- Offset 0x014C - PchDciEn Enable or Disable(Default) PCH DCI 0:Disable, 1:Enable.*

 - UINT8 [MeUmaEnable](#)
- Offset 0x014D - MeUmaEnable Enable or disable ME UMA feature 0:Disable, 1:Enable.*

 - UINT8 [SerialloUartDebugEnabled](#)
- Offset 0x014E - SerialloUartDebugEnabled Enable(Default) or Disable Seriallo Uart debug library in FSP.*

 - UINT8 [UnusedUpdSpace2](#)
- Offset 0x014F.*

 - UINT16 [SerialloUartDebugloBase](#)
- Offset 0x0150 - ISA Serial Base selection Select ISA Serial Base address could be initialized by boot loader.*

 - UINT8 [UnusedUpdSpace3](#) [2]
- Offset 0x0152.*

 - UINT8 [ReservedMemoryInitUpd](#) [16]
- Offset 0x0154.*

11.36.1 Detailed Description

FSP-M Configuration.

Definition at line 43 of file FspmUpd.h.

11.36.2 Member Data Documentation

11.36.2.1 UINT32 FSPM_CONFIG::BoardTypeBitmask

Offset 0x00A4 - BoardTypeBitmask Board Type Bitmask.

Default: 0x1

Definition at line 336 of file FspmUpd.h.

11.36.2.2 UINT8 FSPM_CONFIG::BusRatio[8]

Offset 0x0060 - Bus Ratio Indicates the ratio of Bus/MMIOL/IO resource to be allocated for each CPU's IIO.

Default 0x1

Definition at line 54 of file FspmUpd.h.

11.36.2.3 UINT8 FSPM_CONFIG::D2KCreditConfig

Offset 0x0068 - D2K Credit Config Set the D2K Credit Config.

1: Min, 2: Med (Default), 3: Max. 1:Min, 2:Med, 3:Max

Definition at line 60 of file FspmUpd.h.

11.36.2.4 UINT8 FSPM_CONFIG::DdrtQosMode

Offset 0x0080 - DDRT QoS Mode DDRT QoS.

0: Mode 0(Default), 1: Mode 1, 2: Mode 2 0:Mode 0, 1:Mode 1, 2:Mode 2)

Definition at line 163 of file FspmUpd.h.

11.36.2.5 UINT8 FSPM_CONFIG::DebugPrintLevel

Offset 0x006E - UPI Debug Print Level UPI Debug Print Level Bitmask.

0: Disable, 1: Fatal, 2: Warning, 4: Summary, 8: Detail, 0xF: All(Default) 1:Fatal, 2:Warning, 4:Summary, 8:Detail, 0x0F:All

Definition at line 96 of file FspmUpd.h.

11.36.2.6 UINT8 FSPM_CONFIG::DeEmphasis

Offset 0x00F5 - IIO DeEmphasis IIO DeEmphasis.

Default: 0x1

Definition at line 458 of file FspmUpd.h.

11.36.2.7 UINT8 FSPM_CONFIG::Degrade4SPreference

Offset 0x0073 - Degrade 4 Socket Preference Degrade 4 Socket Preference.

0: Fully Connect(Default), 1: Dual Link Ring 0:Fully Connect, 1:Dual Link Ring

Definition at line 127 of file FspmUpd.h.

11.36.2.8 UINT8 FSPM_CONFIG::DegradePrecedence

Offset 0x0072 - Degrade Precedence Degrade Precedence.

0: Topology(Default), 1: Feature 0:Topology, 1:Feature

Definition at line 121 of file FspmUpd.h.

11.36.2.9 UINT8 FSPM_CONFIG::DfxDnTxPreset

Offset 0x00F8 - IIO DfxDnTxPreset IIO Downstream Transmitter Preset.

Default: Auto(0xFF), otherwise preset 0-10

Definition at line 475 of file FspmUpd.h.

11.36.2.10 UINT8 FSPM_CONFIG::DfxRxPreset

Offset 0x00F9 - IIO DfxRxPreset IIO Downstream Receiver Preset.

Default: Auto(0xFF), otherwise preset 0-10

Definition at line 480 of file FspmUpd.h.

11.36.2.11 UINT8 FSPM_CONFIG::DfxUpTxPreset

Offset 0x00FA - IIO DfxUpTxPreset IIO Upstream Transmitter Preset.

Default: Auto(0xFF), otherwise preset 0-10

Definition at line 485 of file FspmUpd.h.

11.36.2.12 UINT8 FSPM_CONFIG::IOPcieMaxPayload

Offset 0x00F7 - IIO PCIe Root Port Max Payload IIO PCIe Root Port Max Payload.

0: 128B, 1: 256B, 2: 512B, 7: **Auto(Default)** 0:128B, 1: 256B, 2:512B, 7:Auto

Definition at line 470 of file FspmUpd.h.

11.36.2.13 UINT8 FSPM_CONFIG::IOPciePortLinkSpeed

Offset 0x00F6 - IIO PCIe Root Port Link Speed IIO PCIe Root Port Link Speed.

0: Auto(Default), 1: Gen1, 2: Gen2, 3: Gen3, 4: Gen4 0:Auto, 1:Gen1, 2:Gen2, 3:Gen3, 4:Gen4

Definition at line 464 of file FspmUpd.h.

11.36.2.14 UINT8 FSPM_CONFIG::IoDcMode

Offset 0x0071 - IODC Mode IODC Mode.

0: Disable, **1: Auto(Default)**, 2: Push, 3: AllocFlow, 4: NonAlloc, 5: WCILF 0:Disable, 1:Auto, 2:Push, 3:AllocFlow, 4:NonAlloc, 5:WCILF

Definition at line 115 of file FspmUpd.h.

11.36.2.15 UINT8 FSPM_CONFIG::IrqThreshold

Offset 0x008B - IRQ Threshold IRQ Threshold.

0: Disable, **1: Auto(Default)**, 2: Low, 3: Medium, 4: High 0:Disable, 1:Auto, 2:Low, 3:Medium, 4:High

Definition at line 230 of file FspmUpd.h.

11.36.2.16 UINT8 FSPM_CONFIG::IsKtiNvramDataReady

Offset 0x00BC - IsKtiNvramDataReady DEPRECATED.

Set to zero. \$EN_DIS

Definition at line 367 of file FspmUpd.h.

11.36.2.17 UUINT8 FSPM_CONFIG::KtiCpuSktHotPlugTopology

Offset 0x0089 - KTI CPU Socket HotPlug Topology KTI CPU Socket HotPlug Topology.

0: 4 Socket(Default), 1: 8 Socket 0:4Socket, 1:8Socket

Definition at line 218 of file FspmUpd.h.

11.36.2.18 UUINT8 FSPM_CONFIG::KtiCrcMode

Offset 0x0087 - KTI CRC Mode KTI CRC Mode.

0: 16bit, 1: 32bit, 2: Auto(Default) 0:16bit, 1:32bit, 2:Auto

Definition at line 206 of file FspmUpd.h.

11.36.2.19 UUINT8 FSPM_CONFIG::KtiFailoverEn

Offset 0x0085 - KTI Failover KTI Failover.

0: Disable, 1: Enable, 2: Auto(Default) 0:Disable, 1:Enable, 2: Auto

Definition at line 194 of file FspmUpd.h.

11.36.2.20 UUINT8 FSPM_CONFIG::KtiLinkL0pEn

Offset 0x0083 - KTI Link L0p KTI Link L0p.

0: Disable, 1: Enable, 2: Auto(Default) 0:Disable, 1:Enable, 2: Auto

Definition at line 182 of file FspmUpd.h.

11.36.2.21 UUINT8 FSPM_CONFIG::KtiLinkL1En

Offset 0x0084 - KTI Link L1 KTI Link L1.

0: Disable, 1: Enable, 2: Auto(Default) 0:Disable, 1:Enable, 2: Auto

Definition at line 188 of file FspmUpd.h.

11.36.2.22 UUINT8 FSPM_CONFIG::KtiLinkSpeed

Offset 0x0082 - KTI Link Speed KTI Link Speed.

0: 128GT, 1: 144GT, 2: 160GT, 3: Max KTI Link Speed(Default), 4: Frequency Per Link 0:128GT, 1:144GT, 2:160GT, 3:Max KTI Link Speed, 4:Frequency Per Link

Definition at line 176 of file FspmUpd.h.

11.36.2.23 UUINT8 FSPM_CONFIG::KtiLinkSpeedMode

Offset 0x0081 - KTI Link Speed Mode KTI Link Speed Mode.

0: Slow, 1: Full(Default) 0:Slow, 1:Full

Definition at line 169 of file FspmUpd.h.

11.36.2.24 UUINT8 FSPM_CONFIG::LegacyVgaSoc

Offset 0x006B - Legacy VGA Socket Socket that claims the legacy VGA range.

Default: Socket 0

Definition at line 77 of file FspmUpd.h.

11.36.2.25 UINT8 FSPM_CONFIG::LegacyVgaStack

Offset 0x006C - Legacy VGA Stack Stack that claims the legacy VGA range.

Default: Stack 0

Definition at line 82 of file FspmUpd.h.

11.36.2.26 UINT8 FSPM_CONFIG::MbeBwCal

Offset 0x008E - MBA BW Calibration MBA BW Calibration setting.

0: Linear, 1: Biased, 2: Legacy, **3: Auto(Default)** 0:Linear, 1:Biased, 2:Legacy, 3:Auto

Definition at line 250 of file FspmUpd.h.

11.36.2.27 UINT8 FSPM_CONFIG::mmCfgBase

Offset 0x0093 - MMCFG Base Address MMCFG Base Address.

0: 1GB, 1: 1.5GB, 2: 1.75GB, 3: 2GB, 4: 2.25GB, 5: 3GB, **6: Auto(Default)** 0:1GB, 1:1.5GB, 2:1.75GB, 3:2GB, 4:2.25GB, 5:3GB, 6:Auto

Definition at line 281 of file FspmUpd.h.

11.36.2.28 UINT8 FSPM_CONFIG::mmCfgSize

Offset 0x0094 - MMCFG Size Select MMCFG Size.

0: 64MB, 1: 128MB, 2: 256MB, 3: 512MB, 4: 1GB, 5: 2GB, **6: Auto(Default)** 0:64MB, 1:128MB, 2:256MB, 3:512MB, 4:1GB, 5:2GB, 6: Auto

Definition at line 288 of file FspmUpd.h.

11.36.2.29 UINT32 FSPM_CONFIG::mmiohBase

Offset 0x0098 - MMIO High Base Address MMIO High Base Address, a hex number for Bit[51:32].

Default: 0x6 (Gives 0x200)

Definition at line 297 of file FspmUpd.h.

11.36.2.30 UINT16 FSPM_CONFIG::mmiohSize

Offset 0x009E - MMIO High Size MMIO High Size, Number of 1GB contiguous regions to be assigned for MMIOH space per CPU.

Range 1-1024, Default: 3

Definition at line 315 of file FspmUpd.h.

11.36.2.31 UINT8 FSPM_CONFIG::NtbBarSizeEmBarSZ1

Offset 0x0103 - IIO NtbBarSizeEmBarSZ1 IIO NtbBarSizeEmBarSZ1.

. Default: 0x16

Definition at line 533 of file FspmUpd.h.

11.36.2.32 UINT8 FSPM_CONFIG::NtbBarSizeEmBarSZ2

Offset 0x0104 - IIO NtbBarSizeEmBarSZ2 IIO NtbBarSizeEmBarSZ2.

. Default: 0x16

Definition at line 538 of file FspmUpd.h.

11.36.2.33 UINT8 FSPM_CONFIG::NtbBarSizeEmBarSZ2_0

Offset 0x0105 - IIO NtbBarSizeEmBarSZ2_0 IIO NtbBarSizeEmBarSZ2_0.

. Default: 0x0C

Definition at line 543 of file FspmUpd.h.

11.36.2.34 UINT8 FSPM_CONFIG::NtbBarSizeEmBarSZ2_1

Offset 0x0106 - IIO NtbBarSizeEmBarSZ2_1 IIO NtbBarSizeEmBarSZ2_1.

. Default: 0x0C

Definition at line 548 of file FspmUpd.h.

11.36.2.35 UINT8 FSPM_CONFIG::NtbBarSizeImBar1

Offset 0x00FF - IIO NtbBarSizeImBar1 IIO NtbBarSizeImBar1.

Default: 0x16

Definition at line 513 of file FspmUpd.h.

11.36.2.36 UINT8 FSPM_CONFIG::NtbBarSizeImBar2

Offset 0x0100 - IIO NtbBarSizeImBar2 IIO PNtbBarSizeImBar2.

Default: 0x16

Definition at line 518 of file FspmUpd.h.

11.36.2.37 UINT8 FSPM_CONFIG::NtbBarSizeImBar2_0

Offset 0x0101 - IIO NtbBarSizeImBar2_0 IIO PNtbBarSizeImBar2_0.

Default: 0x0C

Definition at line 523 of file FspmUpd.h.

11.36.2.38 UINT8 FSPM_CONFIG::NtbBarSizeImBar2_1

Offset 0x0102 - IIO NtbBarSizeImBar2_1 IIO NtbBarSizeImBar2_1.

Default: 0x0C

Definition at line 528 of file FspmUpd.h.

11.36.2.39 UINT8 FSPM_CONFIG::NtbPpd

Offset 0x00FC - IIO Non-Transparent Port Definition IIO Non-Transparent Port Definition.

0: Transparent(Default), 1: Non-Transparent Bridge, 2: Non-Transparent Root Port 0:Transparent, 1:Non-Transparent Bridge, 2:Non-Transparent Root Port

Definition at line 498 of file FspmUpd.h.

11.36.2.40 UINT8 FSPM_CONFIG::NtbXlinkCtlOverride

Offset 0x0107 - IIO Non-Transparent Cross Link Override IIO Non-Transparent Cross Link Override.

1:Operate as RP, 2:Operate as NTB-NTB (NT Port), **3:Operate as NTB-> DSP (NTB EP)(Default)** 1:Operate as RP, 2:Operate as NTB-NTB (NT Port), 3:Operate as NTB-> DSP (NTB EP)

Definition at line 555 of file FspmUpd.h.

11.36.2.41 UINT8 FSPM_CONFIG::PchPciePIISsc

Offset 0x010F - PCH PCIE PLL Ssc Valid spread range : 0x00-0x14 (A value of 0 is SSC of 0.0%.

A value of 20 is SSC of 2.0%), Auto : 0xFE(Set to hardware default), **Disable(Default) : 0xFF**

Definition at line 603 of file FspmUpd.h.

11.36.2.42 UINT8 FSPM_CONFIG::PchSirqMode

Offset 0x010B - PchSirqMode PchSirqMode.

0: Quiet Mode(Default) 1: Continuous Mode 0:Quiet Mode, 1:Continuous Mode

Definition at line 577 of file FspmUpd.h.

11.36.2.43 UINT8 FSPM_CONFIG::PcieCommonClock

Offset 0x00FB - IIO PCIe Common Clock IIO PCIe Common Clock.

0: Disable, **1: Enable(Default)**, 2: Auto 0:Disable, 1:Enable, 2:Auto

Definition at line 491 of file FspmUpd.h.

11.36.2.44 UINT8 FSPM_CONFIG::SerialloUartDebugEnabled

Offset 0x014E - SerialloUartDebugEnabled **Enable(Default)** or Disable Seriallo Uart debug library in FSP.

0:Disable, 1:Enable

Definition at line 638 of file FspmUpd.h.

11.36.2.45 UINT16 FSPM_CONFIG::SerialloUartDebugloBase

Offset 0x0150 - ISA Serial Base selection Select ISA Serial Base address could be initialized by boot loader.

Default is 0x3F8 0x3F8:0x3F8, 0x2F8:0x2F8

Definition at line 648 of file FspmUpd.h.

11.36.2.46 UINT8 FSPM_CONFIG::SnoopAllCores

Offset 0x006A - Snoop Throttle Config Set the Snoop All Core Config.

0: Disable(Default), 1: Enable, 2: Auto 0:Disable, 1:Enable, 2:Auto

Definition at line 72 of file FspmUpd.h.

11.36.2.47 UINT8 FSPM_CONFIG::SnoopThrottleConfig

Offset 0x0069 - Snoop Throttle Config Set the Snoop Throttle Config.

0: Disable(Default), 1: Min, 2: Med, 3: Max 0:Disable, 1:Min, 2:Med, 3:Max

Definition at line 66 of file FspmUpd.h.

11.36.2.48 UINT8 FSPM_CONFIG::SplitLock

Offset 0x0092 - Split Lock Split Lock.

0: Disable(Default), 1: Enable, 2: Auto 0:Disable, 1:Enable, 2:Auto

Definition at line 274 of file FspmUpd.h.

11.36.2.49 UINT8 FSPM_CONFIG::StaleAtoSOptEn

Offset 0x0090 - HA A to S directory optimization HA A to S directory optimization.

0: Disable, 1: Enable, **2: Auto(Default)** 0:Disable, 1:Enable, 2:Auto

Definition at line 262 of file FspmUpd.h.

11.36.2.50 UINT8 FSPM_CONFIG::ThermalDeviceEnable

Offset 0x010D - ThermalDeviceEnable Thermal Device Mode.

0: Disable, 1: Enabled in PCI mode, **2: Enabled in ACPI mode(Default)** 0: Disable, 1: Enabled in PCI mode, 2: Enabled in ACPI mode

Definition at line 590 of file FspmUpd.h.

11.36.2.51 UINT8 FSPM_CONFIG::TorThresLoctoremEmpty

Offset 0x008D - TOR threshold - Loctorem threshold empty TOR threshold - Loctorem threshold empty.

0: Disable, **1: Auto(Default)**, 2: Low, 3: Medium, 4: High 0:Disable, 1:Auto, 2:Low, 3:Medium, 4:High

Definition at line 244 of file FspmUpd.h.

11.36.2.52 UINT8 FSPM_CONFIG::TorThresLoctoremNorm

Offset 0x008C - TOR threshold - Loctorem threshold normal TOR threshold - Loctorem threshold normal.

0: Disable, **1: Auto(Default)**, 2: Low, 3: Medium, 4: High 0:Disable, 1:Auto, 2:Low, 3:Medium, 4:High

Definition at line 237 of file FspmUpd.h.

11.36.2.53 UINT8 FSPM_CONFIG::TscSyncEn

Offset 0x008F - TSC Sync in Sockets TSC Sync in Sockets.

0: Disable, 1: Enable, **2: Auto(Default)** 0:Disable, 1:Enable, 2:Auto

Definition at line 256 of file FspmUpd.h.

11.36.2.54 UINT8 FSPM_CONFIG::UmaClustering

Offset 0x0070 - UMA Clustering Set number of enabled UMA Clusters.

0: Disable(Default), 2: Two Clusters, 4: Four Clusters 0:Disable, 2:Two Clusters, 4:Four Clusters

Definition at line 109 of file FspmUpd.h.

11.36.2.55 UINT32 FSPM_CONFIG::WaitTimeForPSBP

Offset 0x00B8 - WaitTimeForPSBP Number of milliseconds to wait for remote CPUs to initialize.

Default: 30 sec

Definition at line 361 of file FspmUpd.h.

11.36.2.56 UINT8 FSPM_CONFIG::XptPrefetchEn

Offset 0x0075 - XPT Prefetch Enable XPT Prefetch.

0: Disable, 1: Enable, **2: Auto(Default)** 0:Disable, 1:Enable, 2:Auto

Definition at line 139 of file FspmUpd.h.

The documentation for this struct was generated from the following file:

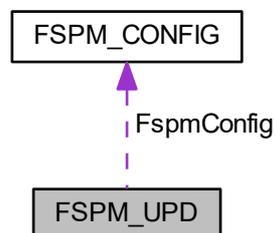
- [FspmUpd.h](#)

11.37 FSPM_UPD Struct Reference

Fsp M UPD Configuration.

```
#include <FspmUpd.h>
```

Collaboration diagram for FSPM_UPD:



Public Attributes

- FSP_UPD_HEADER [FspUpdHeader](#)

- *Offset 0x0000.*
- [FSPM_ARCH_UPD FspmArchUpd](#)
- *Offset 0x0020.*
- [FSPM_CONFIG FspmConfig](#)
- *Offset 0x0040.*
- [UINT8 UnusedUpdSpace4 \[2\]](#)
- *Offset 0x0164.*
- [UINT16 UpdTerminator](#)
- *Offset 0x0166.*

11.37.1 Detailed Description

Fsp M UPD Configuration.

Definition at line 661 of file FspmUpd.h.

The documentation for this struct was generated from the following file:

- [FspmUpd.h](#)

11.38 FSPS_CONFIG Struct Reference

FSP-S Configuration.

```
#include <FspsUpd.h>
```

Public Attributes

- [UINT8 BifurcationPcie0](#)
Offset 0x0020 - PCIe Controller 0 Bifurcation Configure PCI Express controller 0 bifurcation.
- [UINT8 BifurcationPcie1](#)
Offset 0x0021 - PCIe Controller 1 Bifurcation Configure PCI Express controller 1 bifurcation.
- [UINT8 ActiveCoreCount](#)
Offset 0x0022 - Active Core Count Select # of Active Cores (Default: 0, 0:ALL, 1..15 = 1..15 Cores) 0:ALL, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10:10, 11:11, 12:12, 13:13, 14:14, 15:15.
- [UINT8 UnusedUpdSpace0](#)
Offset 0x0023.
- [UINT32 CpuMicrocodePatchBase](#)
Offset 0x0024.
- [UINT32 CpuMicrocodePatchSize](#)
Offset 0x0028.
- [UINT8 EnablePcie0](#)
Offset 0x002C - PCIe Controller 0 Enable / Disable PCI Express controller 0 \$EN_DIS.
- [UINT8 EnablePcie1](#)
Offset 0x002D - PCIe Controller 1 Enable / Disable PCI Express controller 1 \$EN_DIS.
- [UINT8 EnableEmmc](#)
Offset 0x002E - Embedded Multi-Media Controller (eMMC) Enable / Disable Embedded Multi-Media controller \$EN_DIS.
- [UINT8 EnableGbE](#)
Offset 0x002F - LAN Controllers Enable / Disable LAN controllers, refer to FSP Integration Guide for details.
- [UINT32 FiaMuxConfigRequestPtr](#)
Offset 0x0030.

- UINT8 [PcieRootPort0DeEmphasis](#)
Offset 0x0034 - PCIe Root Port 0 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort1DeEmphasis](#)
Offset 0x0035 - PCIe Root Port 1 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort2DeEmphasis](#)
Offset 0x0036 - PCIe Root Port 2 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort3DeEmphasis](#)
Offset 0x0037 - PCIe Root Port 3 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort4DeEmphasis](#)
Offset 0x0038 - PCIe Root Port 4 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort5DeEmphasis](#)
Offset 0x0039 - PCIe Root Port 5 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort6DeEmphasis](#)
Offset 0x003A - PCIe Root Port 6 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT8 [PcieRootPort7DeEmphasis](#)
Offset 0x003B - PCIe Root Port 7 DeEmphasis Desired DeEmphasis level for PCIe root port 0:6dB, 1:3.5dB.
- UINT32 [EMMCDLLConfigPtr](#)
Offset 0x003C - eMMC DLL Configuration Data Pointer to eMMC DLL Configuration Data.
- UINT8 [ReservedSiliconInitUpd](#) [16]
Offset 0x0040.

11.38.1 Detailed Description

FSP-S Configuration.

Definition at line 43 of file FspsUpd.h.

11.38.2 Member Data Documentation

11.38.2.1 UINT8 FSPS_CONFIG::BifurcationPcie0

Offset 0x0020 - PCIe Controller 0 Bifurcation Configure PCI Express controller 0 bifurcation.

0:X2X2X2X2, 1:X2X2X4, 2:X4X2X2, 3:X4X4, 4:X8

Definition at line 49 of file FspsUpd.h.

11.38.2.2 UINT8 FSPS_CONFIG::BifurcationPcie1

Offset 0x0021 - PCIe Controller 1 Bifurcation Configure PCI Express controller 1 bifurcation.

0:X2X2X2X2, 1:X2X2X4, 2:X4X2X2, 3:X4X4, 4:X8

Definition at line 55 of file FspsUpd.h.

11.38.2.3 UINT8 FSPS_CONFIG::EnableGbE

Offset 0x002F - LAN Controllers Enable / Disable LAN controllers, refer to FSP Integration Guide for details.

0:Disable LAN 0 & LAN 1, 1:Enable LAN 0 & LAN 1, 2:Disable LAN 1 only

Definition at line 98 of file FspsUpd.h.

The documentation for this struct was generated from the following file:

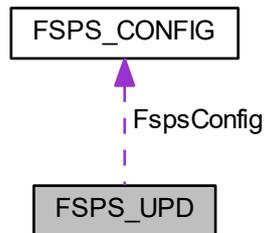
- [FspsUpd.h](#)

11.39 FSPS_UPD Struct Reference

Fsp S UPD Configuration.

```
#include <FspsUpd.h>
```

Collaboration diagram for FSPS_UPD:



Public Attributes

- FSP_UPD_HEADER [FspUpdHeader](#)
Offset 0x0000.
- FSPS_CONFIG [FspConfig](#)
Offset 0x0020.
- UINT8 [UnusedUpdSpace1](#) [6]
Offset 0x0050.
- UINT16 [UpdTerminator](#)
Offset 0x0056.

11.39.1 Detailed Description

Fsp S UPD Configuration.

Definition at line 164 of file `FspsUpd.h`.

The documentation for this struct was generated from the following file:

- [FspsUpd.h](#)

11.40 FSPT_CONFIG Struct Reference

FSP-T Configuration.

```
#include <FsptUpd.h>
```

Public Attributes

- UINT8 [FsptPort80RouteDisable](#)
-

Offset 0x0040 - Disable Port80 output in FSP-T Select Port80 Control in FSP-T (0:VPD-Style, 1:Enable Port80 Output, 2:Disable Port80 Output, refer to FSP Integration Guide for details 0:VPD-Style, 1:Enable Port80 Output[Default], 2:Disable Port80 Output.

- UINT8 [ReservedTempRamInitUpd](#) [31]

Offset 0x0041.

11.40.1 Detailed Description

FSP-T Configuration.

Definition at line 68 of file FsptUpd.h.

The documentation for this struct was generated from the following file:

- [FsptUpd.h](#)

11.41 FSPT_CORE_UPD Struct Reference

FSP-T Core UPD.

```
#include <FsptUpd.h>
```

Public Attributes

- UINT32 [MicrocodeRegionBase](#)
Offset 0x0020.
- UINT32 [MicrocodeRegionLength](#)
Offset 0x0024.
- UINT32 [CodeRegionBase](#)
Offset 0x0028.
- UINT32 [CodeRegionLength](#)
Offset 0x002C.
- UINT8 [Reserved1](#) [16]
Offset 0x0030.

11.41.1 Detailed Description

FSP-T Core UPD.

Definition at line 43 of file FsptUpd.h.

The documentation for this struct was generated from the following file:

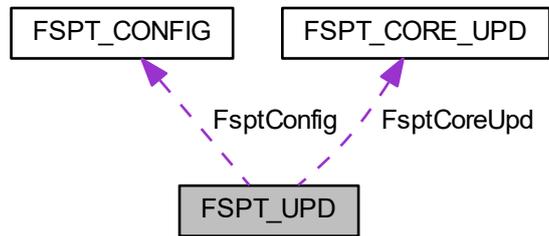
- [FsptUpd.h](#)

11.42 FSPT_UPD Struct Reference

Fsp T UPD Configuration.

```
#include <FsptUpd.h>
```

Collaboration diagram for FSPT_UPD:



Public Attributes

- FSP_UPD_HEADER [FspUpdHeader](#)
Offset 0x0000.
- FSPT_CORE_UPD [FsptCoreUpd](#)
Offset 0x0020.
- FSPT_CONFIG [FsptConfig](#)
Offset 0x0040.
- UINT8 [UnusedUpdSpace0](#) [6]
Offset 0x0060.
- UINT16 [UpdTerminator](#)
Offset 0x0066.

11.42.1 Detailed Description

Fsp T UPD Configuration.

Definition at line 84 of file [FsptUpd.h](#).

The documentation for this struct was generated from the following file:

- [FsptUpd.h](#)

11.43 GUID Struct Reference

128 bit buffer containing a unique identifier value.

```
#include <Base.h>
```

11.43.1 Detailed Description

128 bit buffer containing a unique identifier value.

Unless otherwise specified, aligned on a 64 bit boundary.

Definition at line 222 of file [Base.h](#).

The documentation for this struct was generated from the following file:

- [Base.h](#)

11.44 IPv4_ADDRESS Struct Reference

4-byte buffer.

```
#include <Base.h>
```

11.44.1 Detailed Description

4-byte buffer.

An IPv4 internet protocol address.

Definition at line 232 of file Base.h.

The documentation for this struct was generated from the following file:

- [Base.h](#)

11.45 IPv6_ADDRESS Struct Reference

16-byte buffer.

```
#include <Base.h>
```

11.45.1 Detailed Description

16-byte buffer.

An IPv6 internet protocol address.

Definition at line 239 of file Base.h.

The documentation for this struct was generated from the following file:

- [Base.h](#)

11.46 KTI_HOST_IN Struct Reference

KTIRC input structure.

```
#include <KtiHost.h>
```

Public Attributes

- **UINT8** [BusRatio](#) [MAX_SOCKET]
Indicates the ratio of Bus/MMIOL/IO resource to be allocated for each CPU's IIO.
 - **UINT8** [D2KCreditConfig](#)
1 - Min, 2 - Med (Default), 3- Max
 - **UINT8** [SnoopThrottleConfig](#)
0 - Disabled (Default), 1 - Min, 2 - Med, 3- Max
 - **UINT8** [SnoopAllCores](#)
0 - Disabled, 1 - Enabled, 2 - Auto
-

- UINT8 [LegacyVgaSoc](#)
Socket that claims the legacy VGA range; valid values are 0-7; 0 is default.
 - UINT8 [LegacyVgaStack](#)
Stack that claims the legacy VGA range; valid values are 0-3; 0 is default.
 - UINT8 [ColdResetRequestStart](#)
 - UINT8 [P2pRelaxedOrdering](#)
0 - Disable(default) 1 - Enable
 - UINT8 [DebugPrintLevel](#)
Bit 0 - Fatal, Bit1 - Warning, Bit2 - Info Summary; Bit 3 - Info detailed. 1 - Enable; 0 - Disable.
 - UINT8 [SncEn](#)
0 - Disable, (default) 1 - Enable
 - UINT8 [UmaClustering](#)
0 - Disable, 2 - 2Clusters UMA, 4 - 4Clusters UMA
 - UINT8 [IoDcMode](#)
0 - Disable IODC, 1 - AUTO (default), 2 - IODC_EN_REM_INVITOM_PUSH, 3 - IODC_EN_REM_INVITOM_ALL↔OCFLOW 4 - IODC_EN_REM_INVITOM_ALLOC_NONALLOC, 5 - IODC_EN_REM_INVITOM_AND_WCILF
 - UINT8 [DegradePrecedence](#)
Use DEGRADE_PRECEDENCE definition; TOPOLOGY_PRECEDENCE is default.
 - UINT8 [Degrade4SPreference](#)
4S1LFullConnect topology is default; another option is 4S2LRing topology.
 - UINT8 [DirectoryModeEn](#)
0 - Disable; 1 - Enable (default)
 - UINT8 [XptPrefetchEn](#)
Xpt Prefetch : 1 - Enable; 0 - Disable; 2 - Auto (default)
 - UINT8 [KtiPrefetchEn](#)
Kti Prefetch : 1 - Enable; 0 - Disable; 2 - Auto (default)
 - UINT8 [XptRemotePrefetchEn](#)
Xpt Remote Prefetch : 1 - Enable; 0 - Disable; 2 - Auto (default) (ICX only)
 - UINT8 [RdCurForXptPrefetchEn](#)
RdCur for XPT Prefetch : 0 - Disable, 1 - Enable, 2- Auto (default)
 - UINT8 [KtiFpgaEnable](#) [MAX_SOCKET]
Indicate if should enable Fpga device found in this socket : 0 - Disable, 1 - Enable, 2- Auto.
 - UINT8 [DdrtQosMode](#)
DDRT QoS Feature: 0 - Disable (default), 1 - M2M QoS Enable, Cha QoS Disable 2 - M2M QoS Enable, Cha QoS Enable.
 - UINT8 [KtiLinkSpeedMode](#)
Link speed mode selection; 0 - Slow Speed; 1- Full Speed (default)
 - UINT8 [KtiLinkSpeed](#)
Use KTI_LINKSPEED definition.
 - UINT8 [KtiAdaptationEn](#)
0 - Disable, 1 - Enable
 - UINT8 [KtiAdaptationSpeed](#)
Use KTI_LINK_SPEED definition; MAX_KTI_LINK_SPEED - Auto (i.e BIOS chosen speed)
 - UINT8 [KtiLinkL0pEn](#)
0 - Disable, 1 - Enable, 2- Auto (default)
 - UINT8 [KtiLinkL1En](#)
0 - Disable, 1 - Enable, 2- Auto (default)
 - UINT8 [KtiFailoverEn](#)
0 - Disable, 1 - Enable, 2- Auto (default)
 - UINT8 [KtiLbEn](#)
0 - Disable(default), 1 - Enable
-

- UINT8 [KtiCrcMode](#)
CRC_MODE_16BIT, CRC_MODE_ROLLING_32BIT, CRC_MODE_AUTO or CRC_MODE_PER_LINK.
 - UINT8 [KtiCpuSktHotPlugEn](#)
0 - Disable (default), 1 - Enable
 - UINT8 [KtiCpuSktHotPlugTopology](#)
0 - 4S Topology (default), 1 - 8S Topology
 - UINT8 [KtiSkuMismatchCheck](#)
0 - No, 1 - Yes (default)
 - UINT8 [IrqThreshold](#)
IRQ Threshold setting.
 - UINT8 [TorThresLoctoremNorm](#)
TOR threshold - Loctorem threshold normal.
 - UINT8 [TorThresLoctoremEmpty](#)
TOR threshold - Loctorem threshold empty.
 - UINT8 [MbeBwCal](#)
0 - Linear, 1 - Biased, 2 - Legacy, 3 - AUTO (default = Linear)
 - UINT8 [TscSyncEn](#)
TSC sync in sockets: 0 - Disable, 1 - Enable, 2 - AUTO (Default)
 - UINT8 [StaleAtoSOptEn](#)
HA A to S directory optimization: 1 - Enable; 0 - Disable; 2 - Auto (Default)
 - UINT8 [LLCDeadLineAlloc](#)
LLC dead line alloc: 1 - Enable(Default); 0 - Disable.
 - UINT8 [SplitLock](#)
 - UINT8 [ColdResetRequestEnd](#)
 - KTI_CPU_SETTING [PhyLinkPerPortSetting](#) [MAX_SOCKET]
Phy/Link Layer Options (per Port)
 - UINT8 [mmCfgBase](#)
MMCFG Base address, must be 64MB (SKX, HSX, BDX) / 256MB (GROVEPORT) aligned. Options: {0:1G, 1:1.5G, 2:1.75G, 3:2G, 4:2.25G, 5:3G, 6: Auto}.
 - UINT8 [mmCfgSize](#)
MMCFG Size address, must be 64M, 128M or 256M. Options: {0:64M, 1:128M, 2:256M, 3:512M, 4:1G, 5:2G, 6: Auto}.
 - UINT32 [mmiolBase](#)
MMIOL Base address, must be 64MB aligned.
 - UINT32 [mmiolSize](#)
MMIOL Size address.
 - UINT32 [mmiohBase](#)
Address bits above 4GB, i.e, the hex value here is address Bit[45:32] for SKX family, Bit[51:32] for ICX-SP.
 - UINT8 [CpuPaLimit](#)
Limits the max address to 46bits. This will take precedence over mmiohBase.
 - UINT8 [lowGap](#)
 - UINT8 [highGap](#)
 - UINT16 [mmiohSize](#)
Number of 1GB contiguous regions to be assigned for MMIOH space per CPU. Range 1-1024.
 - UINT8 [isocEn](#)
1 - Enable; 0 - Disable (BIOS will force this for 4S)
 - UINT8 [dcaEn](#)
1 - Enable; 0 - Disable
 - UINT32 [BoardTypeBitmask](#)
BoardTypeBitmask:
 - UINT32 [AllLanesPtr](#)
-

- Pointer to an array of ALL_LANES_EPARAM_LINK_INFO structures.*

 - UUINT32 [PerLanePtr](#)
- Pointer to an array of PER_LANE_EPARAM_LINK_INFO structures.*

 - UUINT32 [AllLanesSizeOfTable](#)

Number of elements in array pointed to by AllLanesPtr.

 - UUINT32 [PerLaneSizeOfTable](#)

Number of elements in array pointed to by PerLanePtr.
- UUINT32 [WaitTimeForPSBP](#)
- the wait time in units of 1000us for PBSP to check in.*
- BOOLEAN [IsKtiNvramDataReady](#)
- Used internally, Reserved.*
- UUINT32 [OemHookPostTopologyDiscovery](#)
- OEM_HOOK_POST_TOPOLOGY_DISCOVERY function pointer. Invoked at the end of topology discovery, used for error reporting.*
- UUINT32 [OemGetResourceMapUpdate](#)
- OEM_GET_RESOURCE_MAP_UPDATE function pointer. Allows platform code to adjust the resource map.*
- UUINT32 [OemGetAdaptedEqSettings](#)
- UUINT32 [OemCheckCpuPartsChangeSwap](#)
- BOOLEAN [WaSerializationEn](#)
- Enable BIOS serialization WA by PcdWaSerializationEn.*
- UUINT8 [KtiInEnableMktme](#)
- 0 - Disabled; 1 - Enabled; Mktme status decides D2Kti feature state*
- UUINT32 [CFRImagePtr](#)
- Pointers to the location of the CFR/SINIT binaries.*
- UUINT8 [S3mCFRCommit](#)
- 0 - Disable S3m CFR flow. 1 - Provision S3m CFR but not Commit. 2 - Provision and Commit S3M CFR.*
- UUINT8 [PucodeCFRCommit](#)
- 0 - Disable Pucode CFR flow. 1 - Provision Pucode CFR but not Commit. 2 - Provision and Commit Pucode CFR.*

11.46.1 Detailed Description

KTIRC input structure.

Definition at line 182 of file KtiHost.h.

11.46.2 Member Data Documentation

11.46.2.1 UUINT32 KTI_HOST_IN::BoardTypeBitmask

BoardTypeBitmask:

- Bits[3:0] - Socket0
- Bits[7:4] - Socket1
- Bits[11:8] - Socket2
- Bits[15:12] - Socket3
- Bits[19:16] - Socket4
- Bits[23:20] - Socket5
- Bits[27:24] - Socket6

- Bits[31:28] - Socket7

Within each Socket-specific field, bits mean:

- Bit0 = CPU_TYPE_STD support; always 1 on Socket0
- Bit1 = CPU_TYPE_F support
- Bit2 = CPU_TYPE_P support
- Bit3 = reserved

Definition at line 282 of file KtiHost.h.

11.46.2.2 UINT8 KTI_HOST_IN::BusRatio[MAX_SOCKET]

Indicates the ratio of Bus/MMIOL/IO resource to be allocated for each CPU's IIO.

Value 0 indicates, that CPU is not relevant for the system. If resource is requested for an CPU that is not currently populated, KTIRC will assume that the ratio is 0 for that CPU and won't allocate any resources for it. If resource is not requested for an CPU that is populated, KTIRC will force the ratio for that CPU to 1.

Definition at line 196 of file KtiHost.h.

11.46.2.3 UINT32 KTI_HOST_IN::CFRImagePtr

Pointers to the location of the CFR/SINIT binaries.

Contains a pointer to a 24 byte fixed length array. The array contains the 3 instances of the following c-struct

```
typedef struct {
    UINT32  CfrImagePtr;
    UINT32  CfrImageSize;
}
```

This allows a maximum of 3 CFR/SINIT binaries to be provided by platform code.

Definition at line 311 of file KtiHost.h.

11.46.2.4 UINT8 KTI_HOST_IN::ColdResetRequestEnd

Deprecated Reserved.

Definition at line 245 of file KtiHost.h.

11.46.2.5 UINT8 KTI_HOST_IN::ColdResetRequestStart

Deprecated Reserved.

Definition at line 203 of file KtiHost.h.

11.46.2.6 UINT8 KTI_HOST_IN::highGap

Deprecated Reserved.

Definition at line 260 of file KtiHost.h.

11.46.2.7 UINT8 KTI_HOST_IN::lowGap

Deprecated Reserved.

Definition at line 259 of file KtiHost.h.

11.46.2.8 UINT32 KTI_HOST_IN::OemCheckCpuPartsChangeSwap

Deprecated Reserved, must be set to 0.

Definition at line 292 of file KtiHost.h.

11.46.2.9 UINT32 KTI_HOST_IN::OemGetAdaptedEqSettings

Deprecated Reserved, must be set to 0.

Definition at line 291 of file KtiHost.h.

11.46.2.10 UINT8 KTI_HOST_IN::SplitLock

Deprecated Reserved, must be set to 0.

Definition at line 244 of file KtiHost.h.

The documentation for this struct was generated from the following file:

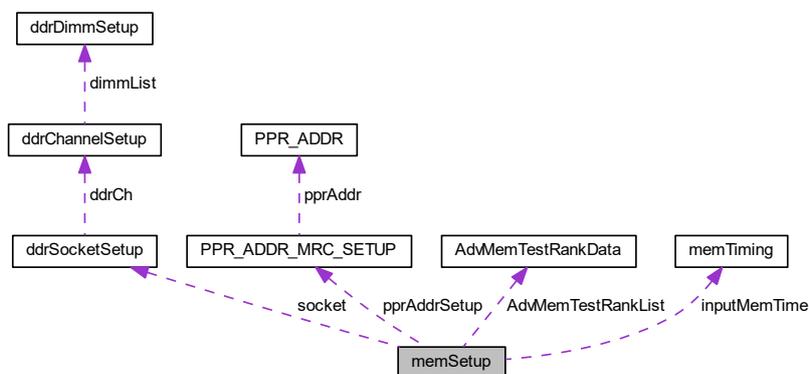
- [KtiHost.h](#)

11.47 memSetup Struct Reference

Host memory setup structure declaration.

```
#include <MemoryPolicyPpi.h>
```

Collaboration diagram for memSetup:



Public Attributes

- UINT32 [options](#)
Flags for enabling (1)/disabling(0) MRC features.
 - UINT32 [optionsExt](#)
Flags for enabling (1)/disabling(0) MRC features.
 - UINT32 [optionsNgn](#)
NGN Flags.
 - UINT8 [PdaModeX16](#)
PDA behavior for x16 devices.
 - UINT8 [imcBclk](#)
IMC BCLK frequency.
 - UINT8 [enforcePOR](#)
Enforce memory POR configurations.
 - UINT8 [ddrFreqLimit](#)
DDR Frequency Limit.
 - UINT8 [chInter](#)
Channels interleave setting.
 - UINT8 [dimmTypeSupport](#)
DIMM types.
 - UINT8 [ckeThrottling](#)
CKE Power managment mode.
 - UINT8 [olttPeakBWLIMITPercent](#)
Open Loop Thermal Throttling.
 - UINT16 [thermalThrottlingOptions](#)
Bitmapped field for Thermal Throttling Modes.
 - UINT8 [TempRefreshOption](#)
Option to manually enter Temperature refresh value.
 - UINT8 [HalfxRefreshValue](#)
Half X temperature refresh value.
 - UINT8 [TwoxRefreshValue](#)
Two X temperature refresh value.
 - UINT8 [FourxRefreshValue](#)
Four X temperature refresh value.
 - UINT8 [MemHotOuputAssertThreshold](#)
Thermal Throttling O/P bits - (High | Mid | Low).
 - UINT8 [ThrottlingMidOnTempLo](#)
Enable/Disable the initialization of THRTMID on TEMPLO.
 - UINT8 [DramRaplEnable](#)
Enable/Disable DRAM RAPL.
 - UINT8 [dramraplbwlimittf](#)
Multiplier of BW_LIMIT_TF when DRAM RAPL is enabled.
 - UINT8 [CmsEnableDramPm](#)
Notify PCU to enable/disable DRAM PM of memory controller.
 - UINT8 [dramraplRefreshBase](#)
DRAM RAPL Refresh Base.
 - UINT8 [perBitDeSkew](#)
Enable disable per Bit DeSkew Training.
 - UINT8 [FactoryResetClear](#)
NVDIMM Factory Reset Clear.
 - UINT8 [enableBacksideRMT](#)
-

- Enable Backside RMT.*

 - UINT8 [enableBacksideCMDRMT](#)
Enable Backside CMD RMT.
 - UINT8 [enableNgnBcomMargining](#)
Enable NVMDIMM BCOM margining support.
 - UINT8 [trainingResultOffsetFunctionEnable](#)
Training Result Offset function enable or disable.
 - INT16 [offsetTxDq](#)
Platform value to offset the final memory training result of TxDq.
 - INT16 [offsetRxDq](#)
Platform value to offset the final memory training result of RxDq.
 - INT16 [offsetTxVref](#)
Platform value to offset the final memory training result of TxVref.
 - INT16 [offsetRxVref](#)
Platform value to offset the final memory training result of RxVref.
 - INT16 [offsetCmdAll](#)
Platform value to offset the final memory training result of CmdAll.
 - INT16 [offsetCmdVref](#)
Platform value to offset the final memory training result of CmdVref.
 - INT16 [offsetCtlAll](#)
Platform value to offset the final memory training result of CtlAll.
 - INT16 [OffsetRecEn](#)
Platform value to offset the final memory training result of RecvEn.
 - UINT32 [rmtPatternLength](#)
Rank Margin Test: patten length.
 - UINT32 [rmtPatternLengthExt](#)
Rank Margin Test: patten length extension.
 - UINT32 [patrolScrubDuration](#)
Memory RAS: Specifies the number of hours it takes for patrol scrub to scrub all system memory.
 - UINT8 [DieSparing](#)
Enable/Disable Memory RAS die sparing.
 - UINT8 [NgnAddressRangeScrub](#)
Memory RAS: Address Range Scrubbing.
 - UINT16 [memTestLoops](#)
Number of MemTests loops to execute for legacy MemTest (type 8 and 10), that provides the ability of inverting the data pattern in every odd pass for detecting opposite polarity faults.
 - UINT32 [AdvMemTestOptions](#)
CPGC MemTest step bit fields to enable different advanced MemTest options.
 - UINT8 [AdvMemTestPpr](#)
Enable/Disable PPR repair during Advanced Memtest.
 - UINT8 [AdvMemTestRetry](#)
Retry the Advanced Memtest step after a PPR repair occurs This option is useful for testing that the PPR repair was successful, but it adds some latency.
 - UINT8 [AdvMemTestResetList](#)
Reset row fail list after executing each Advanced MemTest option This option is useful for testing multiple options.
 - UINT8 [AdvMemTestCondition](#)
Set Test Conditions for Advanced Memtest algorithms ADV_MEM_TEST_COND_DISABLE - Do not modify test conditions during Advanced Memtest ADV_MEM_TEST_COND_AUTO - Modify test conditions automatically based on Advanced Memtest algorithm ADV_MEM_TEST_COND_MANUAL - Modify test conditions manually based on Adv↔ MemTestCond input options.
 - UINT16 [AdvMemTestCondVdd](#)
-

Manually set Vdd level when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify Vdd in units of mV.

- UINT8 [AdvMemTestCondTwr](#)
Manually set host Write Recovery time when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify host tWR value in units of tCK.
 - UINT16 [AdvMemTestCondTrefi](#)
Manually set host tREFI time when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify host tR←EFI in units of usec.
 - UINT32 [AdvMemTestCondPause](#)
Manually set Pause time without refresh when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify the Pause time in units of msec.
 - UINT8 [AdvMemTestRankListNumEntries](#)
Indicate the number of Ranks that will be tested in the system.
 - [AdvMemTestRankData AdvMemTestRankList](#) [ADV_MT_LIST_LIMIT]
The list of Rank addresses in the system that will execute AdvMemTest.
 - UINT16 [scrambleSeedLow](#)
Low 16 bits of the data scrambling seed.
 - UINT16 [scrambleSeedHigh](#)
High 16 bits of the data scrambling seed.
 - UINT8 [ADREn](#)
ADR: Enable/Disable Async DRAM Refresh(ADR) feature
 - UINT8 [LegacyADRMdEn](#)
ADR: Enable/Dsiable Legacy ADR Async DRAM Refresh(ADR) feature
 - UINT8 [MinNormalMemSize](#)
ADR: Minimum memory size assigned as system memory when only JEDEC NVDIMMs are present
 - UINT8 [ADRDataSaveMode](#)
ADR: Data Save Mode for ADR.
 - UINT8 [check_pm_sts](#)
ADR: Use the PCH_PM_STS register as ADR recovery indicator.
 - UINT8 [check_platform_detect](#)
ADR: Use the PlatformDetectADR OEM hook function as ADR recovery indicator.
 - UINT16 [normOpplntvl](#)
Memory RAS: Normal operation duration within sparing interval.
 - SMB_CLOCK_FREQUENCY [SpdSmbSpeed](#)
SM Bus Clock Frequency- see SMB_CLOCK_FREQUENCY.
 - UINT8 [SpdPrintEn](#)
Enable(1)/Disable(0) SPD data Print.
 - UINT16 [SpdPrintLength](#)
Pirnt length of SPD data.
 - struct [ddrSocketSetup socket](#) [MAX_SOCKET]
Socket setup configuration.
 - struct [memTiming inputMemTime](#)
Memory timing settings.
 - UINT8 [customRefreshRate](#)
Custom tuning multiplier of Refresh rate from 2.0x to 4.0x in units of 0.1x.
 - UINT8 [partialmirrorsad0](#)
Enable Mirror on entire memory for TAD0.
 - UINT16 [partialmirrorsize](#) [MAX_PARTIAL_MIRROR]
Size of each partial mirror to be created in order.
-

- UINT8 [partialMirrorUEFI](#)
Imitate behavior of UEFI based Address Range Mirror with setup option.
 - UINT32 [partialmirrorpercent](#)
Numerator of the mirror ratio.
 - UINT8 [partialmirrorsts](#)
Partial mirror status.
 - UINT8 [ImmediateFailoverAction](#)
Immediate failover enable or disable when mirror scrub reads a uncorrected error.
 - UINT8 [dIIResetTestLoops](#)
Number of times to loop through RMT to test the DLL Reset.
 - UINT32 [memFlows](#)
Flags to enable(1)/disable(0) memory training steps in MRC flow.
 - UINT32 [memFlowsExt](#)
Extension of flags to enable(1)/disable(0) memory training steps in MRC flow.
 - UINT8 [writePreamble](#)
Write Preamble timing.
 - UINT8 [readPreamble](#)
Read Preamble timing.
 - UINT8 [DramRapIExtendedRange](#)
Enable extended range for DRAM RAPL.
 - UINT16 [spareErrTh](#)
Memory RAS: Threshold value for logging Correctable Errors(CE).
 - UINT8 [NsddcEn](#)
Memory RAS: Enable/Disable New 48B SDDC.

 - UINT8 [EsddcEn](#)
Memory RAS: Enable/Disable enhanced sddc.

 - UINT8 [ColumnCorrectionDisable](#)
Disable - Turns ON Column Correction feature. Enable - Turns OFF Column Correction feature.
 - UINT8 [leakyBktTimeWindow](#)
Memory RAS: Enable/Disable leaky bucket time window based interface.

 - UINT8 [leakyBktLo](#)
Leaky bucket low mask position.
 - UINT8 [leakyBktHi](#)
Leaky bucket high mask position.
 - UINT16 [leakyBktHour](#)
Leaky bucket time window based interface Hour(0 - 3744).
 - UINT8 [leakyBktMinute](#)
Leaky bucket time window based interface Minute" (0 - 60).
 - UINT8 [spareRanks](#)
Number of spare ranks per channel.
 - UINT8 [interNVDIMMS](#)
Controls if NVDIMMs are interleaved together or not.
 - UINT8 [restoreNVDIMMS](#)
Control if BIOS will perform NVDIMM Restore operation.
 - UINT8 [eraseArmNVDIMMS](#)
Control if BIOS will perform NVDIMM erase & ARM operations.
 - UINT8 [cmdSetupPercentOffset](#)
Cmd setup percent offset for late cmd traning result.
-

- UINT8 [pprType](#)
Memory RAS.
 - [PPR_ADDR_MRC_SETUP](#) [pprAddrSetup](#) [MAX_PPR_ADDR_ENTRIES]
PPR Address.
 - UINT8 [imcInter](#)
IMC interleave setting (within a socket). Valid options are 1 or 2 way interleave.
 - UINT8 [oneRankTimingModeEn](#)
Enable/Disable support for JEDEC RCD v2.0+ One Rank Timing Mode.
 - UINT8 [volMemMode](#)
Volatile Memory Mode.
 - UINT8 [CacheMemType](#)
For 2LM, the caching type.
 - UINT8 [DdrCacheSize](#)
Size of channel DDR to use as 2LM cache.
 - UINT8 [PmemCaching](#)
Caching control for AppDirect.
 - UINT8 [EadrSupport](#)
eADR support.
 - UINT8 [FadrSupport](#)
Enable or disable fADR support.
 - UINT8 [memInterleaveGran1LM](#)
Memory interleave mode for 1LM.
 - UINT8 [EnableTwoWayNmCache](#)
Enable or disable biased 2-way near memory cache.
 - UINT16 [NonPreferredWayMask](#)
A 10-bit mask to control the bias counter ratio.
 - UINT8 [PreferredReadFirst](#)
Reads are issued to the non-preferred or preferred way first.
 - UINT8 [FastZeroMemSupport](#)
Enable or disable boot-time fast zero memory support.
 - UINT8 [DdrtMemPwrSave](#)
Enable/Disable DDRT memory power saving.
 - UINT8 [patrolScrubAddrMode](#)
Memory RAS: Patrol Scrub Address Mode.
 - UINT8 [SrefProgramming](#)
Self Refresh control programming.
 - UINT8 [OppSrefEn](#)
Opportunistic self-refresh setting.
 - UINT8 [MdllOffEn](#)
Master DLLs (MDLL) setting.
 - UINT8 [PkgcSrefEn](#)
Enables or disables Self Refresh in PkgC flow.
 - UINT8 [CkMode](#)
Configures CK behavior during self-refresh.
 - UINT8 [CkeProgramming](#)
CKE Registers Programming Mode.
 - UINT8 [CkIdleTimer](#)
CKE Idle Timer.
 - UINT8 [ApdEn](#)
CKE Active Power Down Mode for DDR4 DIMMs.
 - UINT8 [PpdEn](#)
-

- CKE Precharge Power Down (PPD).*
- UINT8 [DdrtCkeEn](#)
 - CKE Active Power Down Mode for DDR-T DIMMs.*
- UINT8 [DataDIIOff](#)
 - Turn off DDRIO data DLL in CKE Power Down or OppSR low power mode.*
- UINT8 [ExtendedADDDCEn](#)
 - RAS: Enable/Disable Extended ADDDC sparing.*
- UINT8 [Blockgnt2cmd1cyc](#)
 - DDRT Defeature Enable/Disable BLOCK GNT2CMD1CYC.*
- UINT8 [Disddrtopprd](#)
 - Enable/Disable NVMDIMM OPPRD.*
- UINT8 [setSecureEraseAllDIMMs](#)
 - NGNVM DIMM Secure Erase Unit, Erases the persistent memory region of the selected DIMMs".*
- UINT8 [setSecureEraseSktCh](#) [MAX_SOCKET][MAX_CH]
 - Enable/Disable secure erase of persistent memory region of NVMDIMM.*
- UINT8 [FastGoConfig](#)
 - Select Crystal Ridge FastGo QoS Configuration Profiles.*
- UINT8 [NvmdimmPerfConfig](#)
 - Non-Volatile Memory DIMM baseline performance settings depending on the workload behavior.*
- EFI_MEMORY_TOPOLOGY_TYPE [MemoryTopology](#) [MAX_SOCKET][MAX_CH]
 - Memory topology of each channel per socket.*
- EFI_MEMORY_DIMM_CONNECTOR_TYPE [MemoryConnectorType](#) [MAX_SOCKET][MAX_CH]
 - Memory connector type of each channel per socket.*
- UINT8 [AppDirectMemoryHole](#)
 - Enable/Disable the App Direct Memory Hole.*
- UINT8 [LatchSystemShutdownState](#)
 - Enable/disable Latch System Shutdown (LSS) of all enabled NVDIMMs.*
- UINT8 [EliminateDirectoryInFarMemory](#)
 - Select snoopy mode for 2LM.*
- UINT8 [NvmdimmPowerCyclePolicy](#)
 - Power Cycle Policy on NVM Surprise Clock Stop.*
- UINT8 [NvDimmEnergyPolicy](#)
 - NV DIMM Energy Policy Management.*
- UINT8 [RxDfeEn](#)
 - Option to force Rx DFE enabled or disabled.*
- UINT8 [TxRiseFallSlewRate](#)
 - Enable/Disable TX Rise Fall Slew Rate Training.*
- UINT8 [ForcePxclnit](#)
 - Forces PXC (Phase-based Crosstalk Cancellation) initialization.*
- UINT8 [CmiInitOption](#)
 - CMI Initialize Option.*
- UINT8 [DisableDirForAppDirect](#)
 - Snoopy mode for AD.*
- UINT8 [NvmMediaStatusException](#)
 - Enable/Disable Crystal Ridge MediaStatus Exception.*
- UINT8 [NvmQos](#)
 - Select Crystal Ridge QoS tuning recipes.*
- UINT8 [ExtendedType17](#)
 - Disable/Enable using extended Type 17 SMBIOS Structures.*
- UINT16 [DcpmmAveragePowerLimit](#)
 - Gen 2 Intel Optane DC Persistent Memory (DCPMM) Average Power Limit (in mW)".*

- UINT8 [DcpmmAveragePowerTimeConstant](#)
Gen 2 DCPMM Average Power Time Constant for Turbo Mode support (in mSec).
- UINT32 [DcpmmMbbAveragePowerTimeConstant](#)
Gen 2 DCPMM Average Power Time Constant for Memory Bandwidth Boost Feature support(in mSec).
- UINT8 [DcpmmMbbFeature](#)
Gen 2 DCPMM Turbo Mode/Memory Bandwidth Boost Feature Enable.
- UINT16 [DcpmmMbbMaxPowerLimit](#)
DCPPM Power limit in mW for Turbo Mode/Memory Bandwidth Boost Feature.
- UINT8 [LsxImplementation](#)
Select LSx (LSI/LSR/LSW) ACPI method implementation.
- UINT32 [NvdimmSmbusMaxAccessTime](#)
Set Smbus maximum access time
- UINT32 [NvdimmSmbusReleaseDelay](#)
Set Smbus release delay.
- UINT8 [NfitPublishMailboxStructsDisable](#)
Controls Mailbox structures in the NFIT.
- UINT8 [EnforcePopulationPor](#)
Enforce memory population POR configurations.
- UINT8 [TrefiPerChannel](#)
Configure Stagger Host Refresh feature.
- UINT8 [TrainingCompOptions](#)
Training Comp Options Values.
- UINT8 [PeriodicRcomp](#)
Periodic Rcomp Control.
- UINT8 [PeriodicRcompInterval](#)
Periodic Rcomp Interval.
- BOOLEAN [UseSmbusForMrwEarly](#)
Use SMBUS for early MRW commands.
- UINT8 [AepNotSupportedException](#)
Enable/Disable AEP DIMM Not Supported Exception.
- UINT8 [PanicWm](#)
Select between Panic/High Watermark of Auto or High or Low.
- UINT8 [DataBufferDfe](#)
Enable/Disable LRDIMM DB DFE.
- UINT8 [VirtualNumaEnable](#)
Enable/Disable Virtual NUMA.
- UINT32 [smartTestKey](#)
Smart Test Key pattern.
- BOOLEAN [RmtMinimumMarginCheckEnable](#)
Enable RMT minimum margin check.

11.47.1 Detailed Description

Host memory setup structure declaration.

Definition at line 422 of file MemoryPolicyPpi.h.

11.47.2 Member Data Documentation

11.47.2.1 UINT8 memSetup::ADRSaveMode

ADR: Data Save Mode for ADR.

0=Disabled,
1=Batterybacked,
2=NVDIMM.

Definition at line 952 of file MemoryPolicyPpi.h.

11.47.2.2 UINT8 memSetup::ADREn

ADR: Enable/Disable Async DRAM Refresh(ADR) feature

0 - Disable.
1 - Enable.

Definition at line 922 of file MemoryPolicyPpi.h.

11.47.2.3 UINT32 memSetup::AdvMemTestCondPause

Manually set Pause time without refresh when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify the Pause time in units of msec.

It is applied between write and read steps to test data retention.

Definition at line 886 of file MemoryPolicyPpi.h.

11.47.2.4 UINT16 memSetup::AdvMemTestCondTrefi

Manually set host tREFI time when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify host tREFI in units of usec.

7800 = 1x refresh rate; 15600 = 0.5x refresh rate

Definition at line 879 of file MemoryPolicyPpi.h.

11.47.2.5 UINT8 memSetup::AdvMemTestCondTwr

Manually set host Write Recovery time when AdvMemTestCondition = ADV_MEM_TEST_COND_MANUAL Specify host tWR value in units of tCK.

This timing is only applicable in Open Page mode.

Definition at line 873 of file MemoryPolicyPpi.h.

11.47.2.6 UINT8 memSetup::AdvMemTestRankListNumEntries

Indicate the number of Ranks that will be tested in the system.

A value of 0 will test all Ranks

Definition at line 891 of file MemoryPolicyPpi.h.

11.47.2.7 UINT8 memSetup::AdvMemTestResetList

Reset row fail list after executing each Advanced MemTest option This option is useful for testing multiple options.

Definition at line 853 of file MemoryPolicyPpi.h.

11.47.2.8 UINT8 memSetup::AepNotSupportedException

Enable/Disable AEP DIMM Not Supported Exception.

0 = Disable.

1 = Enable.

Definition at line 1894 of file MemoryPolicyPpi.h.

11.47.2.9 UINT8 memSetup::ApdEn

CKE Active Power Down Mode for DDR4 DIMMs.

0 = APD is disabled.

1 = APD is enabled.

Definition at line 1475 of file MemoryPolicyPpi.h.

11.47.2.10 UINT8 memSetup::AppDirectMemoryHole

Enable/Disable the App Direct Memory Hole.

0 = disable.

1 = enable.

Definition at line 1606 of file MemoryPolicyPpi.h.

11.47.2.11 UINT8 memSetup::Blockgnt2cmd1cyc

DDRT Defeature Enable/Disable BLOCK GNT2CMD1CYC.

0 = Disabled.

1 = Enabled.

Definition at line 1520 of file MemoryPolicyPpi.h.

11.47.2.12 UINT8 memSetup::CacheMemType

For 2LM, the caching type.

Only valid if volMemMode is 2LM 0 - DDR caching DDRT.

Definition at line 1325 of file MemoryPolicyPpi.h.

11.47.2.13 UINT8 memSetup::check_platform_detect

ADR: Use the PlatformDetectADR OEM hook function as ADR recovery indicator.

0 - Disable.

1 - Enable.

Definition at line 970 of file MemoryPolicyPpi.h.

11.47.2.14 UINT8 memSetup::check_pm_sts

ADR: Use the PCH_PM_STS register as ADR recovery indicator.

0 - Disable.

1 - Enable.

Definition at line 961 of file MemoryPolicyPpi.h.

11.47.2.15 UINT8 memSetup::chInter

Channels interleave setting.

Valid options are 1, 2, or 3 way interleave. Other values defaults to 3 way interleave.

Definition at line 581 of file MemoryPolicyPpi.h.

11.47.2.16 UINT8 memSetup::CkeldleTimer

CKE Idle Timer.

Set the number of rank idle cycles that causes CKE power-down entrance. The number of idle cycles (in DCLKs) are based from command CS assertion. It is important to program this parameter to be greater than roundtrip latency parameter in order to avoid the CKE de-assertion sooner than data return.

Definition at line 1466 of file MemoryPolicyPpi.h.

11.47.2.17 UINT8 memSetup::CkeProgramming

CKE Registers Programming Mode.

Select manual or auto programming registers Control for CKE (DRAM powerdown modes). at Load Line point 0/1/2/3.

0 - auto - MRC determines the value.

1 - manual - use value from user Setup.

Definition at line 1455 of file MemoryPolicyPpi.h.

11.47.2.18 UINT8 memSetup::ckeThrottling

CKE Power management mode.

0 = Disabled.

1 = APD Enabled, PPD Disabled.

2 = APD Disabled, PPDF Enabled.

3 = APD Disabled, PPDS Enabled.

4 = APD Enabled, PPDF Enabled.

5 = APD Enabled, PPDS Enabled.

Definition at line 602 of file MemoryPolicyPpi.h.

11.47.2.19 UINT8 memSetup::CkMode

Configures CK behavior during self-refresh.

0 - CK is driven during self refresh.

2 - CK is pulled low during self refresh.

Definition at line 1444 of file MemoryPolicyPpi.h.

11.47.2.20 UINT8 memSetup::cmdSetupPercentOffset

Cmd setup percent offset for late cmd traning result.

The possible values are from 0 to 100.

Definition at line 1276 of file MemoryPolicyPpi.h.

11.47.2.21 UINT8 memSetup::CmilnitOption

CMI Initialize Option.

0 = Initialize with desired credit.

1 = Inialize with default(Reset Value)credit.

Definition at line 1687 of file MemoryPolicyPpi.h.

11.47.2.22 UINT8 memSetup::CmsEnableDramPm

Notify PCU to enable/disable DRAM PM of memory controller.

0 - Disable.

1 - Enable.

Definition at line 692 of file MemoryPolicyPpi.h.

11.47.2.23 `UINT8 memSetup::DataBufferDfe`

Enable/Disable LRDIMM DB DFE.

0 - Disable;
1 - Pmem Only;
2 - All LRDIMM;

Definition at line 1915 of file MemoryPolicyPpi.h.

11.47.2.24 `UINT8 memSetup::DataDIIOff`

Turn off DDRIO data DLL in CKE Power Down or OppSR low power mode.

0 = Do not turn off data DLL.
1 = Turn off data DLL.

Definition at line 1502 of file MemoryPolicyPpi.h.

11.47.2.25 `UINT16 memSetup::DcpmmAveragePowerLimit`

Gen 2 Intel Optane DC Persistent Memory (DCPMM) Average Power Limit (in mW)".
Valid range for power limit starts from 10000mW and must be a multiple of 250mW."

Definition at line 1738 of file MemoryPolicyPpi.h.

11.47.2.26 `UINT8 memSetup::DcpmmAveragePowerTimeConstant`

Gen 2 DCPMM Average Power Time Constant for Turbo Mode support (in mSec).

This value is used as a base time window for power usage measurements.

Definition at line 1746 of file MemoryPolicyPpi.h.

11.47.2.27 `UINT32 memSetup::DcpmmMbbAveragePowerTimeConstant`

Gen 2 DCPMM Average Power Time Constant for Memory Bandwidth Boost Feature support(in mSec).

This value is used as a base time window for power usage measurements.

Definition at line 1755 of file MemoryPolicyPpi.h.

11.47.2.28 `UINT8 memSetup::DcpmmMbbFeature`

Gen 2 DCPMM Turbo Mode/Memory Bandwidth Boost Feature Enable.

0 = Disable.
1 = Enable.

Definition at line 1764 of file MemoryPolicyPpi.h.

11.47.2.29 UUINT16 memSetup::DcpmmMbbMaxPowerLimit

DCPPM Power limit in mW for Turbo Mode/Memory Bandwidth Boost Feature.

DCPPM Power limit in mW used for limiting the Turbo Mode/Memory Bandwidth Boost power consumption (Valid range starts from 15000mW).

Definition at line 1773 of file MemoryPolicyPpi.h.

11.47.2.30 UUINT8 memSetup::DdrCacheSize

Size of channel DDR to use as 2LM cache.

Size of channel DDR to use as 2LM cache when Volatile Memory Mode under Crystal Ridge is 1LM+2LM.

Definition at line 1334 of file MemoryPolicyPpi.h.

11.47.2.31 UUINT8 memSetup::ddrFreqLimit

DDR Frequency Limit.

Forces a DDR frequency slower than the common tCK detected via SPD.

A DDR frequency faster than the common frequency is a config error.

Options are 0=AUTO, 1=DDR_800, 3=DDR_1066, 5=DDR_1333, 7=DDR_1600, 9=DDR_1866, 11=DDR_2133, 13=DDR2400.

Definition at line 572 of file MemoryPolicyPpi.h.

11.47.2.32 UUINT8 memSetup::DdrTckeEn

CKE Active Power Down Mode for DDR-T DIMMs.

0 = APD is disabled.

1 = APD is enabled.

Definition at line 1493 of file MemoryPolicyPpi.h.

11.47.2.33 UUINT8 memSetup::dimmTypeSupport

DIMM types.

0=RDIMM, 1=UDIMM, 2 = RDIMMandUDIMM or SODIMM, 9=LRDIMM; 10=QRDIMM, 11=NVMDIMM.

Definition at line 589 of file MemoryPolicyPpi.h.

11.47.2.34 UUINT8 memSetup::DisableDirForAppDirect

Snoopy mode for AD.

Snoopy mode for AD: Disable/Enable new AD specific feature to avoid directory updates to DDRT memory from non-NUMA optimized workloads.

0 = Disable.

1 = Enable.

Definition at line 1698 of file MemoryPolicyPpi.h.

11.47.2.35 UINT8 memSetup::Disddrtopprd

Enable/Disable NVMDIMM OPRD.

0 = DDRT RPQ Reads will not be scheduled in DDR4 mode DDRT Underfill Reads will not be scheduled in DDR4 mode.

1 = DDRT RPQ Reads will be scheduled in DDR4 mode. GNTs continue to be blocked in DDR4 mode. This should be set for DDRT 2N mode. DDRT Underfill Reads will be scheduled in DDR4 mode. GNTs continue to be blocked in DDR4 mode This bit should be set for DDRT 2N mod.

Definition at line 1533 of file MemoryPolicyPpi.h.

11.47.2.36 UINT8 memSetup::DramRaplEnable

Enable/Disable DRAM RAPL.

0 - disable.

1 - enable.

Definition at line 678 of file MemoryPolicyPpi.h.

11.47.2.37 UINT8 memSetup::dramraplRefreshBase

DRAM RAPL Refresh Base.

Allows custom tuning of Power scaling by Refresh rate in units of 0.1x when DRAM RAPL is enabled.

Definition at line 701 of file MemoryPolicyPpi.h.

11.47.2.38 UINT8 memSetup::EliminateDirectoryInFarMemory

Select snoopy mode for 2LM.

Set to 0 to Enables new 2LM specific feature to avoid directory updates to far-memory from non-NUMA optimized workloads.

0 = Enable eliminating directory in far memory.

1 = Disable eliminating directory in far memory.

Definition at line 1630 of file MemoryPolicyPpi.h.

11.47.2.39 UIN8 memSetup::EnforcePopulationPor

Enforce memory population POR configurations.

0 (ENFORCE_POPULATION_POR_DIS) - Do not enforce memory population POR.

1 (ENFORCE_POPULATION_POR_ENFORCE_SUPPORTED) - Enforce supported memory populations.

2 (ENFORCE_POPULATION_POR_ENFORCE_VALIDATED) - Enforce validated memory populations.

Definition at line 1817 of file MemoryPolicyPpi.h.

11.47.2.40 UIN8 memSetup::enforcePOR

Enforce memory POR configurations.

0 (ENFORCE_POR_EN) - Enforce memory POR.

1 (ENFORCE_STRETCH_EN) - Enforce memory frequency stretch goal.

2 (ENFORCE_POR_DIS) - Do not enforce POR configurations.

Definition at line 561 of file MemoryPolicyPpi.h.

11.47.2.41 UIN8 memSetup::ExtendedADDDCEn

RAS: Enable/Disable Extended ADDDC sparing.

0 = Disabled.

1 = Enabled.

Definition at line 1511 of file MemoryPolicyPpi.h.

11.47.2.42 UIN8 memSetup::ExtendedType17

Disable/Enable using extended Type 17 SMBIOS Structures.

0 = Disable.

1 = Enable .

Definition at line 1730 of file MemoryPolicyPpi.h.

11.47.2.43 UIN8 memSetup::FastGoConfig

Select Crystal Ridge FastGo QoS Configuration Profiles.

CR_FASTGO_DEFAULT 0;

CR_FASTGO_DISABLE 1;

```
CR_FASTGO_DISABLE_MLC_SQ_THRESHOLD_5 2;  
CR_FASTGO_DISABLE_MLC_SQ_THRESHOLD_6 3;  
CR_FASTGO_DISABLE_MLC_SQ_THRESHOLD_8 4;  
CR_FASTGO_DISABLE_MLC_SQ_THRESHOLD_10 5;  
CR_FASTGO_AUTOMATIC 6;  
CR_FASTGO_LAST_OPTION CR_FASTGO_AUTOMATIC;  
CR_FASTGO_KNOB_DEFAULT CR_FASTGO_AUTOMATIC.
```

Definition at line 1568 of file MemoryPolicyPpi.h.

11.47.2.44 UINT8 memSetup::ForcePxcInit

Forces PXC (Phase-based Crosstalk Cancellation) initialization.

Forces PXC (Phase-based Crosstalk Cancellation) initialization even if PXC training is not enabled.

0 = Disable.

1 = Enable.

Definition at line 1678 of file MemoryPolicyPpi.h.

11.47.2.45 UINT8 memSetup::imcBclk

IMC BCLK frequency.

0 - Auto, MRC code determine the value.

1 - 100MHz.

2 - 133MHz.

Definition at line 551 of file MemoryPolicyPpi.h.

11.47.2.46 UINT8 memSetup::LatchSystemShutdownState

Enable/disable Latch System Shutdown (LSS) of all enabled NVDIMMs.

LSS is supposed to be done by the persistent memory driver in OS using ACPI DSM function, before any write to persistent memory is done. BIOS knob is implemented to enable Latch LSS for operating systems that would not call DSM. Enabling latch twice is not a problem so the BIOS action does not colide with OSES that use DSM to enable latch.

0 = Disable.

1 = Enable.

Definition at line 1619 of file MemoryPolicyPpi.h.

11.47.2.47 UINT8 memSetup::LegacyADRMdEn

ADR: Enable/Dsiable Legacy ADR Async DRAM Refresh(ADR) feature

0 - Disable.

1 - Enable.

Definition at line 931 of file MemoryPolicyPpi.h.

11.47.2.48 UINT8 memSetup::LsxImplementation

Select LSx (LSI/LSR/LSW) ACPI method implementation.

0 = Software SMI.

1 = ASL.

Definition at line 1782 of file MemoryPolicyPpi.h.

11.47.2.49 UINT8 memSetup::MdllOffEn

Master DLLs (MDLL) setting.

Memory power management feature:

Master DLLs (MDLL) setting in Self Refresh controls at Load Line point 0/1/2/3 registers.

When 0 - Master DLLs (MDLL) cannot be turned off in Self Refresh.

When 1 - Master DLLs (MDLL) can be turned off in Self Refresh.

Definition at line 1425 of file MemoryPolicyPpi.h.

11.47.2.50 UINT32 memSetup::memFlows

Flags to enable(1)/disable(0) memory training steps in MRC flow.

The following are bit to MRC training step map.

```
MF_X_OVER_EN BIT0;
MF_SENSE_AMP_EN BIT1;
MF_E_CMDCLK_EN BIT2;
MF_REC_EN BIT3;
MF_RD_DQS_EN BIT4;
MF_WR_LVL_EN BIT5;
MF_WR_FLYBY_EN BIT6;
MF_WR_DQ_EN BIT7;
MF_CMDCLK_EN BIT8;
MF_RD_ADV_EN BIT9;
MF_WR_ADV_EN BIT10;
MF_RD_VREF_EN BIT11;
MF_WR_VREF_EN BIT12;
MF_RT_OPT_EN BIT13;
MF_RX_DESKEW_EN BIT14;
MF_TX_DESKEW_EN BIT14;
MF_TX_EQ_EN BIT15;
MF_IMODE_EN BIT16;
MF_EARLY_RID_EN BIT17;
MF_DQ_SWIZ_EN BIT18;
MF_LRBUF_RD_EN BIT19;
MF_LRBUF_WR_EN BIT20;
MF_RANK_MARGIN_EN BIT21;
MF_E_WR_VREF_EN BIT22;
MF_E_RD_VREF_EN BIT23;
MF_L_RD_VREF_EN BIT24;
```

```
MF_MEMINIT_EN BIT25;
MF_NORMAL_MODE_EN BIT27;
MF_CMD_VREF_EN BIT28;
MF_L_WR_VREF_EN BIT29;
MF_MEMTEST_EN BIT30;
MF_E_CTLCLK_EN BIT31.
```

Definition at line 1130 of file MemoryPolicyPpi.h.

11.47.2.51 UINT32 memSetup::memFlowsExt

Extension of flags to enable(1)/disable(0) memory training steps in MRC flow.

```
MF_EXT_RX_CTLE_EN BIT0
MF_EXT_PXC_EN BIT1
MF_EXT_CMD_NORM_EN BIT2
MF_EXT_LRDIMM_BKSIDE_EN BIT3
MF_EXT_CHECK_POR BIT6
MF_EXT_MMRC_RUN BIT7
MF_EXT_THROTTLING_EARLY BIT8
MF_EXT_THROTTLING BIT9
MF_EXT_POST_TRAINING BIT10
MF_EXT_E_CONFIG BIT11
MF_EXT_L_CONFIG BIT12
MF_EXT_MCODT_EN BIT14
MF_EXT_MCRON_EN BIT15
MF_EXT_DIMMRON_EN BIT16
MF_EXT_CACLK_BACKSIDE_EN BIT17
MF_DQ_SWIZ_X16_EN BIT18
MF_EXT_TCO_COMP_EN BIT19
MF_EXT_TX_SLEW_RATE_EN BIT20
MF_EXT_INIT_MEM_EN BIT21
MF_EXT_CMD_TX_EQ_EN BIT22
MF_EXT_RCOMP_STAT_LEG BIT23
MF_EXT_DDJC_EN BIT24
MF_EXT_RX_DFE_EN BIT25
MF_EXT_CSCLK_EN BIT26
MF_EXT_CSCLK_BACKSIDE_EN BIT27
MF_EXT_CACLK_EN BIT28
MF_X_OVER_HWFSM_EN BIT29
MF_EXT_INIT_CMI_EN BIT30
MF_EXT_QxCA_CLK_EN BIT31
```

Definition at line 1166 of file MemoryPolicyPpi.h.

11.47.2.52 UINT8 memSetup::MemHotOuputAssertThreshold

Thermal Throttling O/P bits - (High | Mid | Low).

0= Memhot output disabled,
 1 = Memhot on High,
 2 = Memhot on High|Mid,

3 = Memhot on High|Mid|Low.

Definition at line 660 of file MemoryPolicyPpi.h.

11.47.2.53 EFI_MEMORY_DIMM_CONNECTOR_TYPE memSetup::MemoryConnectorType[MAX_SOCKET][MAX_CH]

Memory connector type of each channel per socket.

0 = DimmConnectorPth.

1 = DimmConnectorSmt.

2 = DimmConnectorMemoryDown.

Definition at line 1597 of file MemoryPolicyPpi.h.

11.47.2.54 EFI_MEMORY_TOPOLOGY_TYPE memSetup::MemoryTopology[MAX_SOCKET][MAX_CH]

Memory topology of each channel per socket.

0 = DaisyChainTopology.

1 = InvSlotsDaisyChainTopology.

2 = TTopology.

Definition at line 1587 of file MemoryPolicyPpi.h.

11.47.2.55 UINT8 memSetup::MinNormalMemSize

ADR: Minimum memory size assigned as system memory when only JEDEC NVDIMMs are present

2 - 2GB.

4 - 4GB.

6 - 6GB.

8 - 8GB.

Definition at line 942 of file MemoryPolicyPpi.h.

11.47.2.56 UINT8 memSetup::NfitPublishMailboxStructsDisable

Controls Mailbox structures in the NFIT.

0 - Publish Mailbox structures in the NFIT 1 - Do not publish Mailbox structures in the NFIT

Definition at line 1807 of file MemoryPolicyPpi.h.

11.47.2.57 UINT16 memSetup::normOppIntvl

Memory RAS: Normal operation duration within sparing interval.

Definition at line 976 of file MemoryPolicyPpi.h.

11.47.2.58 UUINT8 memSetup::NvDimmEnergyPolicy

NV DIMM Energy Policy Management.

1 = Setting Energy Policy to Device Managed.

2 = Setting Energy Policy to Host Managed.

Definition at line 1647 of file MemoryPolicyPpi.h.

11.47.2.59 UUINT32 memSetup::NvdimmSmbusMaxAccessTime

Set Smbus maximum access time

Maximum amount of time (ms) UEFI mgmt driver is allowed to use the SMBus.

Definition at line 1790 of file MemoryPolicyPpi.h.

11.47.2.60 UUINT32 memSetup::NvdimmSmbusReleaseDelay

Set Smbus release delay.

Delay time (ms) before releasing after UEFI mgmt driver requests SMBus release.

Definition at line 1798 of file MemoryPolicyPpi.h.

11.47.2.61 UUINT8 memSetup::NvmdimmPerfConfig

Non-Volatile Memory DIMM baseline performance settings depending on the workload behavior.

0 = BW Optimized.

1 = Latency Optimized.

Definition at line 1577 of file MemoryPolicyPpi.h.

11.47.2.62 UUINT8 memSetup::NvmdimmPowerCyclePolicy

Power Cycle Policy on NVM Surprise Clock Stop.

Enable/Disable power cycle policy when NVMDIMM receive surprise clock stop.

Definition at line 1638 of file MemoryPolicyPpi.h.

11.47.2.63 UUINT8 memSetup::NvmMediaStatusException

Enable/Disable Crystal Ridge MediaStatus Exception.

0 = Disable.
1 = Enable.

Definition at line 1707 of file MemoryPolicyPpi.h.

11.47.2.64 UUINT8 memSetup::NvmQos

Select Crystal Ridge QoS tuning recipes.

0 = Enables tuning recipe 1 for CR QoS knobs
(recommended for 2-2-2 memory configuration in AD);
1 = Enables tuning recipe 2 for CR QoS knobs
(recommended for other memory configuration in AD);
2 = Enables tuning recipe 3 for CR QoS knobs
(recommended for 1 DIMM per channel config);
3 = Disable CR QoS feature.

Definition at line 1721 of file MemoryPolicyPpi.h.

11.47.2.65 UUINT8 memSetup::olttPeakBWLIMITPercent

Open Loop Thermal Throttling.

$(\text{value}/100) * 255 / \text{max number of dimms per channel} = \text{DIMM_TEMP_THRT_LMT THRT_HI}$.

Definition at line 610 of file MemoryPolicyPpi.h.

11.47.2.66 UUINT8 memSetup::OppSrefEn

Opportunistic self-refresh setting.

Memory power management feature:
opportunistic self-refresh setting in Self Refresh controls at Load Line point 0/1/2/3 registers.
0 - disable;
1 - enable.

Definition at line 1414 of file MemoryPolicyPpi.h.

11.47.2.67 UUINT32 memSetup::options

Flags for enabling (1)/disabling(0) MRC features.

TEMPHIGH_EN BIT0, enables support for 95 degree DIMMs.
ATTEMPT_FAST_BOOT_COLD BIT1.
PDWN_SR_CKE_MODE BIT2, enables CKE to be tri-stated during register clock off power down self-refresh.
OPP_SELF_REF_EN BIT3, enables the opportunistic self refresh mechanism.
MDLL_SHUT_DOWN_EN BIT4, enables MDLL shutdown.
PAGE_POLICY BIT5, Clear for open page, set for closed page. Open page has better performance and power usage in general. Close page may benefit some applications with poor locality.

ALLOW2XREF_EN BIT6, enables 2X refresh if needed for extended operating temperature range (95degrees) If TEMPHIGH_EN is also set, setting this bit will result in 2X refresh timing for the IMC refresh control register.

MULTI_THREAD_MRC_EN BIT7, enables multithreaded MRC. This reduces boot time for systems with multiple processor sockets.

ADAPTIVE_PAGE_EN BIT8, enables adaptive page mode. The memory controller will dynamically determine how long to keep pages open to improve performance.

CMD_CLK_TRAINING_EN BIT9, enables command to clock training step.

SCRAMBLE_EN BIT10, set to enable data scrambling. This should always be enabled except for debug purposes.

SCRAMBLE_EN_DDRT BIT11, set to enable data scrambling. This should always be enabled except for debug purposes.

DISPLAY_EYE_EN BIT12,

DDR_RESET_LOOP BIT13, enables infinite channel reset loop without retries for gathering of margin data.

NUMA_AWARE BIT14, enables configuring memory interleaving appropriately for NUMA aware OS.

DISABLE_WMM_OPP_READ BIT15, disables issuing read commands opportunistically during WMM.

RMT_COLD_FAST_BOOT BIT16.

ECC_CHECK_EN BIT17, enables ECC checking.

ECC_MIX_EN BIT18, enables ECC in a system with mixed ECC and non-ECC memory in a channel by disabling ECC when this configuration is detected.

DISABLE_ECC_SUPPORT BIT19, disables ECC check.

CA_PARITY_EN BIT20,

PER_NIBBLE_EYE_EN BIT22. RAS_TO_INDP_EN BIT23, switches from lockstep or mirror mode to independent channel mode when memory is present on channel 2 and this is enabled.

MARGIN_RANKS_EN BIT25, enables the rank margin tool.

MEM_OVERRIDE_EN BIT26, enables use of inputMemTime inputs as hard overrides.

DRAMDLL_OFF_PD_EN BIT27,

MEMORY_TEST_EN BIT28, enables execution of MemTest if on cold boot

MEMORY_TEST_COLD_FAST_BOOT_EN BIT29, enables the memory test when going through a cold fast boot path

ATTEMPT_FAST_BOOT BIT30, attempts to take a fast boot path if the NVRAM structure is good and the memory config hasn't changed. For example, on a warm boot, this will take the "fast warm" path through MRC which attempts to make it as close as possible to the S3 path.

SW_MEMORY_TEST_EN BIT31.

Definition at line 479 of file MemoryPolicyPpi.h.

11.47.2.68 UINT32 memSetup::optionsExt

Flags for enabling (1)/disabling(0) MRC features.

PD_CRC_CHECK BIT0
 SET_MEM_TESTED_EN BIT1
 AVAILABLE BIT2
 TURNAROUND_OPT_EN_DDRT BIT3
 PDA_EN BIT5
 TURNAROUND_OPT_EN BIT6
 AVAILABLE BIT7
 ALLOW_CORRECTABLE_ERROR BIT8
 ALLOW_CORRECTABLE_MEM_TEST_ERROR BIT9
 AVAILABLE BIT10
 AVAILABLE BIT11
 AVAILABLE BIT12
 PER_BIT_MARGINS BIT13
 DUTY_CYCLE_EN BIT14
 LRDIMM_BACKSIDE_VREF_EN BIT15

AVAILABLE BIT16
 DRAM_RX_EQ_EN BIT17
 AVAILABLE BIT18
 AVAILABLE BIT19
 AVAILABLE BIT20
 OPTIONS_EXT_RESERVED1 BIT21
 AVAILABLE BIT22
 WR_CRC BIT23
 OPTIONS_EXT_RESERVED2 BIT24
 AVAILABLE BIT25
 AVAILABLE BIT26
 AVAILABLE BIT27
 AVAILABLE BIT28
 DIMM_ISOLATION_EN BIT29
 AVAILABLE BIT30

Definition at line 516 of file MemoryPolicyPpi.h.

11.47.2.69 UUINT32 memSetup::optionsNgn

NGN Flags.

NGN_CMD_TIME BIT1
 NGN_DEBUG_LOCK BIT6
 NGN_ARS_ON_BOOT BIT7
 NGN_ARS_PUBLISH BIT9
 NGN_ECC_EXIT_CORR BIT10
 NGN_ECC_CORR BIT11
 NGN_ECC_WR_CHK BIT12
 NGN_ECC_RD_CHK BIT13

Definition at line 531 of file MemoryPolicyPpi.h.

11.47.2.70 UUINT8 memSetup::PanicWm

Select between Panic/High Watermark of Auto or High or Low.

0 = Auto 1 = High
 2 = Low

Definition at line 1907 of file MemoryPolicyPpi.h.

11.47.2.71 UUINT32 memSetup::partialmirrorpercent

Numerator of the mirror ratio.

Given the Numerator (N) and Denominator (D) returned by this function, and the total memory size (T), the mirror size (M) should be computed as follows:

$$M = (T * N) / D$$

MirroredAmountAbove4GB is the amount of available memory above 4GB that needs to be mirrored measured in basis point (hundredths of percent e.g. 12.75% = 1275). In a multi-socket system, platform is required to distribute

the mirrored memory ranges such that the amount mirrored is approximately proportional to the amount of memory on each NUMA node. E.g. on a two node machine with 64GB on node 0 and 32GB on node 1, a request for 12GB of mirrored memory should be allocated with 8GB of mirror on node 0 and 4GB on node 1.

Definition at line 1067 of file MemoryPolicyPpi.h.

11.47.2.72 UINT8 memSetup::partialmirrorsad0

Enable Mirror on entire memory for TAD0.

0 - Disable.

1 - Enable.

Definition at line 1039 of file MemoryPolicyPpi.h.

11.47.2.73 UINT8 memSetup::partialmirrorsts

Partial mirror status.

MIRROR_STATUS_SUCCESS 0
MIRROR_STATUS_MIRROR_INCAPABLE 1
MIRROR_STATUS_VERSION_MISMATCH 2
MIRROR_STATUS_INVALID_REQUEST 3
MIRROR_STATUS_UNSUPPORTED_CONFIG 4
MIRROR_STATUS_OEM_SPECIFIC_CONFIGURATION 5

Definition at line 1080 of file MemoryPolicyPpi.h.

11.47.2.74 UINT8 memSetup::partialMirrorUEFI

Imitate behavior of UEFI based Address Range Mirror with setup option.

It controls whether to enable partial mirror in 1LM and 2LM or not.

Definition at line 1052 of file MemoryPolicyPpi.h.

11.47.2.75 UINT8 memSetup::patrolScrubAddrMode

Memory RAS: Patrol Scrub Address Mode.

Selects the address mode between System Physical Address (or) Reverse Address.

0 - PATROL_SCRUB_REVERSE_ADDR,

1 - PATROL_SCRUB_SPA,

Definition at line 1390 of file MemoryPolicyPpi.h.

11.47.2.76 UINT8 memSetup::PdaModeX16

PDA behavior for x16 devices.

0 - will disable PDA operation when a x16 device is detected.

1 - will not modify PDA Mode.

Definition at line 541 of file MemoryPolicyPpi.h.

11.47.2.77 UINT8 memSetup::PeriodicRcomp

Periodic Rcomp Control.

Enable/Disable memory periodic Rcomp with PCU.

0 - Disable;

1 - Enable;

2 - Auto;

Definition at line 1869 of file MemoryPolicyPpi.h.

11.47.2.78 UINT8 memSetup::PeriodicRcompInterval

Periodic Rcomp Interval.

Interval of periodic Rcomp controlled by PCU.

Definition at line 1877 of file MemoryPolicyPpi.h.

11.47.2.79 UINT8 memSetup::PkgcSrefEn

Enables or disables Self Refresh in PkgC flow.

Memory power managment feature.

0 - Didable.

1 - Enable.

Definition at line 1435 of file MemoryPolicyPpi.h.

11.47.2.80 UINT8 memSetup::PpdEn

CKE Precharge Power Down (PPD).

0 = PPD is disabled.

1 = PPD is enabled.

Definition at line 1484 of file MemoryPolicyPpi.h.

11.47.2.81 PPR_ADDR_MRC_SETUP memSetup::pprAddrSetup[MAX_PPR_ADDR_ENTRIES]

PPR Address.

Buffer to hold DRAM Address that need to be repaired by PPR (Post Package Repair).

Platform Sample Implementation:

RAS code uses pprAddrSetup to cause MRC to launch PPR (Post Package Repair) on a subsequent boot. RAS code passes failed DRAM information into pprAddrSetup via the UEFI variable PPR_ADDR_VARIABLE.

Definition at line 1298 of file MemoryPolicyPpi.h.

11.47.2.82 UINT8 memSetup::pprType

Memory RAS.

Power-up DDR4 Post Package Repair (PPR) type.

- 0 - PPR disabled.
- 1 - PPR type hard.
- 2 - PPR type soft.

Definition at line 1286 of file MemoryPolicyPpi.h.

11.47.2.83 UINT8 memSetup::readPreamble

Read Preamble timing.

- 0 = PREAMBLE_1TCLK;
- 1 = PREAMBLE_2TCLK;
- 2 = PREAMBLE_3TCLK;
- 3 = PREAMBLE_4TCLK.

Definition at line 1195 of file MemoryPolicyPpi.h.

11.47.2.84 UINT8 memSetup::RxDfeEn

Option to force Rx DFE enabled or disabled.

- 0 = Disable Rx DFE.
- 1 = Enable Rx DFE.
- 2 = Auto. MRC code determines if enable or disable.

Definition at line 1657 of file MemoryPolicyPpi.h.

11.47.2.85 UINT8 memSetup::setSecureEraseAllDIMMs

NGNVM DIMM Secure Erase Unit, Erases the persistent memory region of the selected DIMMs".

0 - Erase DIMMs according to setting of setSecureEraseSktCh.

Definition at line 1543 of file MemoryPolicyPpi.h.

11.47.2.86 UINT8 memSetup::setSecureEraseSktCh[MAX_SOCKET][MAX_CH]

Enable/Disable secure erase of persistent memory region of NVMDIMM.

0 = Disable erasing the persistent memory region of NVMDIMM in <Channel 0, Memory controller 0, Socket 0.
1 = Enable erasing the persistent memory region of NVMDIMM in Channel 0, Memory controller 0, Socket 0.

Definition at line 1552 of file MemoryPolicyPpi.h.

11.47.2.87 UINT32 memSetup::smartTestKey

Smart Test Key pattern.

Option to enter the confidential key to be used

Definition at line 1932 of file MemoryPolicyPpi.h.

11.47.2.88 UINT16 memSetup::spareErrTh

Memory RAS: Threshold value for logging Correctable Errors(CE).

Threshold of 10 logs 10th CE, "All" logs every CE, and "None" means no CE logging. All and None are not valid with Rank Sparing.

Definition at line 1209 of file MemoryPolicyPpi.h.

11.47.2.89 UINT8 memSetup::SpdPrintEn

Enable(1)/Disable(0) SPD data Print.

0 - Disable.

1 - Enable.

Definition at line 996 of file MemoryPolicyPpi.h.

11.47.2.90 UINT16 memSetup::SpdPrintLength

Print length of SPD data.

0 - AUTO(512 for DDR4, 1024 for DDR5).
256.

512.

Definition at line 1006 of file MemoryPolicyPpi.h.

11.47.2.91 SMB_CLOCK_FREQUENCY memSetup::SpdSmbSpeed

SM Bus Clock Frequency- see SMB_CLOCK_FREQUENCY.

- 0 - SMB_CLK_100K.
- 1 - SMB_CLK_400K.
- 2 - SMB_CLK_700K.
- 3 - SMB_CLK_1M.

Definition at line 987 of file MemoryPolicyPpi.h.

11.47.2.92 UINT8 memSetup::SrefProgramming

Self Refresh control programming.

Memory power managment feature:

Select manual or auto programming Self Refresh controls at Load Line point 0/1/2/3 registers.

- 0 - auto - MRC determines the value;
- 1 - manual - use value from user Setup.

Definition at line 1402 of file MemoryPolicyPpi.h.

11.47.2.93 UINT8 memSetup::TempRefreshOption

Option to manually enter Temperature refresh value.

Select Manual to use value from HalfxRefreshValue, TwoxRefreshValue and FourxRefreshValue. Auto for default value in MRC code.

- 0 = Auto.
- 1 = Manual option select.

Definition at line 629 of file MemoryPolicyPpi.h.

11.47.2.94 UINT16 memSetup::thermalThrottlingOptions

Bitmapped field for Thermal Throttling Modes.

Defined in mem.thermalThrottlingOptions section.

Definition at line 618 of file MemoryPolicyPpi.h.

11.47.2.95 UINT8 memSetup::ThrottlingMidOnTempLo

Enable/Disable the initialization of THRTMID on TEMPLO.

0 = THRTMID on TEMPLO disabled,
1 = THRTMID on TEMPLO enabled.

Definition at line 669 of file MemoryPolicyPpi.h.

11.47.2.96 UINT8 memSetup::TrainingCompOptions

Training Comp Options Values.

Options for issuing a Comp. cycle (RCOMP) at specific points in training.

0 - One RCOMP cycle only on PHY Init (MMRC Init);
1 - One RCOMP cycle after every JEDEC Init;
2 - One RCOMP cycle right before every training step;

Definition at line 1858 of file MemoryPolicyPpi.h.

11.47.2.97 UINT8 memSetup::trainingResultOffsetFunctionEnable

Training Result Offset function enable or disable.

It controls whether to enable the function to offset the final training results or not.

Enable - Enables training results to be offset.
Disable - Disables this feature; current default is Enable disable

Definition at line 759 of file MemoryPolicyPpi.h.

11.47.2.98 UINT8 memSetup::TxRiseFallSlewRate

Enable/Disable TX Rise Fall Slew Rate Training.

0 = Dsiable.
1 = Enable.
2 = AUTO, will enable if DDR Freq >= 2933.

Definition at line 1667 of file MemoryPolicyPpi.h.

11.47.2.99 BOOLEAN memSetup::UseSmbusForMrwEarly

Use SMBUS for early MRW commands.

Option to require all MRW commands to be sent over SMBUS until QCA training is complete

Definition at line 1885 of file MemoryPolicyPpi.h.

11.47.2.100 UINT8 memSetup::VirtualNumaEnable

Enable/Disable Virtual NUMA.

0 - disable.
1 - enable.

Definition at line 1924 of file MemoryPolicyPpi.h.

11.47.2.101 UINT8 memSetup::volMemMode

Volatile Memory Mode.

0 - 1LM;
1 - 2LM;

Definition at line 1319 of file MemoryPolicyPpi.h.

11.47.2.102 UINT8 memSetup::writePreamble

Write Preamble timing.

0 = PREAMBLE_1TCLK;
1 = PREAMBLE_2TCLK;
2 = PREAMBLE_3TCLK;
3 = PREAMBLE_4TCLK.

Definition at line 1184 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.48 memTiming Struct Reference

Memory Timings Settings.

```
#include <MemoryPolicyPpi.h>
```

Public Attributes

- [UINT8 nCL](#)
Column Latency.
 - [UINT8 nRP](#)
Row Precharge.
 - [UINT8 nRCD](#)
RAS to CAS Delay.
 - [UINT8 nRRD](#)
Row to Row Delay.
 - [UINT8 nWTR](#)
Write to Read Delay.
 - [UINT8 nRAS](#)
Row Active Strobe.
 - [UINT8 nRTP](#)
-

- Read To Precharge delay.*
- UINT8 [nWR](#)
 - Write Recovery time.*
- UINT8 [nFAW](#)
 - Four Activate Window.*
- UINT8 [nCWL](#)
 - CAS (WRITE) latency (CWL).*
- UINT8 [nRC](#)
 - Row Cycle.*
- UINT8 [nCMDRate](#)
 - Command Rate.*
- UINT8 [ddrFreqLimit](#)
 - The limit of DDR frequency ratio, based on base clock frequency.*
- UINT16 [vdd](#)
 - Vdd for DRAM core.*
- UINT8 [ucVolt](#)
 - XMP Memory Controller Voltage Level.*
- UINT64 [casSup](#)
 - Bits map to indicate if a CAS in a CAS list is supported.*
- UINT16 [tREFI](#)
 - Refresh Interval.*
- UINT16 [nRFC](#)
 - Refresh to Activate Delay.*
- UINT16 [ddrFreq](#)
 - Frequency of DDR.*

11.48.1 Detailed Description

Memory Timings Settings.

Definition at line 31 of file MemoryPolicyPpi.h.

11.48.2 Member Data Documentation

11.48.2.1 UINT8 memTiming::nCL

Column Latency.

Column Latency (CL) time is the number of clock cycles needed to access a certain column of data in RAM. It's also known as CAS (column address strobe) time.

Definition at line 41 of file MemoryPolicyPpi.h.

11.48.2.2 UINT8 memTiming::nCMDRate

Command Rate.

Command Rate / Command per Clock (1T/2T) is the delay between a memory chip is selected and the first active command can be issued.

Definition at line 140 of file MemoryPolicyPpi.h.

11.48.2.3 UINT8 memTiming::nFAW

Four Activate Window.

Four Activate Window, which specifies the time window in which four activates are allowed on the same rank.

Definition at line 117 of file MemoryPolicyPpi.h.

11.48.2.4 UINT8 memTiming::nRAS

Row Active Strobe.

Row Active Strobe (RAS) time is the minimum number of clock cycles needed to access a certain row of data in RAM between the data request and the precharge command. It's known as active to precharge delay.

Definition at line 91 of file MemoryPolicyPpi.h.

11.48.2.5 UINT8 memTiming::nRC

Row Cycle.

Row Cycle time, the minimum time in cycles taken for a row to complete a full cycle, which typically can be calculated by $nRC = nRAS + nRP$.

Definition at line 131 of file MemoryPolicyPpi.h.

11.48.2.6 UINT8 memTiming::nRCD

RAS to CAS Delay.

RAS to CAS Delay (RCD) is the number of clock cycles delay required between an active command row address strobe (RAS) and a CAS. It is the time required between the memory controller asserting a row address, and then asserting a column address during the subsequent read or write command. RCD stands for row address to column address delay time.

Definition at line 62 of file MemoryPolicyPpi.h.

11.48.2.7 UINT16 memTiming::nRFC

Refresh to Activate Delay.

Refresh to Activate Delay or Refresh Cycle Time. The number of clocks from a Refresh command to the first Activate command.

Definition at line 174 of file MemoryPolicyPpi.h.

11.48.2.8 UINT8 memTiming::nRP

Row Precharge.

RP (row precharge) time is the number of clock cycles needed to terminate access to an open row of memory, and open access to the next row.

Definition at line 50 of file MemoryPolicyPpi.h.

11.48.2.9 UINT8 memTiming::nRRD

Row to Row Delay.

Active to Active Delay, Row to Row Delay or RAS to RAS Delay. The amount of cycles that taken to activate the next bank of memory.

Definition at line 71 of file MemoryPolicyPpi.h.

11.48.2.10 UINT8 memTiming::nRTP

Read To Precharge delay.

The number of clocks between a read command to a row pre-charge command.

Definition at line 99 of file MemoryPolicyPpi.h.

11.48.2.11 UINT8 memTiming::nWR

Write Recovery time.

The amount of cycles that are required between a valid write operation and precharge, to make sure that data is written properly.

Definition at line 108 of file MemoryPolicyPpi.h.

11.48.2.12 UINT8 memTiming::nWTR

Write to Read Delay.

Write to Read Delay. The amount of cycles required between a valid write command and the next read command.

Definition at line 81 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.49 PCH_DCI_CONFIG Struct Reference

This structure contains the policies which are related to Direct Connection Interface (DCI).

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [DciEn](#): 1
(Test) DCI enable (HDCIEN bit) when Enabled, allow DCI to be enabled.
- UINT32 [DciAutoDetect](#): 1
(Test) When set to Auto detect mode, it detects DCI being connected during BIOS post time and enable DCI.
- UINT32 [RsvdBits](#): 30
Reserved bits.

11.49.1 Detailed Description

This structure contains the policies which are related to Direct Connection Interface (DCI).

Definition at line 1742 of file PchPolicyCommon.h.

11.49.2 Member Data Documentation

11.49.2.1 UINT32 PCH_DCI_CONFIG::DciAutoDetect

(Test) When set to Auto detect mode, it detects DCI being connected during BIOS post time and enable DCI.

Else it disable DCI. This policy only apply when DciEn is disabled. NOTE: this policy should not be visible to end customer. 0: Disable AUTO mode, 1: **Enable AUTO mode**

Definition at line 1757 of file PchPolicyCommon.h.

11.49.2.2 UINT32 PCH_DCI_CONFIG::DciEn

(Test) DCI enable (HDCIEN bit) when Enabled, allow DCI to be enabled.

When Disabled, the Host control is not enabling DCI feature. BIOS provides policy to enable or disable DCI, and user would be able to use BIOS option to change this policy. The user changing the setting from disable to enable, is taken as a consent from the user to enable this DCI feature. **0:Disabled**; 1:Enabled

Definition at line 1750 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.50 PCH_DEVICE_INTERRUPT_CONFIG Struct Reference

The [PCH_DEVICE_INTERRUPT_CONFIG](#) block describes interrupt pin, IRQ and interrupt mode for PCH device.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- [UINT8 Device](#)
Device number.
- [UINT8 Function](#)
Device function.
- [UINT8 IntX](#)
Interrupt pin: INTA-INTD (see PCH_INT_PIN)
- [UINT8 Irq](#)
IRQ to be set for device.

11.50.1 Detailed Description

The [PCH_DEVICE_INTERRUPT_CONFIG](#) block describes interrupt pin, IRQ and interrupt mode for PCH device.

Definition at line 1415 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.51 PCH_DMI_CONFIG Struct Reference

The [PCH_DMI_CONFIG](#) block describes the expected configuration of the PCH for DMI.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [DmiAspm](#): 1
*0: Disable; 1: **Enable** ASPM on PCH side of the DMI Link.*
- UINT32 [PwrOptEnable](#): 1
*0: **Disable**; 1: Enable DMI Power Optimizer on PCH side.*
- UINT32 [Rsvd0](#) [6]
Reserved bytes.

11.51.1 Detailed Description

The [PCH_DMI_CONFIG](#) block describes the expected configuration of the PCH for DMI.
Definition at line 1347 of file PchPolicyCommon.h.

11.51.2 Member Data Documentation

11.51.2.1 UINT32 PCH_DMI_CONFIG::DmiAspm

0: Disable; 1: **Enable** ASPM on PCH side of the DMI Link.

While DmiAspm is enabled, DMI ASPM will be set to Intel recommended value.

Definition at line 1352 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

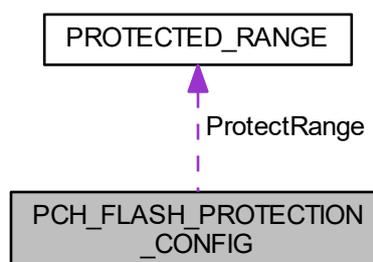
- [PchPolicyCommon.h](#)

11.52 PCH_FLASH_PROTECTION_CONFIG Struct Reference

PCH Flash Protection Configuration.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_FLASH_PROTECTION_CONFIG:



11.52.1 Detailed Description

PCH Flash Protection Configuration.

Definition at line 1686 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.53 PCH_GBL2HOST_EN Union Reference

This [PCH_GBL2HOST_EN](#) specifies enable bits related to the "Convert Global Resets to Host Resets" (G2H) feature.

```
#include <PchPolicyCommon.h>
```

11.53.1 Detailed Description

This [PCH_GBL2HOST_EN](#) specifies enable bits related to the "Convert Global Resets to Host Resets" (G2H) feature.

Definition at line 1128 of file PchPolicyCommon.h.

The documentation for this union was generated from the following file:

- [PchPolicyCommon.h](#)

11.54 PCH_GENERAL_CONFIG Struct Reference

PCH General Configuration.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT16 [SubSystemVendorId](#)
Subsystem Vendor ID and Subsystem ID of the PCH devices.
- UUINT16 [SubSystemId](#)
*Default Subsystem ID of the PCH devices. Default is **0x7270***
- UUINT32 [Crid](#): 1
This member describes whether or not the Compatibility Revision ID (CRID) feature of PCH should be enabled.
- UUINT32 [RsvdBits0](#): 29
Reserved bits.
- UUINT32 [Rsvd0](#) [2]
Reserved bytes.

11.54.1 Detailed Description

PCH General Configuration.

Definition at line 29 of file PchPolicyCommon.h.

11.54.2 Member Data Documentation

11.54.2.1 UINT32 PCH_GENERAL_CONFIG::Crid

This member describes whether or not the Compatibility Revision ID (CRID) feature of PCH should be enabled.

0: Disable; 1: Enable

Definition at line 41 of file PchPolicyCommon.h.

11.54.2.2 UINT16 PCH_GENERAL_CONFIG::SubSystemVendorId

Subsystem Vendor ID and Subsystem ID of the PCH devices.

This fields will be ignored if the value of SubSystemVendorId and SubSystemId are both 0. Default Subsystem Vendor ID of the PCH devices. Default is **0x8086**

Definition at line 35 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.55 PCH_HDAUDIO_CONFIG Struct Reference

This structure contains the policies which are related to HD Audio device (cAVS).

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 2
This member describes whether or not Intel HD Audio (Azalia) should be enabled.
- UINT32 [DspEnable](#): 1
*DSP enablement: 0: Disable; 1: **Enable***
- UINT32 [Pme](#): 1
*Azalia wake-on-ring, 0: **Disable**; 1: Enable.*
- UINT32 [IoBufferOwnership](#): 2
*I/O Buffer Ownership Select: 0: **HD-A Link**; 1: Shared, HD-A Link and I2S Port; 3: I2S Ports.*
- UINT32 [IoBufferVoltage](#): 1
*I/O Buffer Voltage Mode Select: 0: **3.3V**; 1: 1.8V.*
- UINT32 [VcType](#): 1
*Virtual Channel Type Select: 0: **VC0**, 1: VC1.*
- UINT32 [HdAudioLinkFrequency](#): 4
*HDA-Link frequency (PCH_HDAUDIO_LINK_FREQUENCY enum): 2: **24MHz**, 1: 12MHz, 0: 6MHz.*
- UINT32 [IDispLinkFrequency](#): 4
*iDisp-Link frequency (PCH_HDAUDIO_LINK_FREQUENCY enum): 4: **96MHz**, 3: 48MHz*
- UINT32 [IDispLinkTmode](#): 1
*iDisp-Link T-Mode (PCH_HDAUDIO_IDISP_TMODE enum): 0: **2T**, 1: 1T*
- UINT32 [DspUaaCompliance](#): 1
*Universal Audio Architecture compliance for DSP enabled system: 0: **Not-UAA Compliant (Intel SST driver supported only)**, 1: UAA Compliant (HDA Inbox driver or SST driver supported)*
- UINT32 [IDispCodecDisconnect](#): 1
*iDisplay Audio Codec disconnection, 0: **Not disconnected, enumerable**; 1: Disconnected SDI, not enumerable*
- UINT32 [RsvdBits0](#): 13

- Reserved bits 1.*
- UINT32 [DspEndpointDmic](#): 2
 - Bitmask of supported DSP endpoint configuration exposed via NHLT ACPI table:*
- UINT32 [DspEndpointBluetooth](#): 1
 - Bluetooth enablement: 0: **Disable**; 1: Enable.*
- UINT32 [DspEndpointI2s](#): 1
 - I2S enablement: 0: **Disable**; 1: Enable.*
- UINT32 [RsvdBits1](#): 28
 - Reserved bits 2.*
- UINT32 [DspFeatureMask](#)
 - Bitmask of supported DSP features: [BIT0] - WoV; [BIT1] - BT Sideband; [BIT2] - Codec VAD; [BIT5] - BT Intel HFP; [BIT6] - BT Intel A2DP [BIT7] - DSP based speech pre-processing disabled; [BIT8] - 0: Intel WoV, 1: Windows Voice Activation Default is **zero**.*
- UINT32 [DspPpModuleMask](#)
 - Bitmask of supported DSP Pre/Post-Processing Modules.*
- UINT16 [ResetWaitTimer](#)
 - (Test) The delay timer after Azalia reset, the value is number of microseconds. Default is **600**.*
- UINT8 [Rsvd0](#) [2]
 - Reserved bytes, align to multiple 4.*

11.55.1 Detailed Description

This structure contains the policies which are related to HD Audio device (cAVS).

Definition at line 784 of file PchPolicyCommon.h.

11.55.2 Member Data Documentation

11.55.2.1 UINT32 PCH_HDAUDIO_CONFIG::DspEndpointDmic

Bitmask of supported DSP endpoint configuration exposed via NHLT ACPI table:

DMIC Select (PCH_HDAUDIO_DMIC_TYPE enum): 0: Disable; 1: 2ch array; **2: 4ch array**; 3: 1ch array

Definition at line 811 of file PchPolicyCommon.h.

11.55.2.2 UINT32 PCH_HDAUDIO_CONFIG::DspPpModuleMask

Bitmask of supported DSP Pre/Post-Processing Modules.

Specific pre/post-processing module bit position must be coherent with the ACPI implementation: `_SB.PCI0.HDA↔S._DSM` Function 3: Query Pre/Post Processing Module Support. `DspPpModuleMask` is passed to ACPI as 'ADPM' NVS variable Default is **zero**.

Definition at line 829 of file PchPolicyCommon.h.

11.55.2.3 UINT32 PCH_HDAUDIO_CONFIG::Enable

This member describes whether or not Intel HD Audio (Azalia) should be enabled.

If enabled (in Auto mode) and no codec exists the reference code will automatically disable the HD Audio device. 0: Disable, 1: Enable, **2: Auto (enabled if codec detected and initialized, disabled otherwise)**

Definition at line 791 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.56 PCH_HPET_CONFIG Struct Reference

The [PCH_HPET_CONFIG](#) block passes the bus/device/function value for HPET.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 1
Determines if enable HPET timer.
- UINT32 [BdfValid](#): 1
*Whether the BDF value is valid. 0: **Disable**; 1: **Enable**.*
- UINT32 [RsvdBits0](#): 6
Reserved bits.
- UINT32 [BusNumber](#): 8
*Bus Number HPETn used as Requestor / Completer ID. Default is **0xF0**.*
- UINT32 [DeviceNumber](#): 5
*Device Number HPETn used as Requestor / Completer ID. Default is **0x1F**.*
- UINT32 [FunctionNumber](#): 3
*Function Number HPETn used as Requestor / Completer ID. Default is **0x00**.*
- UINT32 [RsvdBits1](#): 8
Reserved bits.
- UINT32 [Base](#)
*The HPET base address. Default is **0xFED00000**.*

11.56.1 Detailed Description

The [PCH_HPET_CONFIG](#) block passes the bus/device/function value for HPET.

The address resource range of HPET must be reserved in E820 and ACPI as system resource.

Definition at line 719 of file `PchPolicyCommon.h`.

11.56.2 Member Data Documentation

11.56.2.1 UINT32 PCH_HPET_CONFIG::Enable

Determines if enable HPET timer.

0: **Disable**; 1: **Enable**. The HPET timer address decode is always enabled. This policy is used to configure the HPET timer count, and also the `_STA` of HPET device in ACPI. While enabled, the HPET timer is started, else the HPET timer is halted.

Definition at line 726 of file `PchPolicyCommon.h`.

The documentation for this struct was generated from the following file:

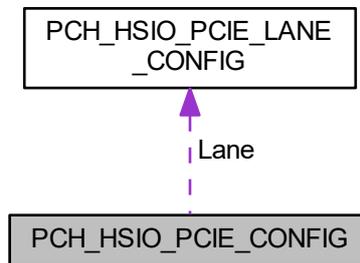
- [PchPolicyCommon.h](#)

11.57 PCH_HSIO_PCIE_CONFIG Struct Reference

The [PCH_HSIO_PCIE_CONFIG](#) block describes the configuration of the HSIO for PCIe lanes.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_HSIO_PCIE_CONFIG:



Public Attributes

- [PCH_HSIO_PCIE_LANE_CONFIG Lane](#) [PCH_MAX_PCIE_ROOT_PORTS]
These members describe the configuration of HSIO for PCIe lanes.
- UINT32 [Rsvd0](#) [3]
Reserved bytes.

11.57.1 Detailed Description

The [PCH_HSIO_PCIE_CONFIG](#) block describes the configuration of the HSIO for PCIe lanes.

Definition at line 416 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.58 PCH_HSIO_PCIE_LANE_CONFIG Struct Reference

The [PCH_HSIO_PCIE_LANE_CONFIG](#) describes HSIO settings for PCIe lane.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [HsioRxSetCtleEnable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 3 Set CTLE Value
- UINT32 [HsioRxSetCtle](#): 6
PCH PCIe Gen 3 Set CTLE Value.
- UINT32 [HsioTxGen1DownscaleAmpEnable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 1 TX Output Downscale Amplitude Adjustment value override
- UINT32 [HsioTxGen1DownscaleAmp](#): 6
PCH PCIe Gen 1 TX Output Downscale Amplitude Adjustment value.
- UINT32 [HsioTxGen2DownscaleAmpEnable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 2 TX Output Downscale Amplitude Adjustment value override

- UINT32 [HsioTxGen2DownscaleAmp](#): 6
PCH PCIe Gen 2 TX Output Downscale Amplitude Adjustment value.
- UINT32 [HsioTxGen3DownscaleAmpEnable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 3 TX Output Downscale Amplitude Adjustment value override
- UINT32 [HsioTxGen3DownscaleAmp](#): 6
PCH PCIe Gen 3 TX Output Downscale Amplitude Adjustment value.
- UINT32 [RsvdBits0](#): 4
Reserved Bits.
- UINT32 [HsioTxGen1DeEmphEnable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 1 TX Output De-Emphasis Adjustment Setting value override
- UINT32 [HsioTxGen1DeEmph](#): 6
PCH PCIe Gen 1 TX Output De-Emphasis Adjustment Setting.
- UINT32 [HsioTxGen2DeEmph3p5Enable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 2 TX Output -3.5dB Mode De-Emphasis Adjustment Setting value override
- UINT32 [HsioTxGen2DeEmph3p5](#): 6
PCH PCIe Gen 2 TX Output -3.5dB Mode De-Emphasis Adjustment Setting.
- UINT32 [HsioTxGen2DeEmph6p0Enable](#): 1
0: Disable; 1: Enable PCH PCIe Gen 2 TX Output -6.0dB Mode De-Emphasis Adjustment Setting value override
- UINT32 [HsioTxGen2DeEmph6p0](#): 6
PCH PCIe Gen 2 TX Output -6.0dB Mode De-Emphasis Adjustment Setting.
- UINT32 [RsvdBits1](#): 11
Reserved Bits.
- UINT32 [HsioIcFgAdjLimitLo](#): 5
< **0: Disable**; 1: Enable Set the floor on how many ticks the autovref can take.
- UINT32 [HsioSampOffstEvenErrSpEnable](#): 1
< Set the floor on how many ticks the autovref can take. (offset 0x9c)
- UINT32 [HsioSampOffstEvenErrSp](#): 8
< **0: Disable**; 1: Enable EVEN ERR P sampler manual offset.
- UINT32 [RsvdBits2](#): 17
< EVEN ERR P sampler manual offset. (offset 0xA0)
- UINT32 [HsioRemainingSamplerOff](#): 24
< **0: Disable**; 1: Enable Remaining EVEN/ODD ERR P and N sampler manual offset.
- UINT32 [HsioVgaGainCalEnable](#): 1
< Remaining EVEN/ODD ERR P and N sampler manual offset. (offset 0xA4)
- UINT32 [HsioVgaGainCal](#): 5
< **0: Disable**; 1: Enable VGA Gain CAL
- UINT32 [RsvdBits3](#): 1
< VGA Gain Calibration Value (offset 0x1C)
- UINT32 [Rsvd4](#) [12]
Reserved bytes.

11.58.1 Detailed Description

The [PCH_HSIO_PCIE_LANE_CONFIG](#) describes HSIO settings for PCIe lane.

Definition at line 370 of file PchPolicyCommon.h.

11.58.2 Member Data Documentation

11.58.2.1 UINT32 PCH_HSIO_PCIE_LANE_CONFIG::RsvdBits2

< EVEN ERR P sampler manual offset. (offset 0xA0)

Reserved Bits

Definition at line 401 of file PchPolicyCommon.h.

11.58.2.2 UINT32 PCH_HSIO_PCIE_LANE_CONFIG::RsvdBits3

< VGA Gain Calibration Value (offset 0x1C)

Reserved Bits

Definition at line 407 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

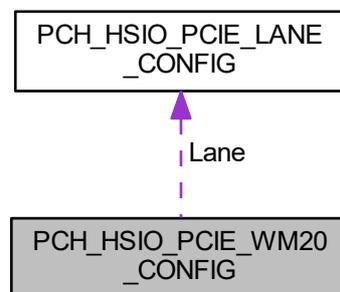
- [PchPolicyCommon.h](#)

11.59 PCH_HSIO_PCIE_WM20_CONFIG Struct Reference

The [PCH_HSIO_PCIE_WM20_CONFIG](#) block describes the configuration of the HSIO for PCIe lanes.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_HSIO_PCIE_WM20_CONFIG:



Public Attributes

- [PCH_HSIO_PCIE_LANE_CONFIG Lane](#) [PCH_MAX_WM20_LANES_NUMBER]
These members describe the configuration of HSIO for PCIe lanes.
- `UINT32 Rsvd0` [3]
Reserved bytes.

11.59.1 Detailed Description

The [PCH_HSIO_PCIE_WM20_CONFIG](#) block describes the configuration of the HSIO for PCIe lanes.

Definition at line 429 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

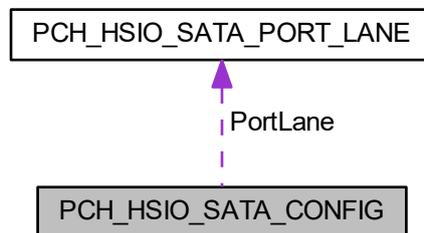
- [PchPolicyCommon.h](#)

11.60 PCH_HSIO_SATA_CONFIG Struct Reference

The [PCH_HSIO_SATA_CONFIG](#) block describes the HSIO configuration of the SATA controller.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_HSIO_SATA_CONFIG:



Public Attributes

- [PCH_HSIO_SATA_PORT_LANE](#) [PortLane](#) [PCH_MAX_SATA_PORTS]
These members describe the configuration of HSIO for SATA lanes.
- UINT32 [Rsvd0](#) [8]
Reserved bytes.

11.60.1 Detailed Description

The [PCH_HSIO_SATA_CONFIG](#) block describes the HSIO configuration of the SATA controller.

Definition at line 671 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.61 PCH_HSIO_SATA_PORT_LANE Struct Reference

The [PCH_HSIO_SATA_PORT_LANE](#) describes HSIO settings for SATA Port lane.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [HsioRxGen1EqBoostMagEnable](#): 1
0: Disable; 1: Enable Receiver Equalization Boost Magnitude Adjustment Value override
- UINT32 [HsioRxGen1EqBoostMag](#): 6
SATA 1.5 Gb/s Receiver Equalization Boost Magnitude Adjustment value.
- UINT32 [HsioRxGen2EqBoostMagEnable](#): 1
0: Disable; 1: Enable Receiver Equalization Boost Magnitude Adjustment Value override
- UINT32 [HsioRxGen2EqBoostMag](#): 6
SATA 3.0 Gb/s Receiver Equalization Boost Magnitude Adjustment value.
- UINT32 [HsioRxGen3EqBoostMagEnable](#): 1
0: Disable; 1: Enable Receiver Equalization Boost Magnitude Adjustment Value override
- UINT32 [HsioRxGen3EqBoostMag](#): 6
SATA 6.0 Gb/s Receiver Equalization Boost Magnitude Adjustment value.
- UINT32 [HsioTxGen1DownscaleAmpEnable](#): 1
0: Disable; 1: Enable SATA 1.5 Gb/s TX Output Downscale Amplitude Adjustment value override
- UINT32 [HsioTxGen1DownscaleAmp](#): 6
SATA 1.5 Gb/s TX Output Downscale Amplitude Adjustment value.
- UINT32 [RsvdBits0](#): 4
Reserved bits.
- UINT32 [HsioTxGen2DownscaleAmpEnable](#): 1
0: Disable; 1: Enable SATA 3.0 Gb/s TX Output Downscale Amplitude Adjustment value override
- UINT32 [HsioTxGen2DownscaleAmp](#): 6
SATA 3.0 Gb/s TX Output Downscale Amplitude Adjustment.
- UINT32 [HsioTxGen3DownscaleAmpEnable](#): 1
0: Disable; 1: Enable SATA 6.0 Gb/s TX Output Downscale Amplitude Adjustment value override
- UINT32 [HsioTxGen3DownscaleAmp](#): 6
SATA 6.0 Gb/s TX Output Downscale Amplitude Adjustment.
- UINT32 [HsioTxGen1DeEmphEnable](#): 1
0: Disable; 1: Enable SATA 1.5 Gb/s TX Output De-Emphasis Adjustment Setting value override
- UINT32 [HsioTxGen1DeEmph](#): 6
SATA 1.5 Gb/s TX Output De-Emphasis Adjustment Setting.
- UINT32 [HsioTxGen2DeEmphEnable](#): 1
0: Disable; 1: Enable SATA 3.0 Gb/s TX Output De-Emphasis Adjustment Setting value override
- UINT32 [HsioTxGen2DeEmph](#): 6
SATA 3.0 Gb/s TX Output De-Emphasis Adjustment Setting.
- UINT32 [RsvdBits1](#): 4
Reserved bits.
- UINT32 [HsioTxGen3DeEmphEnable](#): 1
0: Disable; 1: Enable SATA 6.0 Gb/s TX Output De-Emphasis Adjustment Setting value override
- UINT32 [HsioTxGen3DeEmph](#): 6
SATA 6.0 Gb/s TX Output De-Emphasis Adjustment Setting value override.
- UINT32 [RsvdBits2](#): 25
Reserved bits.
- UINT32 [Rsvd0](#) [8]
Reserved bytes.

11.61.1 Detailed Description

The [PCH_HSIO_SATA_PORT_LANE](#) describes HSIO settings for SATA Port lane.

Definition at line 630 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

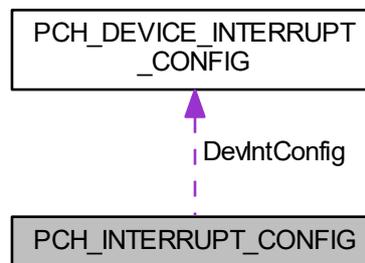
- [PchPolicyCommon.h](#)

11.62 PCH_INTERRUPT_CONFIG Struct Reference

The [PCH_INTERRUPT_CONFIG](#) block describes interrupt settings for PCH.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_INTERRUPT_CONFIG:



Public Attributes

- UINT8 [NumOfDevIntConfig](#)
Number of entries in DevIntConfig table.
- UINT8 [Rsvd0](#) [2]
Reserved bytes, align to multiple 4.
- [PCH_DEVICE_INTERRUPT_CONFIG DevIntConfig](#) [[PCH_MAX_DEVICE_INTERRUPT_CONFIG](#)]
Array which stores PCH devices interrupts settings.
- UINT8 [PxRcConfig](#) [[PCH_MAX_PXRC_CONFIG](#)]
Array which stores interrupt routing for 8259 controller.
- UINT8 [GpioIrqRoute](#)
*Interrupt routing for GPIO. Default is **14**.*
- UINT8 [ScilrqSelect](#)
*Interrupt select for SCI. Default is **9**.*
- UINT8 [TcolrqSelect](#)
*Interrupt select for TCO. Default is **9**.*
- UINT8 [TcolrqEnable](#)
*Enable IRQ generation for TCO. **0: Disable**; 1: Enable.*
- UINT8 [ShutdownPolicySelect](#)
Shutdown mode 0: PCH will drive INIT#; 1: PCH will drive PLTRST# active.

11.62.1 Detailed Description

The [PCH_INTERRUPT_CONFIG](#) block describes interrupt settings for PCH.

Definition at line 1428 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.63 PCH_IOAPIC_CONFIG Struct Reference

The [PCH_IOAPIC_CONFIG](#) block describes the expected configuration of the PCH IO APIC, it's optional and PCH code would ignore it if the BdfValid bit is not TRUE.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [BdfValid](#): 1
Set to 1 if BDF value is valid, PCH code will not program these fields if this bit is not TRUE. 0: Disable; 1: Enable.
- UUINT32 [RsvdBits0](#): 7
Reserved bits.
- UUINT32 [BusNumber](#): 8
*Bus/Device/Function used as Requestor / Completer ID. Default is **0xF0**.*
- UUINT32 [DeviceNumber](#): 5
*Bus/Device/Function used as Requestor / Completer ID. Default is **0x1F**.*
- UUINT32 [FunctionNumber](#): 3
*Bus/Device/Function used as Requestor / Completer ID. Default is **0x00**.*
- UUINT32 [IoApicEntry24_119](#): 1
0: Disable; 1: Enable IOAPIC Entry 24-119
- UUINT32 [RsvdBits1](#): 7
Reserved bits.
- UUINT8 [IoApicId](#)
*This member determines IOAPIC ID. Default is **0x02**.*
- UUINT8 [ApicRangeSelect](#)
*Define address bits 19:12 for the IOxAPIC range. Default is **0***
- UUINT8 [Rsvd0](#) [2]
Reserved bytes.

11.63.1 Detailed Description

The [PCH_IOAPIC_CONFIG](#) block describes the expected configuration of the PCH IO APIC, it's optional and PCH code would ignore it if the BdfValid bit is not TRUE.

Bus:device:function fields will be programmed to the register P2SB IBDF(P2SB PCI offset R6Ch-6Dh), it's using for the following purpose: As the Requester ID when initiating Interrupt Messages to the processor. As the Completer ID when responding to the reads targeting the IOxAPI's Memory-Mapped I/O registers. This field defaults to Bus 0: Device 31: Function 0 after reset. BIOS can program this field to provide a unique Bus:Device:Function number for the internal IOxAPIC. The address resource range of IOAPIC must be reserved in E820 and ACPI as system resource.

Definition at line 697 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.64 PCH_LAN_CONFIG Struct Reference

PCH intergrated LAN controller configuration settings.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [Enable](#): 1
*Determines if enable PCH internal LAN, 0: Disable; 1: **Enable**.*
- UUINT32 [K1OffEnable](#): 1
*Use CLKREQ for GbE power management; 1: Enabled, 0: **Disabled**;*
- UUINT32 [RsvdBits0](#): 4
Reserved bits.
- UUINT32 [ClkReqSupported](#): 1
*Indicate whether dedicated CLKREQ# is supported; 1: Enabled, 0: **Disabled**;*
- UUINT32 [ClkReqNumber](#): 4
CLKREQ# used by GbE. Valid if ClkReqSupported is TRUE.
- UUINT32 [RsvdBits1](#): 21
Reserved bits.
- UUINT32 [Rsvd0](#)
Reserved bytes.

11.64.1 Detailed Description

PCH intergrated LAN controller configuration settings.

Definition at line 841 of file PchPolicyCommon.h.

11.64.2 Member Data Documentation

11.64.2.1 UUINT32 PCH_LAN_CONFIG::Enable

Determines if enable PCH internal LAN, 0: Disable; 1: **Enable**.

When Enable is changed (from disabled to enabled or from enabled to disabled), it needs to set LAN Disable register, which might be locked by FDSWL register. So it's recommended to issue a global reset when changing the status for PCH Internal LAN.

Definition at line 848 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.65 PCH_LOCK_DOWN_CONFIG Struct Reference

The [PCH_LOCK_DOWN_CONFIG](#) block describes the expected configuration of the PCH for security requirement.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [GlobalSmi](#): 1
(Test) Enable SMI_LOCK bit to prevent writes to the Global SMI Enable bit.
- UUINT32 [BiosInterface](#): 1
(Test) 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.
- UUINT32 [RtcLock](#): 1
(Test) Enable RTC lower and upper 128 byte Lock bits to lock Bytes 38h-3Fh in the upper and lower 128-byte bank of RTC RAM.
- UUINT32 [BiosLock](#): 1
Enable the BIOS Lock Enable (BLE) feature and set EISS bit (D31:F5:RegDCh[5]) for the BIOS region protection.
- UUINT32 [SpiEiss](#): 1
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.
- UUINT32 [GpioLockDown](#): 1
Lock configuration and/or state of vendor-defined set of GPIOs.
- UUINT32 [TcoLock](#): 1
Lock TCO Base Address.
- UUINT32 [EvaLockDown](#): 1
(Test) Enable Lock bit for Device Function Hide Register in MS Unit Device Function Hide Control Register (MSD↔EVFUNCHIDE) 0: Disable; 1: **Enable**.
- UUINT32 [RsvdBits0](#): 24
Reserved bits.

11.65.1 Detailed Description

The [PCH_LOCK_DOWN_CONFIG](#) block describes the expected configuration of the PCH for security requirement. Definition at line 891 of file PchPolicyCommon.h.

11.65.2 Member Data Documentation

11.65.2.1 UUINT32 PCH_LOCK_DOWN_CONFIG::BiosInterface

(Test) 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.

0: Disable; 1: **Enable**.

Definition at line 900 of file PchPolicyCommon.h.

11.65.2.2 UUINT32 PCH_LOCK_DOWN_CONFIG::BiosLock

Enable the BIOS Lock Enable (BLE) feature and set EISS bit (D31:F5:RegDCh[5]) for the BIOS region protection.

When it is enabled, the BIOS Region can only be modified from SMM after EndOfDxe protocol is installed. Note: When BiosLock is enabled, platform code also needs to update to take care of BIOS modification (including Set↔Variable) in DXE or runtime phase after EndOfDxe protocol is installed. **0: Disable**; 1: Enable.

Definition at line 914 of file PchPolicyCommon.h.

11.65.2.3 UUINT32 PCH_LOCK_DOWN_CONFIG::GlobalSmi

(Test) Enable SMI_LOCK bit to prevent writes to the Global SMI Enable bit.

0: Disable; 1: **Enable**.

Definition at line 895 of file PchPolicyCommon.h.

11.65.2.4 UUINT32 PCH_LOCK_DOWN_CONFIG::GpioLockDown

Lock configuration and/or state of vendor-defined set of GPIOs.

0: Don't lock; 1: Lock

Definition at line 930 of file PchPolicyCommon.h.

11.65.2.5 UUINT32 PCH_LOCK_DOWN_CONFIG::RtcLock

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

0: Disable; 1: **Enable**.

Definition at line 905 of file PchPolicyCommon.h.

11.65.2.6 UUINT32 PCH_LOCK_DOWN_CONFIG::SpiEiss

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. In recovery path, platform can temporary disable EISS for SPI programming in PEI phase or early DXE phase. 0: Clear EISS bit; 1: **Set EISS bit**.

Definition at line 925 of file PchPolicyCommon.h.

11.65.2.7 UUINT32 PCH_LOCK_DOWN_CONFIG::TcoLock

Lock TCO Base Address.

D31:F4 (SMBus Controller) Offset 54h: TCOCTL (TCO Control Register) Bit 0: TCO_BASE_LOCK (TCO Base Lock) 0: Don't lock; 1: Lock

Definition at line 936 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.66 PCH_LPC_CONFIG Struct Reference

This structure contains the policies which are related to LPC.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [EnhancePort8xhDecoding](#): 1
Enhance the port 8xh decoding.

- UINT32 [RsvdBits](#): 30
Reserved bits.

11.66.1 Detailed Description

This structure contains the policies which are related to LPC.

Definition at line 1767 of file PchPolicyCommon.h.

11.66.2 Member Data Documentation

11.66.2.1 UINT32 PCH_LPC_CONFIG::EnhancePort8xhDecoding

Enhance the port 8xh decoding.

Original LPC only decodes one byte of port 80h, with this enhancement LPC can decode word or dword of port 80h-83h.

Note

: this will occupy one LPC generic IO range register. While this is enabled, read from port 80h always return 0x00. 0: Disable, 1: **Enable**

Definition at line 1774 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.67 PCH_LPC_SIRQ_CONFIG Struct Reference

The [PCH_LPC_SIRQ_CONFIG](#) block describes the expected configuration of the PCH for Serial IRQ.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [SirqEnable](#): 1
*Determines if enable Serial IRQ. 0: Disable; 1: **Enable**.*
- UINT32 [RsvdBits0](#): 31
Reserved bits.
- PCH_SIRQ_MODE [SirqMode](#)
*Serial IRQ Mode Select. 0: **quiet mode** 1: continuous mode.*
- [PCH_START_FRAME_PULSE](#) [StartFramePulse](#)
*Start Frame Pulse Width. Default is **PchSfpw4Clk**.*
- UINT32 [Rsvd0](#)
Reserved bytes.

11.67.1 Detailed Description

The [PCH_LPC_SIRQ_CONFIG](#) block describes the expected configuration of the PCH for Serial IRQ.

Definition at line 1380 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

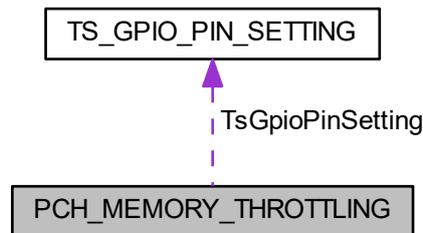
- [PchPolicyCommon.h](#)

11.68 PCH_MEMORY_THROTTLING Struct Reference

This structure supports an external memory thermal sensor (TS-on-DIMM or TS-on-Board).

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_MEMORY_THROTTLING:



Public Attributes

- [UINT32 Enable](#): 1
This will enable PCH memory throttling.
- [TS_GPIO_PIN_SETTING TsGpioPinSetting](#) [2]
GPIO_C and GPIO_D selection for memory throttling.

11.68.1 Detailed Description

This structure supports an external memory thermal sensor (TS-on-DIMM or TS-on-Board).

Definition at line 1056 of file PchPolicyCommon.h.

11.68.2 Member Data Documentation

11.68.2.1 [UINT32 PCH_MEMORY_THROTTLING::Enable](#)

This will enable PCH memory throttling.

While this policy is enabled, must also enable EnableExtts in SA policy. **0: Disable**; 1: Enable

Definition at line 1062 of file PchPolicyCommon.h.

11.68.2.2 [TS_GPIO_PIN_SETTING PCH_MEMORY_THROTTLING::TsGpioPinSetting](#)[2]

GPIO_C and GPIO_D selection for memory throttling.

It's strongly recommended to choose GPIO_C and GPIO_D for memory throttling feature, and route EXTTS# accordingly.

Definition at line 1069 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.69 PCH_P2SB_CONFIG Struct Reference

This structure contains the policies which are related to P2SB device.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [SbiUnlock](#): 1
(Test) This unlock the SBI lock bit to allow SBI after post time.
- UUINT32 [PsfUnlock](#): 1
(Test) The PSF registers will be locked before 3rd party code execution.
- UUINT32 [P2SbReveal](#): 1
Debug The P2SB PCIe device will be hidden at end of PEI stage.

11.69.1 Detailed Description

This structure contains the policies which are related to P2SB device.

Definition at line 1712 of file PchPolicyCommon.h.

11.69.2 Member Data Documentation

11.69.2.1 UUINT32 PCH_P2SB_CONFIG::P2SbReveal

Debug The P2SB PCIe device will be hidden at end of PEI stage.

This policy reveal P2SB PCIe device at end of EXE. **0: Disable (hidden)**; 1: Enable (visible). NOTE: Do not set this policy "P2SbReveal" unless necessary.

Definition at line 1732 of file PchPolicyCommon.h.

11.69.2.2 UUINT32 PCH_P2SB_CONFIG::PsfUnlock

(Test) The PSF registers will be locked before 3rd party code execution.

This policy unlock the PSF space. **0: Disable**; 1: Enable. NOTE: Do not set this policy "PsfUnlock" unless necessary.

Definition at line 1725 of file PchPolicyCommon.h.

11.69.2.3 UUINT32 PCH_P2SB_CONFIG::SbiUnlock

(Test) This unlock the SBI lock bit to allow SBI after post time.

0: Disable; 1: Enable. NOTE: Do not set this policy "SbiUnlock" unless necessary.

Definition at line 1718 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

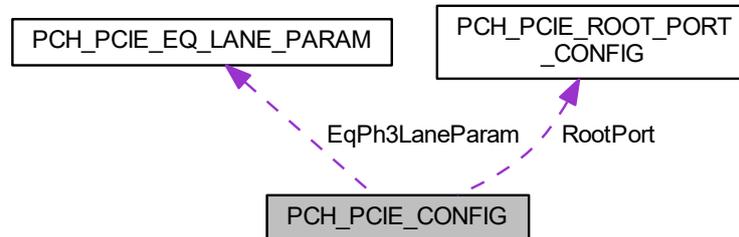
- [PchPolicyCommon.h](#)

11.70 PCH_PCIE_CONFIG Struct Reference

The [PCH_PCIE_CONFIG](#) block describes the expected configuration of the PCH PCI Express controllers.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_PCIE_CONFIG:



Public Attributes

- [PCH_PCIE_ROOT_PORT_CONFIG RootPort](#) [PCH_MAX_PCIE_ROOT_PORTS]
These members describe the configuration of each PCH PCIe root port.
- UINT8 [PciDelayOptimizationEcr](#)
Pci Delay (Latency) Optimization ECR - Engineering Change Request.
- UINT8 [MaxReadRequestSize](#)
Pch Pcie Max Read Request Size.
- [PCH_PCIE_EQ_LANE_PARAM EqPh3LaneParam](#) [PCH_MAX_PCIE_ROOT_PORTS]
Gen3 Equalization settings for physical PCIe lane, index 0 represents PCIe lane 1, etc.
- UINT32 [EnablePort8xhDecode](#): 1
(Test) This member describes whether PCIe root port Port 8xh Decode is enabled.
- UINT32 [PchPciePort8xhDecodePortIndex](#): 5
(Test) The Index of PCIe Port that is selected for Port8xh Decode (0 Based)
- UINT32 [DisableRootPortClockGating](#): 1
This member describes whether the PCI Express Clock Gating for each root port is enabled by platform modules.
- UINT32 [EnablePeerMemoryWrite](#): 1
This member describes whether Peer Memory Writes are enabled on the platform.
- UINT32 [AllowNoLtrIccPllShutdown](#): 1
This member allows BIOS to control ICC PLL Shutdown by determining PCIe devices are LTR capable or leaving untouched.
- UINT32 [ComplianceTestMode](#): 1
Compliance Test Mode shall be enabled when using Compliance Load Board.
- UINT32 [RpFunctionSwap](#): 1
RpFunctionSwap allows BIOS to use root port function number swapping when root port of function 0 is disabled.
- UINT16 [DetectTimeoutMs](#)
The number of milliseconds reference code will wait for link to exit Detect state for enabled ports before assuming there is no device and potentially disabling the port.
- UINT8 [PchPcieUX16CompletionTimeout](#)
These are Completions Timeout settings for Uplink ports in Server PCH.
- UINT8 [PchPcieUX16MaxPayload](#)
Max Payload Size settings for Uplink ports in Server PCH.
- UINT8 [VTdSupport](#)
Intel+ Virtual Technology for Directed I/O (VT-d) Support.

- UINT16 [Rsvd0](#)
Reserved bytes.
- UINT32 [Rsvd1](#) [2]
Reserved bytes.

11.70.1 Detailed Description

The [PCH_PCIE_CONFIG](#) block describes the expected configuration of the PCH PCI Express controllers.
Definition at line 245 of file PchPolicyCommon.h.

11.70.2 Member Data Documentation

11.70.2.1 UINT32 PCH_PCIE_CONFIG::AllowNoLtrccPllShutdown

This member allows BIOS to control ICC PLL Shutdown by determining PCIe devices are LTR capable or leaving untouched.

- **0: Disable, ICC PLL Shutdown is determined by PCIe device LTR capability.**
 - To allow ICC PLL shutdown if all present PCIe devices are LTR capable or if no PCIe devices are presented for maximum power savings where possible.
 - To disable ICC PLL shutdown when BIOS detects any non-LTR capable PCIe device for ensuring device functionality.
- 1: Enable, To allow ICC PLL shutdown even if some devices do not support LTR capability.

Definition at line 290 of file PchPolicyCommon.h.

11.70.2.2 UINT32 PCH_PCIE_CONFIG::ComplianceTestMode

Compliance Test Mode shall be enabled when using Compliance Load Board.

0: Disable, 1: Enable

Definition at line 295 of file PchPolicyCommon.h.

11.70.2.3 UINT16 PCH_PCIE_CONFIG::DetectTimeoutMs

The number of milliseconds reference code will wait for link to exit Detect state for enabled ports before assuming there is no device and potentially disabling the port.

It's assumed that the link will exit detect state before root port initialization (sufficient time elapsed since PLTRST de-assertion) therefore default timeout is zero. However this might be useful if device power-up sequence is controlled by BIOS or a specific device requires more time to detect. I case of non-common clock enabled the default timeout is 15ms. **Default: 0**

Definition at line 322 of file PchPolicyCommon.h.

11.70.2.4 UINT32 PCH_PCIE_CONFIG::DisableRootPortClockGating

This member describes whether the PCI Express Clock Gating for each root port is enabled by platform modules.

0: Disable; 1: Enable.

Definition at line 275 of file PchPolicyCommon.h.

11.70.2.5 UUINT32 PCH_PCIE_CONFIG::EnablePeerMemoryWrite

This member describes whether Peer Memory Writes are enabled on the platform.

0: Disable; 1: Enable.

Definition at line 279 of file PchPolicyCommon.h.

11.70.2.6 UUINT32 PCH_PCIE_CONFIG::EnablePort8xhDecode

(Test) This member describes whether PCIE root port Port 8xh Decode is enabled.

0: Disable; 1: Enable.

Definition at line 266 of file PchPolicyCommon.h.

11.70.2.7 PCH_PCIE_EQ_LANE_PARAM PCH_PCIE_CONFIG::EqPh3LaneParam[PCH_MAX_PCIE_ROOT_PORTS]

Gen3 Equalization settings for physical PCIe lane, index 0 represents PCIe lane 1, etc.

Corresponding entries are used when root port EqPh3Method is PchPcieEqStaticCoeff (default).

Definition at line 262 of file PchPolicyCommon.h.

11.70.2.8 UUINT32 PCH_PCIE_CONFIG::RpFunctionSwap

RpFunctionSwap allows BIOS to use root port function number swapping when root port of function 0 is disabled.

A PCIe device can have higher functions only when Function0 exists. To satisfy this requirement, BIOS will always enable Function0 of a device that contains more than 0 enabled root ports.

- **Enabled: One of enabled root ports get assigned to Function0.** This offers no guarantee that any particular root port will be available at a specific DevNr:FuncNr location
- **Disabled:** Root port that corresponds to Function0 will be kept visible even though it might be not used. That way rootport - to - DevNr:FuncNr assignment is constant. This option will impact ports 1, 9, 17. NOTE: This option will not work if ports 1, 9, 17 are fused or configured for RST PCIe storage NOTE: Disabling function swap may have adverse impact on power management. This option should ONLY be used when each one of root ports 1, 9, 17:
 - is configured as PCIe and has correctly configured ClkReq signal, or
 - does not own any mPhy lanes (they are configured as SATA or USB)

Definition at line 310 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

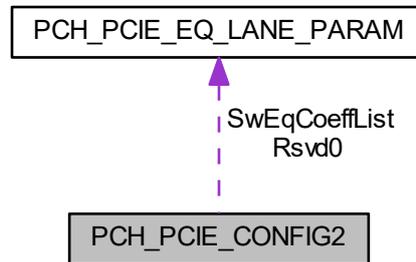
- [PchPolicyCommon.h](#)

11.71 PCH_PCIE_CONFIG2 Struct Reference

The [PCH_PCIE_CONFIG2](#) block describes the additional configuration of the PCH PCI Express controllers.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_PCIE_CONFIG2:



Public Attributes

- [PCH_PCIE_EQ_PARAM SwEqCoeffList](#) [PCH_PCIE_SWEQ_COEFFS_MAX]
List of coefficients used during equalization (applicable to both software and hardware EQ)

11.71.1 Detailed Description

The [PCH_PCIE_CONFIG2](#) block describes the additional configuration of the PCH PCI Express controllers.

Definition at line 348 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.72 PCH_PCIE_EQ_LANE_PARAM Struct Reference

Represent lane specific PCIe Gen3 equalization parameters.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT8 [Cm](#)
Coefficient C-1.
- UINT8 [Cp](#)
Coefficient C+1.
- UINT8 [Rsvd0](#) [2]
Reserved bytes.

11.72.1 Detailed Description

Represent lane specific PCIe Gen3 equalization parameters.

Definition at line 133 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.73 PCH_PCIE_ROOT_PORT_CONFIG Struct Reference

The PCH_PCIE_EXPRESS_ROOT_PORT_CONFIG describe the feature and capability of each PCH PCIe root port.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 1
*Root Port enabling, 0: Disable; 1: **Enable**.*
- UINT32 [HotPlug](#): 1
*Indicate whether the root port is hot plug available. 0: **Disable**; 1: Enable.*
- UINT32 [PmSci](#): 1
*Indicate whether the root port power manager SCI is enabled. 0: Disable; 1: **Enable**.*
- UINT32 [ExtSync](#): 1
*Indicate whether the extended synch is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [TransmitterHalfSwing](#): 1
*Indicate whether the Transmitter Half Swing is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [AcsEnabled](#): 1
*Indicate whether the ACS is enabled. 0: Disable; 1: **Enable**.*
- UINT32 [RsvdBits0](#): 5
Reserved bits.
- UINT32 [ClkReqSupported](#): 1
Indicate whether dedicated CLKREQ# is supported by the port.
- UINT32 [ClkReqNumber](#): 4
The ClkReq Signal mapped to this root port.
- UINT32 [ClkReqDetect](#): 1
Probe CLKREQ# signal before enabling CLKREQ# based power management.
- UINT32 [AdvancedErrorReporting](#): 1
*Indicate whether the Advanced Error Reporting is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [RsvdBits1](#): 3
Reserved fields for future expansion w/o protocol change.
- UINT32 [UnsupportedRequestReport](#): 1
*Indicate whether the Unsupported Request Report is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [FatalErrorReport](#): 1
*Indicate whether the Fatal Error Report is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [NoFatalErrorReport](#): 1
*Indicate whether the No Fatal Error Report is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [CorrectableErrorReport](#): 1
*Indicate whether the Correctable Error Report is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [SystemErrorOnFatalError](#): 1
*Indicate whether the System Error on Fatal Error is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [SystemErrorOnNonFatalError](#): 1
*Indicate whether the System Error on Non Fatal Error is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [SystemErrorOnCorrectableError](#): 1
*Indicate whether the System Error on Correctable Error is enabled. 0: **Disable**; 1: Enable.*
- UINT32 [MaxPayload](#): 2
*Max Payload Size supported, Default **128B**, see enum PCH_PCIE_MAX_PAYLOAD Changes Max Payload Size Supported field in Device Capabilities of the root port.*

- UINT32 [SlotImplemented](#): 1
Indicates how this root port is connected to endpoint.
- UINT32 [DeviceResetPadActiveHigh](#): 1
*Indicated whether PERST# is active **0: Low**; 1: High, See: DeviceResetPad.*
- UINT8 [PcieSpeed](#)
Determines each PCIE Port speed capability.
- UINT8 [Gen3EqPh3Method](#)
PCIe Gen3 Equalization Phase 3 Method (see PCH_PCIE_EQ_METHOD).
- UINT8 [PhysicalSlotNumber](#)
Indicates the slot number for the root port. Default is the value as root port index.
- UINT8 [CompletionTimeout](#)
*The completion timeout configuration of the root port (see: PCH_PCIE_COMPLETION_TIMEOUT). Default is **PchPcieCompletionTO_Default**.*
- UINT32 [DeviceResetPad](#)
The PCH pin assigned to device PERST# signal if available, zero otherwise.
- UINT32 [Rsvd1](#)
Reserved bytes.
- UINT8 [Aspm](#)
*The ASPM configuration of the root port (see: PCH_PCIE_ASPM_CONTROL). Default is **PchPcieAspmAutoConfig**.*
- UINT8 [L1Substates](#)
*The L1 Substates configuration of the root port (see: PCH_PCIE_L1SUBSTATES_CONTROL). Default is **PchPcieL1SubstatesL1_1_2**.*
- UINT8 [LtrEnable](#)
*Latency Tolerance Reporting Mechanism. **0: Disable**; 1: Enable.*
- UINT8 [LtrConfigLock](#)
***0: Disable**; 1: Enable.*
- UINT16 [LtrMaxSnoopLatency](#)
(Test) Latency Tolerance Reporting, Max Snoop Latency.
- UINT16 [LtrMaxNoSnoopLatency](#)
(Test) Latency Tolerance Reporting, Max Non-Snoop Latency.
- UINT8 [SnoopLatencyOverrideMode](#)
(Test) Latency Tolerance Reporting, Snoop Latency Override Mode.
- UINT8 [SnoopLatencyOverrideMultiplier](#)
(Test) Latency Tolerance Reporting, Snoop Latency Override Multiplier.
- UINT16 [SnoopLatencyOverrideValue](#)
(Test) Latency Tolerance Reporting, Snoop Latency Override Value.
- UINT8 [NonSnoopLatencyOverrideMode](#)
(Test) Latency Tolerance Reporting, Non-Snoop Latency Override Mode.
- UINT8 [NonSnoopLatencyOverrideMultiplier](#)
(Test) Latency Tolerance Reporting, Non-Snoop Latency Override Multiplier.
- UINT16 [NonSnoopLatencyOverrideValue](#)
(Test) Latency Tolerance Reporting, Non-Snoop Latency Override Value.
- UINT32 [SlotPowerLimitScale](#): 2
*(Test) Specifies scale used for slot power limit value. Leave as 0 to set to default. Default is **zero**.*
- UINT32 [SlotPowerLimitValue](#): 12
*(Test) Specifies upper limit on power supplied by slot. Leave as 0 to set to default. Default is **zero**.*
- UINT32 [HsioRxSetCtleEnable](#): 1
- UINT32 [HsioRxSetCtle](#): 6
- UINT32 [Uptp](#): 4
(Test) Upstream Port Transmitter Preset used during Gen3 Link Equalization. Used for all lanes. Default is 5.
- UINT32 [Dptp](#): 4

(Test) Downstream Port Transmitter Preset used during Gen3 Link Equalization. Used for all lanes. Default is 7.

- UUINT32 [RsvdBits3](#): 3

Reserved Bits.

- UUINT32 [Rsvd2](#) [16]

Reserved bytes.

11.73.1 Detailed Description

The PCH_PCI_EXPRESS_ROOT_PORT_CONFIG describe the feature and capability of each PCH PCIe root port.

Definition at line 142 of file PchPolicyCommon.h.

11.73.2 Member Data Documentation

11.73.2.1 UUINT32 PCH_PCIE_ROOT_PORT_CONFIG::ClkReqDetect

Probe CLKREQ# signal before enabling CLKREQ# based power management.

Conforming device shall hold CLKREQ# low until CPM is enabled. This feature attempts to verify CLKREQ# signal is connected by testing pad state before enabling CPM. In particular this helps to avoid issues with open-ended PCIe slots. This is only applicable to non hot-plug ports. **0: Disable**; 1: Enable.

Definition at line 164 of file PchPolicyCommon.h.

11.73.2.2 UUINT32 PCH_PCIE_ROOT_PORT_CONFIG::ClkReqNumber

The ClkReq Signal mapped to this root port.

Default is zero. Valid if ClkReqSupported is TRUE. This Number should not exceed the Maximum Available ClkReq Signals for LP and H.

Definition at line 155 of file PchPolicyCommon.h.

11.73.2.3 UUINT32 PCH_PCIE_ROOT_PORT_CONFIG::DeviceResetPad

The PCH pin assigned to device PERST# signal if available, zero otherwise.

This entry is used mainly in Gen3 software equalization flow. It is necessary for some devices (mainly some graphic adapters) to successfully complete the software equalization flow. See also DeviceResetPadActiveHigh

Definition at line 211 of file PchPolicyCommon.h.

11.73.2.4 UUINT8 PCH_PCIE_ROOT_PORT_CONFIG::Gen3EqPh3Method

PCIe Gen3 Equalization Phase 3 Method (see PCH_PCIE_EQ_METHOD).

0: Default; 2: Software Search; 4: Fixed Coefficients

Definition at line 201 of file PchPolicyCommon.h.

11.73.2.5 UUINT32 PCH_PCIE_ROOT_PORT_CONFIG::HsioRxSetCtle

Deprecated , please use HsioRxSetCtle from [PCH_HSIO_PCIE_LANE_CONFIG](#)

Definition at line 232 of file PchPolicyCommon.h.

11.73.2.6 UINT32 PCH_PCIE_ROOT_PORT_CONFIG::HsioRxSetCtleEnable

Deprecated , please use HsioRxSetCtleEnable from [PCH_HSIO_PCIE_LANE_CONFIG](#)

Definition at line 231 of file PchPolicyCommon.h.

11.73.2.7 UINT8 PCH_PCIE_ROOT_PORT_CONFIG::PcieSpeed

Determines each PCIE Port speed capability.

0: Auto; 1: Gen1; 2: Gen2; 3: Gen3 (see: PCH_PCIE_SPEED)

Definition at line 196 of file PchPolicyCommon.h.

11.73.2.8 UINT32 PCH_PCIE_ROOT_PORT_CONFIG::SlotImplemented

Indicates how this root port is connected to endpoint.

0: built-in device; 1: slot Built-in is incompatible with hotplug-capable ports

Definition at line 189 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

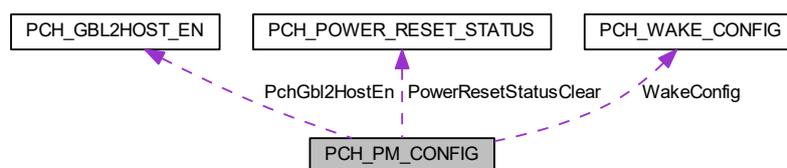
- [PchPolicyCommon.h](#)

11.74 PCH_PM_CONFIG Struct Reference

The [PCH_PM_CONFIG](#) block describes expected miscellaneous power management settings.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_PM_CONFIG:



Public Attributes

- [PCH_POWER_RESET_STATUS](#) **PowerResetStatusClear**
Specify which Power/Reset bits need to be cleared by the PCH Init Driver.
- [PCH_WAKE_CONFIG](#) **WakeConfig**
Specify Wake Policy.
- [PCH_DEEP_SX_CONFIG](#) **PchDeepSxPol**
Deep Sx Policy. Default is **PchDeepSxPolDisable**.
- [PCH_SLP_S3_MIN_ASSERT](#) **PchSlpS3MinAssert**
SLP_S3 Minimum Assertion Width Policy. Default is **PchSlpS350ms**.

- [PCH_SLP_S4_MIN_ASSERT](#) [PchSlpS4MinAssert](#)
*SLP_S4 Minimum Assertion Width Policy. Default is **PchSlpS44s**.*
- [PCH_SLP_SUS_MIN_ASSERT](#) [PchSlpSusMinAssert](#)
*SLP_SUS Minimum Assertion Width Policy. Default is **PchSlpSus4s**.*
- [PCH_SLP_A_MIN_ASSERT](#) [PchSlpAMinAssert](#)
*SLP_A Minimum Assertion Width Policy. Default is **PchSlpA2s**.*
- UUINT32 [PciClockRun](#): 1
This member describes whether or not the PCI ClockRun feature of PCH should be enabled.
- UUINT32 [SlpStrchSusUp](#): 1
0: Disable; 1: Enable SLP_X Stretching After SUS Well Power Up
- UUINT32 [SlpLanLowDc](#): 1
Enable/Disable SLP_LAN# Low on DC Power.
- UUINT32 [PwrBtnOverridePeriod](#): 3
PCH power button override period.
- UUINT32 [DisableEnergyReport](#): 1
(Test) Disable/Enable PCH to CPU enery report feature.
- UUINT32 [DisableDsxAcPresentPullDown](#): 1
When set to Disable, PCH will internal pull down AC_PRESENT in deep SX and during G3 exit.
- UUINT32 [PmcReadDisable](#): 1
(Test) When set to true, this bit disallows Host reads to PMC XRAM.
- UUINT32 [CapsuleResetType](#): 1
This determines the type of reset issued during the capsule update process by UpdateCapsule().
- UUINT32 [DisableNativePowerButton](#): 1
Power button native mode disable.
- UUINT32 [SlpS0Enable](#): 1
Indicates whether SLP_S0# is to be asserted when PCH reaches idle state.
- UUINT32 [DirtyWarmReset](#): 1
DirtyWarmReset enable.
- UUINT32 [StallDirtyWarmReset](#): 1
Stall during DWR.
- UUINT32 [GrPfetDurOnDef](#): 2
Global Reset PFET duration.
- UUINT32 [Dwr_MeResetPrepDone](#): 1
ME Reset Prep Done.
- UUINT32 [Dwr_IeResetPrepDone](#): 1
IE Reset Prep Done.
- UUINT32 [Dwr_BmcRootPort](#): 8
Root port where BMC is connected to.
- UUINT32 [RsvdBits0](#): 6
- UUINT8 [PchPwrCycDur](#)
Reset Power Cycle Duration could be customized in the unit of second.
- UUINT8 [PciePIISsc](#)
Specifies the Pcie PII Spread Spectrum Percentage The value of this policy is in 1/10th percent units.
- UUINT8 [Rsvd0](#) [2]
Reserved bytes.

11.74.1 Detailed Description

The [PCH_PM_CONFIG](#) block describes expected miscellaneous power management settings.

The PowerResetStatusClear field would clear the Power/Reset status bits, please set the bits if you want PCH Init driver to clear it, if you want to check the status later then clear the bits.

Definition at line 1226 of file PchPolicyCommon.h.

11.74.2 Member Data Documentation

11.74.2.1 UINT32 PCH_PM_CONFIG::CapsuleResetType

This determines the type of reset issued during the capsule update process by UpdateCapsule().

The default is **0:S3 Resume**, 1:Warm reset.

Definition at line 1290 of file PchPolicyCommon.h.

11.74.2.2 UINT32 PCH_PM_CONFIG::DisableDsxAcPresentPulldown

When set to Disable, PCH will internal pull down AC_PRESENT in deep SX and during G3 exit.

When set to Enable, PCH will not pull down AC_PRESENT. This setting is ignored when DeepSx is not supported.

Default is **0:Disable**

Definition at line 1278 of file PchPolicyCommon.h.

11.74.2.3 UINT32 PCH_PM_CONFIG::DisableEnergyReport

(Test) Disable/Enable PCH to CPU enery report feature.

0: Disable; 1: Enable. Enery Report is must have feature. Wihtout Energy Report, the performance report by workloads/benchmarks will be unrealistic because PCH's energy is not being accounted in power/performance management algorithm. If for some reason PCH energy report is too high, which forces CPU to try to reduce its power by throttling, then it could try to disable Energy Report to do first debug. This might be due to energy scaling factors are not correct or the LPM settings are not kicking in.

Definition at line 1271 of file PchPolicyCommon.h.

11.74.2.4 UINT32 PCH_PM_CONFIG::DisableNativePowerButton

Power button native mode disable.

While FALSE, the PMC's power button logic will act upon the input value from the GPIO unit, as normal. While TRUE, this will result in the PMC logic constantly seeing the power button as de-asserted. **Default is FALSE.**

Definition at line 1297 of file PchPolicyCommon.h.

11.74.2.5 UINT8 PCH_PM_CONFIG::PchPwrCycDur

Reset Power Cycle Duration could be customized in the unit of second.

Please refer to EDS for all support settings. PCH HW default is 4 seconds, and range is 1~4 seconds, where **0 is default**, 1 is 1 second, 2 is 2 seconds, ... 4 is 4 seconds. And make sure the setting correct, which never less than the following register.

- GEN_PMCON_B.SLP_S3_MIN_ASST_WDTH
- GEN_PMCON_B.SLP_S4_MIN_ASST_WDTH
- PWRM_CFG.SLP_A_MIN_ASST_WDTH
- PWRM_CFG.SLP_LAN_MIN_ASST_WDTH

Definition at line 1327 of file PchPolicyCommon.h.

11.74.2.6 UUINT32 PCH_PM_CONFIG::PciClockRun

This member describes whether or not the PCI ClockRun feature of PCH should be enabled.

0: Disable; 1: Enable

Definition at line 1244 of file PchPolicyCommon.h.

11.74.2.7 UUINT8 PCH_PM_CONFIG::PciePIISsc

Specifies the Pcie PII Spread Spectrum Percentage The value of this policy is in 1/10th percent units.

Valid spread range is 0-20. A value of 0xFF is reserved for AUTO. A value of 0 is SSC of 0.0%. A value of 20 is SSC of 2.0% The default is **0xFF: AUTO - No BIOS override**.

Definition at line 1335 of file PchPolicyCommon.h.

11.74.2.8 UUINT32 PCH_PM_CONFIG::PmcReadDisable

(Test) When set to true, this bit disallows Host reads to PMC XRAM.

BIOS must set this bit (to disable and lock the feature) prior to passing control to OS **0:Disable, 1:Enable**

Definition at line 1285 of file PchPolicyCommon.h.

11.74.2.9 PCH_POWER_RESET_STATUS PCH_PM_CONFIG::PowerResetStatusClear

Specify which Power/Reset bits need to be cleared by the PCH Init Driver.

Usually platform drivers take care of these bits, but if not, let PCH Init driver clear the bits.

Definition at line 1233 of file PchPolicyCommon.h.

11.74.2.10 UUINT32 PCH_PM_CONFIG::PwrBtnOverridePeriod

PCH power button override period.

000b-4s, 001b-6s, 010b-8s, 011b-10s, 100b-12s, 101b-14s **Default is 0: 4s**

Definition at line 1259 of file PchPolicyCommon.h.

11.74.2.11 UUINT32 PCH_PM_CONFIG::RsvdBits0

Todo ADD DESCRIPTION

Definition at line 1314 of file PchPolicyCommon.h.

11.74.2.12 UUINT32 PCH_PM_CONFIG::SlpLanLowDc

Enable/Disable SLP_LAN# Low on DC Power.

0: Disable; 1: Enable. Configure On DC PHY Power Diable according to policy SlpLanLowDc. When this is enabled, SLP_LAN# will be driven low when ACPRESENT is low. This indicates that LAN PHY should be powered off on battery mode. This will override the DC_PP_DIS setting by WolEnableOverride.

Definition at line 1253 of file PchPolicyCommon.h.

11.74.2.13 UINT32 PCH_PM_CONFIG::SlpS0Enable

Indicates whether SLP_S0# is to be asserted when PCH reaches idle state.

When set to one SLP_S0# will be asserted in idle state. When set to zero SLP_S0# will not toggle and is always driven high. 0:Disable, **1:Enable**

Warning: In SKL PCH VCCPRIM_CORE must NOT be reduced based on SLP_S0# being asserted. If a platform is using SLP_S0 to lower PCH voltage the below policy must be disabled.

Definition at line 1307 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.75 PCH_PORT61H_SMM_CONFIG Struct Reference

This structure is used for the emulation feature for Port61h read.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 1
*0: Disable; 1: **Enable** the emulation*
- UINT32 [RsvdBits0](#): 31
Reserved bits.

11.75.1 Detailed Description

This structure is used for the emulation feature for Port61h read.

The port is trapped and the SMI handler will toggle bit4 according to the handler's internal state.

Definition at line 1396 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.76 PCH_POWER_RESET_STATUS Struct Reference

This [PCH_POWER_RESET_STATUS](#) Specifies which Power/Reset bits need to be cleared by the PCH Init Driver.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [MeWakeSts](#): 1
*Clear the ME_WAKE_STS bit in the Power and Reset Status (PRSTS) register. 0: Disable; 1: **Enable**.*
- UINT32 [MeHrstColdSts](#): 1
*Clear the ME_HRST_COLD_STS bit in the Power and Reset Status (PRSTS) register. 0: Disable; 1: **Enable**.*
- UINT32 [MeHrstWarmSts](#): 1
*Clear the ME_HRST_WARM_STS bit in the Power and Reset Status (PRSTS) register. 0: Disable; 1: **Enable**.*

- UUINT32 [MeHostPowerDn](#): 1

Clear the `ME_HOST_PWRDN` bit in the Power and Reset Status (`PRSTS`) register. **0: Disable**; **1: Enable**.

- UUINT32 [WolOvrWkSts](#): 1

Clear the `WOL_OVR_WK_STS` bit in the Power and Reset Status (`PRSTS`) register. **0: Disable**; **1: Enable**.

11.76.1 Detailed Description

This [PCH_POWER_RESET_STATUS](#) Specifies which Power/Reset bits need to be cleared by the PCH Init Driver. Usually platform drivers take care of these bits, but if not, let PCH Init driver clear the bits.

Definition at line 1116 of file `PchPolicyCommon.h`.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.77 PCH_RST_PCIE_STORAGE_CONFIG Struct Reference

This structure describes the details of Intel RST for PCIe Storage remapping Note: In order to use this feature, Intel RST Driver is required.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [Enable](#): 1

This member describes whether or not the Intel RST for PCIe Storage remapping should be enabled.

- UUINT32 [RstPcieStoragePort](#): 5

*Intel RST for PCIe Storage remapping - PCIe Port Selection (1-based, **0 = autodetect**) The supported ports for PCIe Storage remapping is different depend on the platform and cycle router, the assignments are as below: SKL PCH-LP RST PCIe Storage Cycle Router Assignment: i.) RST PCIe Storage Cycle Router 2 -> RP5 - RP8 ii.) RST PCIe Storage Cycle Router 3 -> RP9 - RP12.*

- UUINT32 [RsvdBits0](#): 2

Reserved bit.

- UUINT32 [DeviceResetDelay](#): 8

PCIe Storage Device Reset Delay in milliseconds (ms), which it guarantees such delay gap is fulfilled before PCIe Storage Device configuration space is accessed after an reset caused by the link disable and enable step.

- UUINT32 [RsvdBits1](#): 16

Reserved bits.

- UUINT32 [Rsvd0](#) [2]

Reserved bytes.

11.77.1 Detailed Description

This structure describes the details of Intel RST for PCIe Storage remapping Note: In order to use this feature, Intel RST Driver is required.

Definition at line 548 of file `PchPolicyCommon.h`.

11.77.2 Member Data Documentation

11.77.2.1 UINT32 PCH_RST_PCIE_STORAGE_CONFIG::DeviceResetDelay

PCIe Storage Device Reset Delay in milliseconds (ms), which it guarantees such delay gap is fulfilled before PCIe Storage Device configuration space is accessed after a reset caused by the link disable and enable step.

Default value is **100ms**.

Definition at line 574 of file PchPolicyCommon.h.

11.77.2.2 UINT32 PCH_RST_PCIE_STORAGE_CONFIG::Enable

This member describes whether or not the Intel RST for PCIe Storage remapping should be enabled.

0: Disable; 1: Enable. Note 1: If Sata Controller is disabled, PCIe Storage Remapping should be disabled as well Note 2: If PCIe Storage remapping is enabled, the PCH integrated AHCI controllers Class Code is configured as RAID

Definition at line 554 of file PchPolicyCommon.h.

11.77.2.3 UINT32 PCH_RST_PCIE_STORAGE_CONFIG::RstPcieStoragePort

Intel RST for PCIe Storage remapping - PCIe Port Selection (1-based, **0 = autodetect**) The supported ports for PCIe Storage remapping is different depend on the platform and cycle router, the assignments are as below: SKL PCH-LP RST PCIe Storage Cycle Router Assignment: i.) RST PCIe Storage Cycle Router 2 -> RP5 - RP8 ii.) RST PCIe Storage Cycle Router 3 -> RP9 - RP12.

SKL PCH-H RST PCIe Storage Cycle Router Assignment: i.) RST PCIe Storage Cycle Router 1 -> RP9 - RP12 ii.) RST PCIe Storage Cycle Router 2 -> RP13 - RP16 iii.) RST PCIe Storage Cycle Router 3 -> RP17 - RP20

Definition at line 567 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

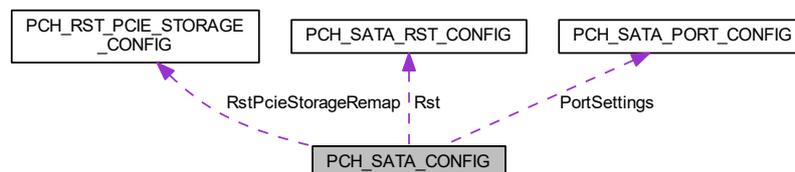
- [PchPolicyCommon.h](#)

11.78 PCH_SATA_CONFIG Struct Reference

The [PCH_SATA_CONFIG](#) block describes the expected configuration of the SATA controllers.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_SATA_CONFIG:



Public Attributes

- UINT32 [Enable](#): 1

This member describes whether or not the SATA controllers should be enabled.

- UUINT32 [TestMode](#): 1
(Test) 0: Disable; 1: Allow entrance to the PCH SATA test modes
- UUINT32 [SalpSupport](#): 1
0: Disable; 1: Enable Aggressive Link Power Management
- UUINT32 [PwrOptEnable](#): 1
0: Disable; 1: Enable SATA Power Optimizer on PCH side.
- UUINT32 [eSATASpeedLimit](#): 1
eSATASpeedLimit When enabled, BIOS will configure the PxSCTL.SPD to 2 to limit the eSATA port speed.
- UUINT32 [EnclosureSupport](#): 1
0: Disable; 1: Enable Enclosure Management Support
- UUINT32 [Rsvdbits](#): 26
Reserved bits.
- PCH_SATA_MODE [SataMode](#)
Determines the system will be configured to which SATA mode (PCH_SATA_MODE).
- PCH_SATA_SPEED [SpeedLimit](#)
Indicates the maximum speed the SATA controller can support 0h: PchSataSpeedDefault; 1h: 1.5 Gb/s (Gen 1); 2h: 3 Gb/s(Gen 2); 3h: 6 Gb/s (Gen 1)
- PCH_SATA_PORT_CONFIG [PortSettings](#) [PCH_MAX_SATA_PORTS]
This member configures the features, property, and capability for each SATA port.
- PCH_SATA_RST_CONFIG [Rst](#)
Setting applicable to Rapid Storage Technology.
- PCH_RST_PCIE_STORAGE_CONFIG [RstPcieStorageRemap](#) [PCH_MAX_RST_PCIE_STORAGE_CR]
This member describes the details of implementation of Intel RST for PCIe Storage remapping (Intel RST Driver is required)
- UUINT32 [Rsvd0](#) [4]
Reserved fields for future expansion.

11.78.1 Detailed Description

The [PCH_SATA_CONFIG](#) block describes the expected configuration of the SATA controllers.

Definition at line 583 of file PchPolicyCommon.h.

11.78.2 Member Data Documentation

11.78.2.1 UUINT32 PCH_SATA_CONFIG::Enable

This member describes whether or not the SATA controllers should be enabled.

0: Disable; 1: **Enable**.

Definition at line 587 of file PchPolicyCommon.h.

11.78.2.2 UUINT32 PCH_SATA_CONFIG::eSATASpeedLimit

eSATASpeedLimit When enabled, BIOS will configure the PxSCTL.SPD to 2 to limit the eSATA port speed.

Please be noted, this setting could be cleared by HBA reset, which might be issued by EFI AHCI driver when POST time, or by SATA inbox driver/RST driver after POST. To support the Speed Limitation when POST, the EFI AHCI driver should preserve the setting before and after initialization. For support it after POST, it's dependent on driver's behavior. **0: Disable**; 1: Enable

Definition at line 601 of file PchPolicyCommon.h.

11.78.2.3 PCH_SATA_MODE PCH_SATA_CONFIG::SataMode

Determines the system will be configured to which SATA mode (PCH_SATA_MODE).

Default is **PchSataModeAhci**.

Definition at line 608 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.79 PCH_SATA_PORT_CONFIG Struct Reference

This structure configures the features, property, and capability for each SATA port.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [Enable](#): 1
Enable SATA port.
- UUINT32 [HotPlug](#): 1
0: Disable; 1: Enable
- UUINT32 [InterlockSw](#): 1
0: Disable; 1: Enable
- UUINT32 [External](#): 1
0: Disable; 1: Enable
- UUINT32 [SpinUp](#): 1
0: Disable; 1: Enable the COMRESET initialization Sequence to the device
- UUINT32 [SolidStateDrive](#): 1
0: HDD; 1: SSD
- UUINT32 [DevSlp](#): 1
0: Disable; 1: Enable DEVSLP on the port
- UUINT32 [EnableDitoConfig](#): 1
0: Disable; 1: Enable DEVSLP Idle Timeout settings (DmVal, DitoVal)
- UUINT32 [DmVal](#): 4
DITO multiplier. Default is 15.
- UUINT32 [DitoVal](#): 10
DEVSLP Idle Timeout (DITO), Default is 625.
- UUINT32 [ZpOdd](#): 1
Support zero power ODD 0: Disable, 1: Enable.
- UUINT32 [RsvdBits0](#): 9
Reserved fields for future expansion w/o protocol change.
- UUINT32 [HsioRxEqBoostMagAdEnable](#): 1
- UUINT32 [HsioRxEqBoostMagAd](#): 6
- UUINT32 [HsioTxGen1DownscaleAmpEnable](#): 1
- UUINT32 [HsioTxGen1DownscaleAmp](#): 6
- UUINT32 [HsioTxGen2DownscaleAmpEnable](#): 1
- UUINT32 [HsioTxGen2DownscaleAmp](#): 6
- UUINT32 [Rsvd0](#): 11
Reserved bits.

11.79.1 Detailed Description

This structure configures the features, property, and capability for each SATA port.

Definition at line 491 of file PchPolicyCommon.h.

11.79.2 Member Data Documentation

11.79.2.1 UINT32 PCH_SATA_PORT_CONFIG::Enable

Enable SATA port.

It is highly recommended to disable unused ports for power savings0: Disable; **1: Enable**

Definition at line 496 of file PchPolicyCommon.h.

11.79.2.2 UINT32 PCH_SATA_PORT_CONFIG::HsioRxEqBoostMagAd

Deprecated , please use HsioRxGen3EqBoostMag

Definition at line 514 of file PchPolicyCommon.h.

11.79.2.3 UINT32 PCH_SATA_PORT_CONFIG::HsioRxEqBoostMagAdEnable

Deprecated , please use HsioRxGen3EqBoostMagEnable

Definition at line 513 of file PchPolicyCommon.h.

11.79.2.4 UINT32 PCH_SATA_PORT_CONFIG::HsioTxGen1DownscaleAmp

Deprecated , please use HsioTxGen1DownscaleAmp in [PCH_HSIO_SATA_PORT_LANE](#)

Definition at line 517 of file PchPolicyCommon.h.

11.79.2.5 UINT32 PCH_SATA_PORT_CONFIG::HsioTxGen1DownscaleAmpEnable

Deprecated , please use HsioTxGen1DownscaleAmpEnable in [PCH_HSIO_SATA_PORT_LANE](#)

Definition at line 516 of file PchPolicyCommon.h.

11.79.2.6 UINT32 PCH_SATA_PORT_CONFIG::HsioTxGen2DownscaleAmp

Deprecated , please use HsioTxGen2DownscaleAmp in [PCH_HSIO_SATA_PORT_LANE](#)

Definition at line 519 of file PchPolicyCommon.h.

11.79.2.7 UINT32 PCH_SATA_PORT_CONFIG::HsioTxGen2DownscaleAmpEnable

Deprecated , please use HsioTxGen2DownscaleAmpEnable in [PCH_HSIO_SATA_PORT_LANE](#)

Definition at line 518 of file PchPolicyCommon.h.

11.79.2.8 UUINT32 PCH_SATA_PORT_CONFIG::ZpOdd

Support zero power ODD **0: Disable**, 1: Enable.

This is also used to disable ModPHY dynamic power gate.

Definition at line 510 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.80 PCH_SATA_RST_CONFIG Struct Reference

Rapid Storage Technology settings.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [RaidAlternateld](#): 1
0: Disable; 1: Enable RAID Alternate ID
- UUINT32 [Raid0](#): 1
0: Disable; 1: Enable RAID0
- UUINT32 [Raid1](#): 1
0: Disable; 1: Enable RAID1
- UUINT32 [Raid10](#): 1
0: Disable; 1: Enable RAID10
- UUINT32 [Raid5](#): 1
0: Disable; 1: Enable RAID5
- UUINT32 [Irrt](#): 1
0: Disable; 1: Enable Intel Rapid Recovery Technology
- UUINT32 [OromUiBanner](#): 1
0: Disable; 1: Enable OROM UI and BANNER
- UUINT32 [OromUiDelay](#): 2
00b: 2 secs; 01b: 4 secs; 10b: 6 secs; 11: 8 secs (see: PCH_SATA_OROM_DELAY)
- UUINT32 [HddUnlock](#): 1
0: Disable; 1: Enable. Indicates that the HDD password unlock in the OS is enabled
- UUINT32 [LedLocate](#): 1
0: Disable; 1: Enable. Indicates that the LED/SGPIO hardware is attached and ping to locate feature is enabled on the OS
- UUINT32 [IrrtOnly](#): 1
0: Disable; 1: Enable. Allow only IRRRT drives to span internal and external ports
- UUINT32 [SmartStorage](#): 1
0: Disable; 1: Enable RST Smart Storage caching Bit
- UUINT32 [EfiRaidDriverLoad](#):1
0: Dont load EFI RST/RSTe driver; 1: Load EFI RST/RSTe driver
- UUINT32 [Resvdbits](#): 18
Reserved Bits.

11.80.1 Detailed Description

Rapid Storage Technology settings.

Definition at line 527 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.81 PCH_SKYCAM_CIO2_FLS_CONFIG Struct Reference

The [PCH_SKYCAM_CIO2_FLS_CONFIG](#) block describes SkyCam CIO2 FLS registers configuration.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [PortATrimEnable](#): 1
0: Disable; 1: Enable - Enable Port A Clk Trim
- UINT32 [PortBTrimEnable](#): 1
0: Disable; 1: Enable - Enable Port B Clk Trim
- UINT32 [PortCTrimEnable](#): 1
0: Disable; 1: Enable - Enable Port C Clk Trim
- UINT32 [PortDTrimEnable](#): 1
0: Disable; 1: Enable - Enable Port D Clk Trim
- UINT32 [PortACtleEnable](#): 1
0: Disable; 1: Enable - Enable Port A Ctle
- UINT32 [PortBCtleEnable](#): 1
0: Disable; 1: Enable - Enable Port B Ctle
- UINT32 [PortCDCtleEnable](#): 1
0: Disable; 1: Enable - Enable Port C/D Ctle
- UINT32 [PortBCtleCapValue](#): 4
Port A Ctle Cap Value.
- UINT32 [PortCDCtleCapValue](#): 4
Port B Ctle Cap Value.
- UINT32 [PortACtleResValue](#): 5
Port C/D Ctle Cap Value.
- UINT32 [PortBCtleResValue](#): 5
Port A Ctle Res Value.
- UINT32 [PortCDCtleResValue](#): 5
Port B Ctle Res Value.
- UINT32 [RsvdBits1](#): 5
Port C/D Ctle Res Value.
- UINT32 [PortBCIkTrimValue](#): 4
Port A Clk Trim Value.
- UINT32 [PortCCIkTrimValue](#): 4
Port B Clk Trim Value.
- UINT32 [PortDCIkTrimValue](#): 4
Port C Clk Trim Value.
- UINT32 [PortADDataTrimValue](#): 16
Port D Clk Trim Value.

- UINT32 [PortBDataTrimValue](#): 16
Port A Data Trim Value.
- UINT32 [PortCDDataTrimValue](#): 16
Port B Data Trim Value.

11.81.1 Detailed Description

The [PCH_SKYCAM_CIO2_FLS_CONFIG](#) block describes SkyCam CIO2 FLS registers configuration.

Definition at line 1460 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.82 PCH_SMBUS_CONFIG Struct Reference

The SMBUS_CONFIG block lists the reserved addresses for non-ARP capable devices in the platform.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 1
This member describes whether or not the SMBus controller of PCH should be enabled.
- UINT32 [ArpEnable](#): 1
*Enable SMBus ARP support, **0: Disable**; 1: Enable.*
- UINT32 [DynamicPowerGating](#): 1
*(Test) **Disable** or Enable Smbus dynamic power gating.*
- UINT32 [RsvdBits0](#): 29
Reserved bits.
- UINT16 [SmbusIoBase](#)
*SMBUS Base Address (IO space). Default is **0xEFA0**.*
- UINT8 [Rsvd0](#)
Reserved bytes.
- UINT8 [NumRsvdSmbusAddresses](#)
The number of elements in the RsvdSmbusAddressTable.
- UINT8 [RsvdSmbusAddressTable](#) [PCH_MAX_SMBUS_RESERVED_ADDRESS]
Array of addresses reserved for non-ARP-capable SMBus devices.

11.82.1 Detailed Description

The SMBUS_CONFIG block lists the reserved addresses for non-ARP capable devices in the platform.

Definition at line 866 of file PchPolicyCommon.h.

11.82.2 Member Data Documentation

11.82.2.1 UINT32 PCH_SMBUS_CONFIG::Enable

This member describes whether or not the SMBus controller of PCH should be enabled.

0: Disable; 1: **Enable**.

Definition at line 871 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.83 PCH_SPI_CONFIG Struct Reference

This structure contains the policies which are related to SPI.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [ShowSpiController](#): 1

Force to show SPI controller.

- UINT32 [RsvdBits](#): 31

Reserved bits.

11.83.1 Detailed Description

This structure contains the policies which are related to SPI.

Definition at line 1784 of file PchPolicyCommon.h.

11.83.2 Member Data Documentation

11.83.2.1 UINT32 PCH_SPI_CONFIG::ShowSpiController

Force to show SPI controller.

0: FALSE, 1: TRUE NOTE: For Windows OS, it MUST BE false. It's optional for other OSs.

Definition at line 1790 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

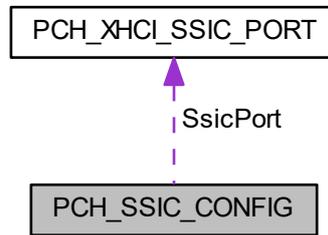
- [PchPolicyCommon.h](#)

11.84 PCH_SSI_CONFIG Struct Reference

These members describe some settings which are related to the SSIC ports.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_SSIC_CONFIG:



11.84.1 Detailed Description

These members describe some settings which are related to the SSIC ports.

Definition at line 1576 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

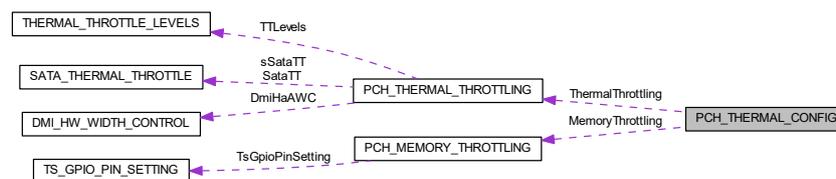
- [PchPolicyCommon.h](#)

11.85 PCH_THERMAL_CONFIG Struct Reference

The [PCH_THERMAL_CONFIG](#) block describes the expected configuration of the PCH for Thermal.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_THERMAL_CONFIG:



Public Attributes

- UUINT32 [ThermalDeviceEnable](#): 2
This field reports the status of Thermal Device.
- UUINT32 [TsmicLock](#): 1
*This locks down "SMI Enable on Alert Thermal Sensor Trip". 0: Disabled, 1: **Enabled**.*
- [PCH_THERMAL_THROTTLING](#) [ThermalThrottling](#)
This field decides the settings of Thermal throttling.
- [PCH_MEMORY_THROTTLING](#) [MemoryThrottling](#)

Memory Thermal Management settings.

- UINT16 [PchHotLevel](#)

*This field decides the temperature, default is **zero**.*

11.85.1 Detailed Description

The [PCH_THERMAL_CONFIG](#) block describes the expected configuration of the PCH for Thermal.

Definition at line 1075 of file PchPolicyCommon.h.

11.85.2 Member Data Documentation

11.85.2.1 UINT16 PCH_THERMAL_CONFIG::PchHotLevel

This field decides the temperature, default is **zero**.

- 0x00 is the hottest
- 0x1FF is the lowest temperature

Definition at line 1098 of file PchPolicyCommon.h.

11.85.2.2 UINT32 PCH_THERMAL_CONFIG::ThermalDeviceEnable

This field reports the status of Thermal Device.

When it reports ThermalDevice is disabled, the PCI configuration space of thermal device will be hidden by setting TCFD and PCR[PSF2] TRH PCIEN[8] prior to end of POST.0: Disabled, **1: Enabled in PCI mode**, 2: Enabled in ACPI mode

Definition at line 1081 of file PchPolicyCommon.h.

11.85.2.3 PCH_THERMAL_THROTTLING PCH_THERMAL_CONFIG::ThermalThrottling

This field decides the settings of Thermal throttling.

When the Suggested Setting is enabled, PCH RC will use the suggested representative values.

Definition at line 1088 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

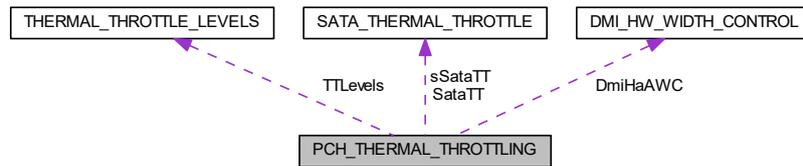
- [PchPolicyCommon.h](#)

11.86 PCH_THERMAL_THROTTLING Struct Reference

This structure decides the settings of PCH Thermal throttling.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_THERMAL_THROTTLING:



11.86.1 Detailed Description

This structure decides the settings of PCH Thermal throttling.

When the Suggested Setting is enabled, PCH RC will use the suggested representative values.

Definition at line 1023 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.87 PCH_TRACE_HUB_CONFIG Struct Reference

The [PCH_TRACE_HUB_CONFIG](#) block describes TraceHub settings for PCH.

```
#include <PchPolicyCommon.h>
```

11.87.1 Detailed Description

The [PCH_TRACE_HUB_CONFIG](#) block describes TraceHub settings for PCH.

Definition at line 1446 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.88 PCH_USB20_PORT_CONFIG Struct Reference

This structure configures per USB2 port physical settings.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [Enable](#): 1
0: Disable; 1: Enable.
- UUINT32 [RsvdBits0](#): 31
Reserved bits.
- UUINT8 [OverCurrentPin](#)
These members describe the specific over current pin number of USB 2.0 Port N.

- UINT8 [Rsvd0](#) [3]
Reserved bytes, align to multiple 4.
- USB2_PHY_PARAMETERS [Afe](#)
USB2 AFE settings.
- UINT32 [Rsvd1](#) [1]
Reserved bytes.

11.88.1 Detailed Description

This structure configures per USB2 port physical settings.

It allows to setup the port location and port length, and configures the port strength accordingly.

Definition at line 1510 of file PchPolicyCommon.h.

11.88.2 Member Data Documentation

11.88.2.1 UINT8 PCH_USB20_PORT_CONFIG::OverCurrentPin

These members describe the specific over current pin number of USB 2.0 Port N.

It is SW's responsibility to ensure that a given port's bit map is set only for one OC pin Description. USB2 and USB3 on the same combo Port must use the same OC pin (see: [USB_OVERCURRENT_PIN](#)).

Definition at line 1519 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.89 PCH_USB30_PORT_CONFIG Struct Reference

This structure describes whether the USB3 Port N of PCH is enabled by platform modules.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 1
*0: Disable; 1: **Enable**.*
- UINT32 [RsvdBits0](#): 31
Reserved bits.
- UINT8 [OverCurrentPin](#)
These members describe the specific over current pin number of USB 3.0 Port N.
- UINT8 [Rsvd0](#) [3]
Reserved bytes, align to multiple 4.
- UINT32 [HsioTxDeEmphEnable](#): 1
*Enable the write to USB 3.0 TX Output -3.5dB De-Emphasis Adjustment, **0: Disable**; 1: Enable.*
- UINT32 [HsioTxDeEmph](#): 6
*USB 3.0 TX Output -3.5dB De-Emphasis Adjustment Setting (ow2tapgen2deemph3p5) HSIO_TX_DWORD5[21:16]
Default = 29h (approximately -3.5dB De-Emphasis)*
- UINT32 [HsioTxDownscaleAmpEnable](#): 1
*Enable the write to USB 3.0 TX Output Downscale Amplitude Adjustment, **0: Disable**; 1: Enable.*
- UINT32 [HsioTxDownscaleAmp](#): 6

USB 3.0 TX Output Downscale Amplitude Adjustment (orate01margin) HSIO_TX_DWORD8[21:16] **Default = 00h**

- UINT32 [RsvdBits1](#): 18
Reserved bits.
- UINT32 [Rsvd1](#) [1]
Reserved bytes.

11.89.1 Detailed Description

This structure describes whether the USB3 Port N of PCH is enabled by platform modules.

Definition at line 1528 of file PchPolicyCommon.h.

11.89.2 Member Data Documentation

11.89.2.1 UINT8 PCH_USB30_PORT_CONFIG::OverCurrentPin

These members describe the specific over current pin number of USB 3.0 Port N.

It is SW's responsibility to ensure that a given port's bit map is set only for one OC pin Description. USB2 and USB3 on the same combo Port must use the same OC pin (see: [USB_OVERCURRENT_PIN](#)).

Definition at line 1537 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

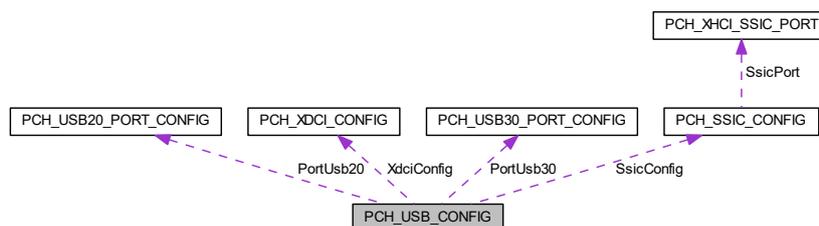
- [PchPolicyCommon.h](#)

11.90 PCH_USB_CONFIG Struct Reference

This member describes the expected configuration of the PCH USB controllers, Platform modules may need to refer Setup options, schematic, BIOS specification to update this field.

```
#include <PchPolicyCommon.h>
```

Collaboration diagram for PCH_USB_CONFIG:



Public Attributes

- UINT32 [UsbPrecondition](#): 1
This feature intends to reduce the necessary initialization time for USB HC and devices on root ports.
- UINT32 [DisableComplianceMode](#): 1
This policy will disable XHCI compliance mode on all ports.
- UINT32 [XhciOcMapEnabled](#): 1

0: To disable OC mapping for USB XHCI ports 1: Set Xhci OC registers, Set Xhci OCCDone bit, XHCI Access Control Bit.

- UUINT32 [XhciWakeOnUsb](#): 1

0: To disable Wake on USB connect/Disconnect 1: Enables Wake on USB connect/disconnect event.

- UUINT32 [RsvdBits0](#): 27

Reserved bits.

- [PCH_USB20_PORT_CONFIG PortUsb20](#) [PCH_MAX_USB2_PORTS]

These members describe whether the USB2 Port N of PCH is enabled by platform modules.

- [PCH_USB30_PORT_CONFIG PortUsb30](#) [PCH_MAX_USB3_PORTS]

These members describe whether the USB3 Port N of PCH is enabled by platform modules.

- [PCH_XDCI_CONFIG XdciParam](#)

This member describes whether or not the xDCI controller should be enabled.

- [PCH_SSIC_CONFIG SsicParam](#)

These members describe some settings which are related to the SSIC ports.

- UUINT32 [Rsvd0](#) [6]

Reserved bytes.

11.90.1 Detailed Description

This member describes the expected configuration of the PCH USB controllers, Platform modules may need to refer Setup options, schematic, BIOS specification to update this field.

The `Usb20OverCurrentPins` and `Usb30OverCurrentPins` field must be updated by referring the schematic.

Definition at line 1600 of file `PchPolicyCommon.h`.

11.90.2 Member Data Documentation

11.90.2.1 UUINT32 `PCH_USB_CONFIG::DisableComplianceMode`

This policy will disable XHCI compliance mode on all ports.

Compliance Mode should be default enabled. For the platform that support USB Type-C, it can disable Compliance Mode, and enable Compliance Mode when testing. **0:Disable** , 1: Enable

Definition at line 1626 of file `PchPolicyCommon.h`.

11.90.2.2 `PCH_USB20_PORT_CONFIG PCH_USB_CONFIG::PortUsb20`[PCH_MAX_USB2_PORTS]

These members describe whether the USB2 Port N of PCH is enabled by platform modules.

Panel and Dock are used to describe the layout of USB port. Panel is only available for Desktop PCH. Dock is only available for Mobile LPT.

Definition at line 1638 of file `PchPolicyCommon.h`.

11.90.2.3 UUINT32 `PCH_USB_CONFIG::UsbPrecondition`

This feature intends to reduce the necessary initialization time for USB HC and devices on root ports.

It is assembled by PCHInit drivers in PEI and DXE phase. In PEI phase, the feature resets all USB HCs on PCH bus, including Intel EHCI and XHCI. After reset USB HC, continue the system initialization without waiting for the USB XHC reset ready. After running to DXE phase, the feature resets those USB devices installed on each USB HC root port in parallel, including any non USB3 speed devices on XHCI root port if XHCI is enabled. For USB3 protocol root port, USB3 speed devices will be advanced to enable state if link training succeeds after XHC reset.

UsbPrecondition = Enable , Force USB Init happen in PEI as part of 2Sec Fast Boot bios optimization. Usb↔Precondition = Disable, USB Init happen in DXE just like traditionally where it happen. Remark: With Precondition Enabled some USB2 devices which are not compliant with usb2 specification are not being detected if installed in the system during S4/S5.

0: Disable; 1: Enable.

Definition at line 1620 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.91 PCH_WAKE_CONFIG Struct Reference

This structure allows to customize PCH wake up capability from S5 or DeepSx by WOL, LAN, PCIE wake events.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [PmeB0S5Dis](#): 1
Corresponds to the PME_B0_S5_DIS bit in the General PM Configuration B (GEN_PMCON_B) register.
- UUINT32 [WolEnableOverride](#): 1
Corresponds to the "WOL Enable Override" bit in the General PM Configuration B (GEN_PMCON_B) register. 0: Disable; 1: Enable.
- UUINT32 [Gp27WakeFromDeepSx](#): 1
- UUINT32 [PcieWakeFromDeepSx](#): 1
Determine if enable PCIe to wake from deep Sx. 0: Disable; 1: Enable.
- UUINT32 [WoWlanEnable](#): 1
Determine if WLAN wake from Sx, corresponds to the "HOST_WLAN_PP_EN" bit in the PWRM_CFG3 register. 0: Disable; 1: Enable.
- UUINT32 [WoWlanDeepSxEnable](#): 1
Determine if WLAN wake from DeepSx, corresponds to the "DSX_WLAN_PP_EN" bit in the PWRM_CFG3 register. 0: Disable; 1: Enable.
- UUINT32 [LanWakeFromDeepSx](#): 1
Determine if enable LAN to wake from deep Sx. 0: Disable; 1: Enable.

11.91.1 Detailed Description

This structure allows to customize PCH wake up capability from S5 or DeepSx by WOL, LAN, PCIE wake events.

Definition at line 1158 of file PchPolicyCommon.h.

11.91.2 Member Data Documentation

11.91.2.1 UUINT32 PCH_WAKE_CONFIG::Gp27WakeFromDeepSx

Deprecated

Definition at line 1166 of file PchPolicyCommon.h.

11.91.2.2 UUINT32 PCH_WAKE_CONFIG::PmeB0S5Dis

Corresponds to the PME_B0_S5_DIS bit in the General PM Configuration B (GEN_PMCON_B) register.

When set to 1, this bit blocks wake events from PME_B0_STS in S5, regardless of the state of PME_B0_EN. When cleared (default), wake events from PME_B0_STS are allowed in S5 if PME_B0_EN = 1. **0: Disable**; 1: Enable.

Definition at line 1164 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.92 PCH_WDT_CONFIG Struct Reference

This policy clears status bits and disable watchdog, then lock the WDT registers.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [DisableAndLock](#): 1
(Test) Set 1 to clear WDT status, then disable and lock WDT registers. 0: Disable; 1: Enable.

11.92.1 Detailed Description

This policy clears status bits and disable watchdog, then lock the WDT registers.

while WDT is designed to be disabled and locked by Policy, bios should not enable WDT by WDT PPI. In such case, bios shows the warning message but not disable and lock WDT register to make sure WDT event trigger correctly.

Definition at line 1701 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.93 PCH_XDCI_CONFIG Struct Reference

The [PCH_XDCI_CONFIG](#) block describes the configurations of the xDCI Usb Device controller.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [Enable](#): 1
This member describes whether or not the xDCI controller should be enabled.
- UUINT32 [RsvdBits0](#): 31
Reserved bits.

11.93.1 Detailed Description

The [PCH_XDCI_CONFIG](#) block describes the configurations of the xDCI Usb Device controller.

Definition at line 1584 of file PchPolicyCommon.h.

11.93.2 Member Data Documentation

11.93.2.1 UINT32 PCH_XDCI_CONFIG::Enable

This member describes whether or not the xDCI controller should be enabled.

0: Disable; 1: **Enable**.

Definition at line 1589 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.94 PCH_XHCI_SSIK_PORT Struct Reference

These members describe some settings which are related to the SSIC ports.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [Enable](#): 1
*0: Disable; 1: **Enable** SSIC support.*

11.94.1 Detailed Description

These members describe some settings which are related to the SSIC ports.

Definition at line 1566 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.95 PER_LANE_EPARAM_LINK_INFO Struct Reference

Per Lane PHY Configuration.

```
#include <KtiHost.h>
```

Public Attributes

- UINT8 [SocketID](#)
Socket ID.
 - UINT8 [AllLanesUseSameTxex](#)
Use same TXEQ on all lanes.
 - UINT8 [Freq](#)
The Link Speed these TXEQ settings should be used for.
 - UINT32 [Link](#)
Port Number.
 - UINT32 [TXEQL](#) [20]
TXEQ Settings.
 - UINT32 [CTLEPEAK](#) [5]
CTLE Peaking Settings.
-

11.95.1 Detailed Description

Per Lane PHY Configuration.

These PHY settings are system dependent. Every socket/link/freq requires an instance of this structure.

Definition at line 107 of file KtiHost.h.

The documentation for this struct was generated from the following file:

- [KtiHost.h](#)

11.96 PPR_ADDR Struct Reference

PPR DRAM Address.

```
#include <MemoryPolicyPpi.h>
```

11.96.1 Detailed Description

PPR DRAM Address.

Definition at line 347 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

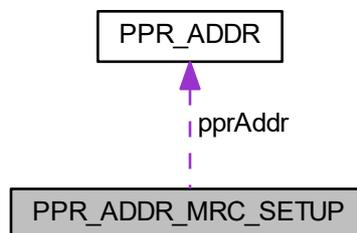
- [MemoryPolicyPpi.h](#)

11.97 PPR_ADDR_MRC_SETUP Struct Reference

PPR Address, buffer to hold DRAM Address that need to be repaired.

```
#include <MemoryPolicyPpi.h>
```

Collaboration diagram for PPR_ADDR_MRC_SETUP:



11.97.1 Detailed Description

PPR Address, buffer to hold DRAM Address that need to be repaired.

Definition at line 359 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.98 PROTECTED_RANGE Struct Reference

The PCH provides a method for blocking writes and reads to specific ranges in the SPI flash when the Protected Ranges are enabled.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [WriteProtectionEnable](#): 1
Write or erase is blocked by hardware. 0: Disable; 1: Enable.
- UUINT32 [ReadProtectionEnable](#): 1
Read is blocked by hardware. 0: Disable; 1: Enable.
- UUINT32 [RsvdBits](#): 30
Reserved.
- UUINT16 [ProtectedRangeLimit](#)
The address of the upper limit of protection This is a left shifted address by 12 bits with address bits 11:0 are assumed to be FFFh for limit comparison.
- UUINT16 [ProtectedRangeBase](#)
The address of the upper limit of protection This is a left shifted address by 12 bits with address bits 11:0 are assumed to be 0.

11.98.1 Detailed Description

The PCH provides a method for blocking writes and reads to specific ranges in the SPI flash when the Protected Ranges are enabled.

[PROTECTED_RANGE](#) is used to specify if flash protection are enabled, the write protection enable bit and the read protection enable bit, and to specify the upper limit and lower base for each register Platform code is responsible to get the range base by PchGetSpiRegionAddresses routine, and set the limit and base accordingly.

Definition at line 1667 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

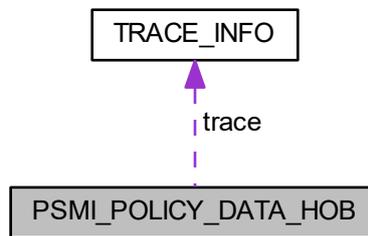
- [PchPolicyCommon.h](#)

11.99 PSMI_POLICY_DATA_HOB Struct Reference

PSMI policy.

```
#include <PsmiPolicyHob.h>
```

Collaboration diagram for PSMI_POLICY_DATA_HOB:



Public Attributes

- [UINT8 GlobalPsmiEnable](#)
*Global PSMI Enable - 0: Disable; 1: **Enable**; 2: Force setup.*
- [UINT8 PsmiTrace](#) [MAX_SOCKET]
PSMI Trace.
- [UINT8 PsmiHandlerSize](#) [MAX_SOCKET]
PSMI Handler Size.
- [TRACE_INFO trace](#) [MAX_SOCKET]
PSMI Trace Info.

11.99.1 Detailed Description

PSMI policy.

Definition at line 25 of file PsmiPolicyHob.h.

The documentation for this struct was generated from the following file:

- [PsmiPolicyHob.h](#)

11.100 RAS_RC_POLICY_PPI Struct Reference

RAS policy being requested of RC.

```
#include <RasRcPolicyPpi.h>
```

Public Attributes

- [UINT8 RasModes](#)
RAS Modes requested per policy.
- [UINT16 RasModesEx](#)
RAS Extended Modes requested per policy.
- [BOOLEAN McBankWarmBootClearError](#)
*Mca Bank Warm Boot Clear Errors - 0: Disable; 1: **Enable**.*

- UINT8 [PoisonEn](#)
*System Memory Poison - 0: Disable; 1: **Enable**.*
- UINT8 [PfdEn](#)
*PFD Enable - 0: Disable; 1: **Enable**.*
- UINT8 [CrashLogFeature](#)
*CrashLog Feature - 0: Disable; 1: **Enable**.*
- UINT8 [CrashLogOnAllReset](#)
*Invoke CrashLog collection on all reset - 0: **Disable**; 1: Enable.*
- UINT8 [CrashLogClear](#)
- UINT8 [CrashLogReArm](#)

11.100.1 Detailed Description

RAS policy being requested of RC.

Definition at line 16 of file RasRcPolicyPpi.h.

11.100.2 Member Data Documentation

11.100.2.1 UINT8 RAS_RC_POLICY_PPI::CrashLogClear

Deprecated

Definition at line 24 of file RasRcPolicyPpi.h.

11.100.2.2 UINT8 RAS_RC_POLICY_PPI::CrashLogReArm

Deprecated

Definition at line 25 of file RasRcPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [RasRcPolicyPpi.h](#)

11.101 SATA_THERMAL_THROTTLE Struct Reference

This structure lists PCH supported SATA thermal throttling register setting for customization.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UINT32 [P0T1M](#): 2
Port 0 T1 Multiplier.
 - UINT32 [P0T2M](#): 2
Port 0 T2 Multiplier.
 - UINT32 [P0T3M](#): 2
Port 0 T3 Multiplier.
 - UINT32 [P0TDisp](#): 2
Port 0 Tdispatch.
-

- UINT32 [P1T1M](#): 2
Port 1 T1 Multiplier.
- UINT32 [P1T2M](#): 2
Port 1 T2 Multiplier.
- UINT32 [P1T3M](#): 2
Port 1 T3 Multiplier.
- UINT32 [P1TDisp](#): 2
Port 1 Tdispatch.
- UINT32 [P0Tinact](#): 2
Port 0 Tinactive.
- UINT32 [P0TDispFinit](#): 1
Port 0 Alternate Fast Init Tdispatch.
- UINT32 [P1Tinact](#): 2
Port 1 Tinactive.
- UINT32 [P1TDispFinit](#): 1
Port 1 Alternate Fast Init Tdispatch.
- UINT32 [SuggestedSetting](#): 1
*0: Disable; 1: **Enable** suggested representative values*
- UINT32 [RsvdBits0](#): 9
Reserved bits.

11.101.1 Detailed Description

This structure lists PCH supported SATA thermal throttling register setting for customization.

The settings is programmed through SATA Index/Data registers. When the SuggestedSetting is enabled, the customized values are ignored.

Definition at line 1000 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.102 SECURITY_POLICY Struct Reference

Security Policy.

```
#include <SecurityPolicy.h>
```

Public Attributes

- UINT8 [EnableTme](#)
TME Enable.
- UINT8 [EnableTmeCR](#)
TME for Optane Persistent Memory. Set to 0 exclude Optane from encryption.
- UINT8 [EnableMktme](#)
MK-TME Enable.
- UINT8 [EnableSgx](#)
Enable SGX.
- UINT8 [SgxFactoryReset](#)
Delete all registration data, if SGX enabled force IPE/FirstBinding flow.

- UINT64 [PrmrrSize](#)
SGX PRMRR size.
- UINT8 [SgxQoS](#)
SGX Quality of Service.
- UINT8 [SgxAutoRegistrationAgent](#)
SGX Auto Registration Agent.
- UINT8 [SgxPackageInfoInBandAccess](#)
SGX Expose Package Info to OS.
- UINT8 [EpochUpdate](#)
SGX EPOCH Update.
- UINT64 [SgxEpoch0](#)
SGX EPOCH0 value {0 - 0xFFFFFFFFFFFFFFFF}.
- UINT64 [SgxEpoch1](#)
SGX EPOCH1 value {0 - 0xFFFFFFFFFFFFFFFF}.
- UINT8 [SgxLeWr](#)
Flexible Launch Enclave Policy (Wr En)
- UINT64 [SgxLePubKeyHash0](#)
Launch Enclave Hash 0.
- UINT64 [SgxLePubKeyHash1](#)
Launch Enclave Hash 1.
- UINT64 [SgxLePubKeyHash2](#)
Launch Enclave Hash 2.
- UINT64 [SgxLePubKeyHash3](#)
Launch Enclave Hash 3.
- UINT8 [SgxSinitNvsData](#)
- UINT8 [SgxSinitDataFromTpm](#)
- UINT8 [SgxDebugMode](#)
- UINT8 [EnableTdx](#)
TDX Enable.
- UINT8 [KeySplit](#)
TDX/MK-TME key split.

11.102.1 Detailed Description

Security Policy.

Definition at line 21 of file SecurityPolicy.h.

11.102.2 Member Data Documentation

11.102.2.1 UINT8 SECURITY_POLICY::SgxDebugMode

Deprecated

Definition at line 70 of file SecurityPolicy.h.

11.102.2.2 UINT8 SECURITY_POLICY::SgxSinitDataFromTpm

Deprecated SGX SVN data from TPM; 0: when SGX is disabled or TPM is not present or no data is present in TPM.

Definition at line 68 of file SecurityPolicy.h.

11.102.2.3 UINT8 SECURITY_POLICY::SgxSinitNvsData

Deprecated SGX NVS data from Flash passed during previous boot using CPU_INFO_PROTOCOL.SGX_INFO; Pass value of zero if there is not data saved or when SGX is disabled.

Definition at line 66 of file SecurityPolicy.h.

The documentation for this struct was generated from the following file:

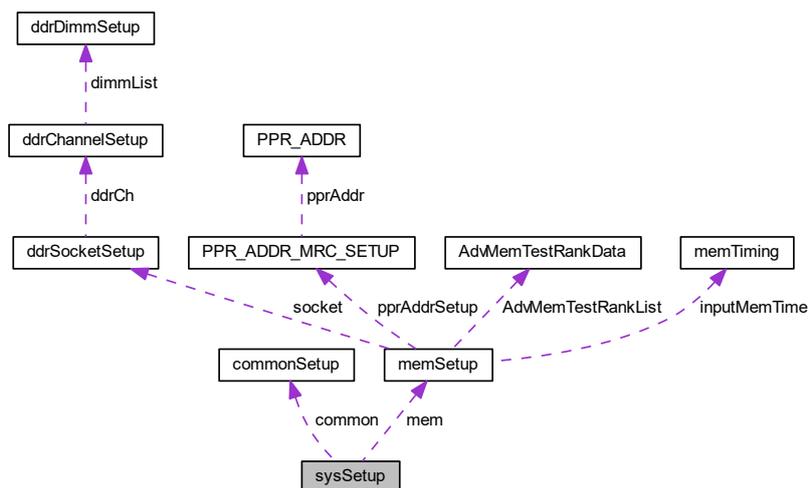
- [SecurityPolicy.h](#)

11.103 sysSetup Struct Reference

Platform Setting for MRC.

```
#include <MemoryPolicyPpi.h>
```

Collaboration diagram for sysSetup:



Public Attributes

- struct [memSetup](#) `mem`
Memory technology related settings for MRC.
- struct [commonSetup](#) `common`
Common platform settings not related to memory technology.
- UINT8 [WFRWAEEnable](#)
WFR Uncore GV Rate Reduction.
- UINT8 [PmaxDisable](#)
Enable/Disable Pmax through BIOS to Pcode Mailbox.
- UINT32 [AdrEvent](#)
Whether of not we should recover from ADR.

11.103.1 Detailed Description

Platform Setting for MRC.

Definition at line 2050 of file MemoryPolicyPpi.h.

11.103.2 Member Data Documentation

11.103.2.1 UINT8 sysSetup::WFRWAEEnable

WFR Uncore GV Rate Reduction.

AUTO: Enable if WFR socket is detected in system.

Enabled: Always enables WFR Uncore GV Rate Reduction.

Definition at line 2069 of file MemoryPolicyPpi.h.

The documentation for this struct was generated from the following file:

- [MemoryPolicyPpi.h](#)

11.104 THERMAL_THROTTLE_LEVELS Struct Reference

This structure lists PCH supported throttling register setting for customization.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- [UINT32 T0Level](#): 9
Customized T0Level value. If SuggestedSetting is used, this setting is ignored.
- [UINT32 T1Level](#): 9
Customized T1Level value. If SuggestedSetting is used, this setting is ignored.
- [UINT32 T2Level](#): 9
Customized T2Level value. If SuggestedSetting is used, this setting is ignored.
- [UINT32 TTEnable](#): 1
Enable the thermal throttle function. If SuggestedSetting is used, this settings is ignored.
- [UINT32 TTState13Enable](#): 1
When set to 1 and the programmed GPIO pin is a 1, then PMSync state 13 will force at least T2 state.
- [UINT32 TTLock](#): 1
When set to 1, this entire register (TL) is locked and remains locked until the next platform reset.
- [UINT32 SuggestedSetting](#): 1
*0: Disable; 1: **Enable** suggested representative values.*
- [UINT32 PchCrossThrottling](#): 1
ULT processors support thermal management and cross thermal throttling between the processor package and LP PCH.
- [UINT32 Rsvd0](#)
Reserved bytes.

11.104.1 Detailed Description

This structure lists PCH supported throttling register setting for customization.

When the SuggestedSetting is enabled, the customized values are ignored.

Definition at line 954 of file PchPolicyCommon.h.

11.104.2 Member Data Documentation

11.104.2.1 UINT32 THERMAL_THROTTLE_LEVELS::PchCrossThrottling

ULT processors support thermal management and cross thermal throttling between the processor package and LP PCH.

The PMSYNC message from PCH to CPU includes specific bit fields to update the PCH thermal status to the processor which is factored into the processor throttling. Enable/Disable PCH Cross Throttling; 0: Disabled, 1: **Enabled**.

Definition at line 976 of file PchPolicyCommon.h.

11.104.2.2 UINT32 THERMAL_THROTTLE_LEVELS::TTLock

When set to 1, this entire register (TL) is locked and remains locked until the next platform reset.

If SuggestedSetting is used, this setting is ignored.

Definition at line 968 of file PchPolicyCommon.h.

11.104.2.3 UINT32 THERMAL_THROTTLE_LEVELS::TTState13Enable

When set to 1 and the programmed GPIO pin is a 1, then PMSync state 13 will force at least T2 state.

If SuggestedSetting is used, this setting is ignored.

Definition at line 963 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)

11.105 TRACE_INFO Struct Reference

Trace Info.

```
#include <PsmiPolicyHob.h>
```

Public Attributes

- UINT8 [PsmiTraceRegion](#) [5]
Psmi Trace Region.
- UINT8 [PsmiTraceBufferSizeRegion](#) [5]
Psmi Trace Buffer Size Region.
- UINT8 [PsmiTraceMemTypeRegion](#) [5]
Psmi Trace Memory Type Region.

11.105.1 Detailed Description

Trace Info.

Definition at line 16 of file PsmiPolicyHob.h.

The documentation for this struct was generated from the following file:

- [PsmiPolicyHob.h](#)

11.106 TS_GPIO_PIN_SETTING Struct Reference

This structure configures PCH memory throttling thermal sensor GPIO PIN settings.

```
#include <PchPolicyCommon.h>
```

Public Attributes

- UUINT32 [PmsyncEnable](#): 1
GPIO PM_SYNC enable, 0:Disabled, 1:Enabled When enabled, RC will overrides the selected GPIO native mode.
- UUINT32 [C0TransmitEnable](#): 1
GPIO Transmit enable in C0 state, 0:Disabled, 1:Enabled
- UUINT32 [PinSelection](#): 1
GPIO Pin assignment selection, 0: default, 1: secondary.

11.106.1 Detailed Description

This structure configures PCH memory throttling thermal sensor GPIO PIN settings.

Definition at line 1033 of file PchPolicyCommon.h.

11.106.2 Member Data Documentation

11.106.2.1 UUINT32 TS_GPIO_PIN_SETTING::PmsyncEnable

GPIO PM_SYNC enable, 0:Disabled, 1:**Enabled** When enabled, RC will overrides the selected GPIO native mode.

For GPIO_C, PinSelection 0: CPU_GP_0 (default) or 1: CPU_GP_1 For GPIO_D, PinSelection 0: CPU_GP_↔
3 (default) or 1: CPU_GP_2 For SKL: CPU_GP_0 is GPP_E3, CPU_GP_1 is GPP_E7, CPU_GP_2 is GPP_B3,
CPU_GP_3 is GPP_B4.

Definition at line 1041 of file PchPolicyCommon.h.

The documentation for this struct was generated from the following file:

- [PchPolicyCommon.h](#)
-

Chapter 12

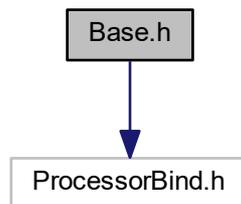
File Documentation

12.1 Base.h File Reference

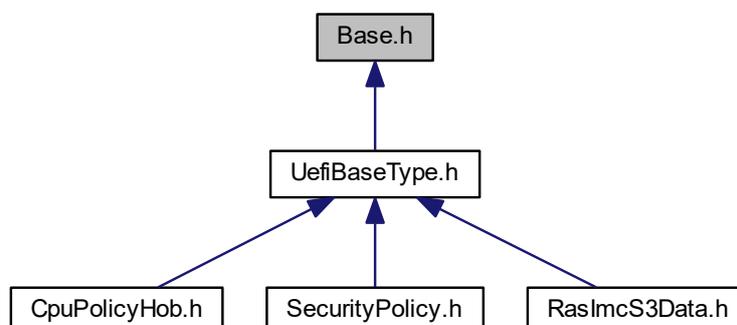
Root include file for Mde Package Base type modules.

```
#include <ProcessorBind.h>
```

Include dependency graph for Base.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [GUID](#)
128 bit buffer containing a unique identifier value.
- struct [IPv4_ADDRESS](#)
4-byte buffer.
- struct [IPv6_ADDRESS](#)
16-byte buffer.
- struct [_LIST_ENTRY](#)
[_LIST_ENTRY](#) structure definition.

Macros

- #define [GLOBAL_REMOVE_IF_UNREFERENCED](#)
Remove the global variable from the linked image if there are no references to it after all compiler and linker optimizations have been performed.
- #define [UNREACHABLE\(\)](#)
Signal compilers and analyzers that this call is not reachable.
- #define [NORETURN](#)
Signal compilers and analyzers that the function cannot return.
- #define [ANALYZER_UNREACHABLE\(\)](#)
Signal the analyzer that this call is not reachable.
- #define [ANALYZER_NORETURN](#)
Signal the analyzer that the function cannot return.
- #define [RETURNS_TWICE](#)
Tell the code optimizer that the function will return twice.
- #define [_CONCATENATE](#)(a, b) [__CONCATENATE](#)(a, b)
Private worker functions for [ASM_PFX\(\)](#)
- #define [ASM_PFX](#)(name) [_CONCATENATE](#) ([__USER_LABEL_PREFIX__](#), name)
The [USER_LABEL_PREFIX](#) macro predefined by GNUC represents the prefix on symbols in assembly language.
- #define [CONST](#) const
Datum is read-only.
- #define [STATIC](#) static
Datum is scoped to the current file or function.
- #define [VOID](#) void
Undeclared type.
- #define [IN](#)
Datum is passed to the function.
- #define [OUT](#)
Datum is returned from the function.
- #define [OPTIONAL](#)
Passing the datum to the function is optional, and a NULL is passed if the value is not supplied.
- #define [TRUE](#) ((BOOLEAN)(1==1))
Boolean true value.
- #define [FALSE](#) ((BOOLEAN)(0==1))
Boolean false value.
- #define [NULL](#) ((VOID *) 0)
*NULL pointer (VOID *)*
- #define [MAX_INT8](#) ((INT8)0x7F)
Maximum values for common UEFI Data Types.
- #define [MIN_INT8](#) (((INT8) -127) - 1)

- Minimum values for the signed UEFI Data Types.*
- #define `_INT_SIZE_OF(n)` `((sizeof (n) + sizeof (UINTN) - 1) &~(sizeof (UINTN) - 1))`
Return the size of argument that has been aligned to sizeof (UINTN).
 - #define `VA_START(Marker, Parameter)` `(Marker = (VA_LIST) ((UINTN) & (Parameter) + _INT_SIZE_OF (Parameter)))`
Retrieves a pointer to the beginning of a variable argument list, based on the name of the parameter that immediately precedes the variable argument list.
 - #define `VA_ARG(Marker, TYPE)` `((*(TYPE *) ((Marker += _INT_SIZE_OF (TYPE)) - _INT_SIZE_OF (TYPE)))`
Returns an argument of a specified type from a variable argument list and updates the pointer to the variable argument list to point to the next argument.
 - #define `VA_END(Marker)` `(Marker = (VA_LIST) 0)`
Terminates the use of a variable argument list.
 - #define `VA_COPY(Dest, Start)` `((void)((Dest) = (Start)))`
Initializes a VA_LIST as a copy of an existing VA_LIST.
 - #define `_BASE_INT_SIZE_OF(TYPE)` `((sizeof (TYPE) + sizeof (UINTN) - 1) / sizeof (UINTN))`
Returns the size of a data type in sizeof(UINTN) units rounded up to the nearest UINTN boundary.
 - #define `BASE_ARG(Marker, TYPE)` `((*(TYPE *) ((Marker += _BASE_INT_SIZE_OF (TYPE)) - _BASE_INT_SIZE_OF (TYPE)))`
Returns an argument of a specified type from a variable argument list and updates the pointer to the variable argument list to point to the next argument.
 - #define `OFFSET_OF(TYPE, Field)` `((UINTN) &(((TYPE *)0)->Field))`
The macro that returns the byte offset of a field in a data structure.
 - #define `STATIC_ASSERT _Static_assert`
Portable definition for compile time assertions.
 - #define `BASE_CR(Record, TYPE, Field)` `((TYPE *) ((CHAR8 *) (Record) - OFFSET_OF (TYPE, Field)))`
Macro that returns a pointer to the data structure that contains a specified field of that data structure.
 - #define `ALIGN_VALUE(Value, Alignment)` `((Value) + (((Alignment) - (Value)) & ((Alignment) - 1)))`
Rounds a value up to the next boundary using a specified alignment.
 - #define `ALIGN_POINTER(Pointer, Alignment)` `((VOID *) (ALIGN_VALUE ((UINTN)(Pointer), (Alignment))))`
Adjust a pointer by adding the minimum offset required for it to be aligned on a specified alignment boundary.
 - #define `ALIGN_VARIABLE(Value)` `ALIGN_VALUE ((Value), sizeof (UINTN))`
Rounds a value up to the next natural boundary for the current CPU.
 - #define `MAX(a, b)` `((a) > (b)) ? (a) : (b)`
Return the maximum of two operands.
 - #define `MIN(a, b)` `((a) < (b)) ? (a) : (b)`
Return the minimum of two operands.
 - #define `ABS(a)` `((a) < 0) ? (-a) : (a)`
Return the absolute value of a signed operand.
 - #define `ENCODE_ERROR(StatusCode)` `((RETURN_STATUS)(MAX_BIT | (StatusCode)))`
Produces a RETURN_STATUS code with the highest bit set.
 - #define `ENCODE_WARNING(StatusCode)` `((RETURN_STATUS)(StatusCode))`
Produces a RETURN_STATUS code with the highest bit clear.
 - #define `RETURN_ERROR(StatusCode)` `((INTN)(RETURN_STATUS)(StatusCode)) < 0`
Returns TRUE if a specified RETURN_STATUS code is an error code.
 - #define `RETURN_SUCCESS` 0
The operation completed successfully.
 - #define `RETURN_LOAD_ERROR ENCODE_ERROR (1)`
The image failed to load.
 - #define `RETURN_INVALID_PARAMETER ENCODE_ERROR (2)`
The parameter was incorrect.
 - #define `RETURN_UNSUPPORTED ENCODE_ERROR (3)`
-

- The operation is not supported.*

 - #define `RETURN_BAD_BUFFER_SIZE ENCODE_ERROR` (4)
The buffer was not the proper size for the request.
 - #define `RETURN_BUFFER_TOO_SMALL ENCODE_ERROR` (5)
The buffer was not large enough to hold the requested data.
 - #define `RETURN_NOT_READY ENCODE_ERROR` (6)
There is no data pending upon return.
 - #define `RETURN_DEVICE_ERROR ENCODE_ERROR` (7)
The physical device reported an error while attempting the operation.
 - #define `RETURN_WRITE_PROTECTED ENCODE_ERROR` (8)
The device can not be written to.
 - #define `RETURN_OUT_OF_RESOURCES ENCODE_ERROR` (9)
The resource has run out.
 - #define `RETURN_VOLUME_CORRUPTED ENCODE_ERROR` (10)
An inconsistency was detected on the file system causing the operation to fail.
 - #define `RETURN_VOLUME_FULL ENCODE_ERROR` (11)
There is no more space on the file system.
 - #define `RETURN_NO_MEDIA ENCODE_ERROR` (12)
The device does not contain any medium to perform the operation.
 - #define `RETURN_MEDIA_CHANGED ENCODE_ERROR` (13)
The medium in the device has changed since the last access.
 - #define `RETURN_NOT_FOUND ENCODE_ERROR` (14)
The item was not found.
 - #define `RETURN_ACCESS_DENIED ENCODE_ERROR` (15)
Access was denied.
 - #define `RETURN_NO_RESPONSE ENCODE_ERROR` (16)
The server was not found or did not respond to the request.
 - #define `RETURN_NO_MAPPING ENCODE_ERROR` (17)
A mapping to the device does not exist.
 - #define `RETURN_TIMEOUT ENCODE_ERROR` (18)
A timeout time expired.
 - #define `RETURN_NOT_STARTED ENCODE_ERROR` (19)
The protocol has not been started.
 - #define `RETURN_ALREADY_STARTED ENCODE_ERROR` (20)
The protocol has already been started.
 - #define `RETURN_ABORTED ENCODE_ERROR` (21)
The operation was aborted.
 - #define `RETURN_ICMP_ERROR ENCODE_ERROR` (22)
An ICMP error occurred during the network operation.
 - #define `RETURN_TFTP_ERROR ENCODE_ERROR` (23)
A TFTP error occurred during the network operation.
 - #define `RETURN_PROTOCOL_ERROR ENCODE_ERROR` (24)
A protocol error occurred during the network operation.
 - #define `RETURN_INCOMPATIBLE_VERSION ENCODE_ERROR` (25)
A function encountered an internal version that was incompatible with a version requested by the caller.
 - #define `RETURN_SECURITY_VIOLATION ENCODE_ERROR` (26)
The function was not performed due to a security violation.
 - #define `RETURN_CRC_ERROR ENCODE_ERROR` (27)
A CRC error was detected.
 - #define `RETURN_END_OF_MEDIA ENCODE_ERROR` (28)
The beginning or end of media was reached.
-

- #define [RETURN_END_OF_FILE_ENCODE_ERROR](#) (31)
The end of the file was reached.
- #define [RETURN_INVALID_LANGUAGE_ENCODE_ERROR](#) (32)
The language specified was invalid.
- #define [RETURN_COMPROMISED_DATA_ENCODE_ERROR](#) (33)
The security status of the data is unknown or compromised and the data must be updated or replaced to restore a valid security status.
- #define [RETURN_HTTP_ERROR_ENCODE_ERROR](#) (35)
A HTTP error occurred during the network operation.
- #define [RETURN_WARN_UNKNOWN_GLYPH_ENCODE_WARNING](#) (1)
The string contained one or more characters that the device could not render and were skipped.
- #define [RETURN_WARN_DELETE_FAILURE_ENCODE_WARNING](#) (2)
The handle was closed, but the file was not deleted.
- #define [RETURN_WARN_WRITE_FAILURE_ENCODE_WARNING](#) (3)
The handle was closed, but the data to the file was not flushed properly.
- #define [RETURN_WARN_BUFFER_TOO_SMALL_ENCODE_WARNING](#) (4)
The resulting buffer was too small, and the data was truncated to the buffer size.
- #define [RETURN_WARN_STALE_DATA_ENCODE_WARNING](#) (5)
The data has not been updated within the timeframe set by local policy for this type of data.
- #define [RETURN_WARN_FILE_SYSTEM_ENCODE_WARNING](#) (6)
The resulting buffer contains UEFI-compliant file system.
- #define [SIGNATURE_16\(A, B\)](#) ((A) | (B << 8))
Returns a 16-bit signature built from 2 ASCII characters.
- #define [SIGNATURE_32\(A, B, C, D\)](#) ([SIGNATURE_16](#) (A, B) | ([SIGNATURE_16](#) (C, D) << 16))
Returns a 32-bit signature built from 4 ASCII characters.
- #define [SIGNATURE_64\(A, B, C, D, E, F, G, H\)](#) ([SIGNATURE_32](#) (A, B, C, D) | ((UINT64) ([SIGNATURE_32](#) (E, F, G, H) << 32))
Returns a 64-bit signature built from 8 ASCII characters.
- #define [RETURN_ADDRESS\(L\)](#) ((VOID *) 0)
Get the return address of the calling function.
- #define [ARRAY_SIZE\(Array\)](#) (sizeof (Array) / sizeof ((Array)[0]))
Return the number of elements in an array.

Typedefs

- typedef struct [_LIST_ENTRY](#) [LIST_ENTRY](#)
LIST_ENTRY structure definition.
- typedef CHAR8 * [VA_LIST](#)
Variable used to traverse the list of arguments.
- typedef UINTN * [BASE_LIST](#)
Pointer to the start of a variable argument list stored in a memory buffer.

12.1.1 Detailed Description

Root include file for Mde Package Base type modules.

This is the include file for any module of type base. Base modules only use types defined via this include file and can be ported easily to any environment. There are a set of base libraries in the Mde Package that can be used to implement base modules.

Copyright (c) 2006 - 2021, Intel Corporation. All rights reserved.
Portions copyright (c) 2008 - 2009, Apple Inc. All rights reserved.
SPDX-License-Identifier: BSD-2-Clause-Patent

12.1.2 Macro Definition Documentation

12.1.2.1 `#define _BASE_INT_SIZE_OF(TYPE) ((sizeof (TYPE) + sizeof (UINTN) - 1) / sizeof (UINTN))`

Returns the size of a data type in sizeof(UINTN) units rounded up to the nearest UINTN boundary.

Parameters

<i>TYPE</i>	The date type to determine the size of.
-------------	---

Returns

The size of TYPE in sizeof (UINTN) units rounded up to the nearest UINTN boundary.

Definition at line 751 of file Base.h.

12.1.2.2 `#define _INT_SIZE_OF(n) ((sizeof (n) + sizeof (UINTN) - 1) &~(sizeof (UINTN) - 1))`

Return the size of argument that has been aligned to sizeof (UINTN).

Parameters

<i>n</i>	The parameter size to be aligned.
----------	-----------------------------------

Returns

The aligned size.

Definition at line 580 of file Base.h.

12.1.2.3 `#define ABS(a) (((a) < 0) ? (-a) : (a))`

Return the absolute value of a signed operand.

This macro returns the absolute value of the signed operand specified by a.

Parameters

<i>a</i>	The signed operand.
----------	---------------------

Returns

The absolute value of the signed operand.

Definition at line 956 of file Base.h.

12.1.2.4 `#define ALIGN_POINTER(Pointer, Alignment) ((VOID *) (ALIGN_VALUE ((UINTN)(Pointer), (Alignment))))`

Adjust a pointer by adding the minimum offset required for it to be aligned on a specified alignment boundary.

This function rounds the pointer specified by Pointer to the next alignment boundary specified by Alignment. The pointer to the aligned address is returned.

Parameters

<i>Pointer</i>	The pointer to round up.
<i>Alignment</i>	The alignment boundary to use to return an aligned pointer.

Returns

Pointer to the aligned address.

Definition at line 898 of file Base.h.

12.1.2.5 `#define ALIGN_VALUE(Value, Alignment) ((Value) + (((Alignment) - (Value)) & ((Alignment) - 1)))`

Rounds a value up to the next boundary using a specified alignment.

This function rounds Value up to the next boundary using the specified Alignment. This aligned value is returned.

Parameters

<i>Value</i>	The value to round up.
<i>Alignment</i>	The alignment boundary used to return the aligned value.

Returns

A value up to the next boundary.

Definition at line 883 of file Base.h.

12.1.2.6 `#define ALIGN_VARIABLE(Value) ALIGN_VALUE ((Value), sizeof (UINTN))`

Rounds a value up to the next natural boundary for the current CPU.

This is 4-bytes for 32-bit CPUs and 8-bytes for 64-bit CPUs.

This function rounds the value specified by Value up to the next natural boundary for the current CPU. This rounded value is returned.

Parameters

<i>Value</i>	The value to round up.
--------------	------------------------

Returns

Rounded value specified by Value.

Definition at line 912 of file Base.h.

12.1.2.7 `#define ANALYZER_NORETURN`

Signal the analyzer that the function cannot return.

This excludes compilers.

Definition at line 158 of file Base.h.

12.1.2.8 `#define ANALYZER_UNREACHABLE()`

Signal the analyzer that this call is not reachable.

This excludes compilers.

Definition at line 132 of file Base.h.

12.1.2.9 `#define ARRAY_SIZE(Array) (sizeof (Array) / sizeof ((Array)[0]))`

Return the number of elements in an array.

Parameters

<i>Array</i>	An object of array type. Array is only used as an argument to the sizeof operator, therefore Array is never evaluated. The caller is responsible for ensuring that Array's type is not incomplete; that is, Array must have known constant size.
--------------	--

Returns

The number of elements in Array. The result has type UINTN.

Definition at line 1313 of file Base.h.

```
12.1.2.10 #define BASE_ARG( Marker, TYPE ) (*(TYPE *) ((Marker += _BASE_INT_SIZE_OF (TYPE)) -
    _BASE_INT_SIZE_OF (TYPE)))
```

Returns an argument of a specified type from a variable argument list and updates the pointer to the variable argument list to point to the next argument.

This function returns an argument of the type specified by TYPE from the beginning of the variable argument list specified by Marker. Marker is then updated to point to the next argument in the variable argument list. The method for computing the pointer to the next argument in the argument list is CPU specific following the EFI API ABI.

Parameters

<i>Marker</i>	The pointer to the beginning of a variable argument list.
<i>TYPE</i>	The type of argument to retrieve from the beginning of the variable argument list.

Returns

An argument of the type specified by TYPE.

Definition at line 769 of file Base.h.

```
12.1.2.11 #define BASE_CR( Record, TYPE, Field ) ((TYPE *) ((CHAR8 *) (Record) - OFFSET_OF (TYPE, Field)))
```

Macro that returns a pointer to the data structure that contains a specified field of that data structure.

This is a lightweight method to hide information by placing a public data structure inside a larger private data structure and using a pointer to the public data structure to retrieve a pointer to the private data structure.

This function computes the offset, in bytes, of field specified by Field from the beginning of the data structure specified by TYPE. This offset is subtracted from Record, and is used to return a pointer to a data structure of the type specified by TYPE. If the data type specified by TYPE does not contain the field specified by Field, then the module will not compile.

Parameters

<i>Record</i>	Pointer to the field specified by Field within a data structure of type TYPE.
<i>TYPE</i>	The name of the data structure type to return. This data structure must contain the field specified by Field.
<i>Field</i>	The name of the field in the data structure specified by TYPE to which Record points.

Returns

A pointer to the structure from one of its elements.

Definition at line 869 of file Base.h.

```
12.1.2.12 #define ENCODE_ERROR( StatusCode ) ((RETURN_STATUS)(MAX_BIT | (StatusCode)))
```

Produces a RETURN_STATUS code with the highest bit set.

Parameters

<i>StatusCode</i>	The status code value to convert into a warning code. StatusCode must be in the range 0x00000000..0x7FFFFFFF.
-------------------	---

Returns

The value specified by StatusCode with the highest bit set.

Definition at line 973 of file Base.h.

12.1.2.13 `#define ENCODE_WARNING(StatusCode)((RETURN_STATUS)(StatusCode))`

Produces a RETURN_STATUS code with the highest bit clear.

Parameters

<i>StatusCode</i>	The status code value to convert into a warning code. StatusCode must be in the range 0x00000000..0x7FFFFFFF.
-------------------	---

Returns

The value specified by StatusCode with the highest bit clear.

Definition at line 984 of file Base.h.

12.1.2.14 `#define FALSE ((BOOLEAN)(0==1))`

Boolean false value.

UEFI Specification defines this value to be 0, but this form is more portable.

Definition at line 316 of file Base.h.

12.1.2.15 `#define MAX(a, b)(((a) > (b)) ? (a) : (b))`

Return the maximum of two operands.

This macro returns the maximum of two operand specified by *a* and *b*. Both *a* and *b* must be the same numerical types, signed or unsigned.

Parameters

<i>a</i>	The first operand with any numerical type.
<i>b</i>	The second operand. Can be any numerical type as long as is the same type as <i>a</i> .

Returns

Maximum of two operands.

Definition at line 928 of file Base.h.

12.1.2.16 `#define MIN(a, b)(((a) < (b)) ? (a) : (b))`

Return the minimum of two operands.

This macro returns the minimal of two operand specified by *a* and *b*. Both *a* and *b* must be the same numerical types, signed or unsigned.

Parameters

<i>a</i>	The first operand with any numerical type.
<i>b</i>	The second operand. It should be the same any numerical type with a.

Returns

Minimum of two operands.

Definition at line 943 of file Base.h.

12.1.2.17 #define NORETURN

Signal compilers and analyzers that the function cannot return.

It is up to the compiler to remove any code past a call to functions flagged with this attribute.

Definition at line 108 of file Base.h.

12.1.2.18 #define OFFSET_OF(TYPE, Field) ((UINTN) &(((TYPE *)0)->Field))

The macro that returns the byte offset of a field in a data structure.

This function returns the offset, in bytes, of field specified by Field from the beginning of the data structure specified by TYPE. If TYPE does not contain Field, the module will not compile.

Parameters

<i>TYPE</i>	The name of the data structure that contains the field specified by Field.
<i>Field</i>	The name of the field in the data structure.

Returns

Offset, in bytes, of field.

Definition at line 789 of file Base.h.

12.1.2.19 #define RETURN_ADDRESS(L) ((VOID *) 0)

Get the return address of the calling function.

Parameters

<i>L</i>	Return Level.
----------	---------------

Returns

0 as compilers don't support this feature.

Definition at line 1299 of file Base.h.

12.1.2.20 #define RETURN_BUFFER_TOO_SMALL ENCODE_ERROR (5)

The buffer was not large enough to hold the requested data.

The required buffer size is returned in the appropriate parameter when this error occurs.

Definition at line 1029 of file Base.h.

12.1.2.21 `#define RETURN_ERROR(StatusCode) (((INTN)(RETURN_STATUS)(StatusCode)) < 0)`

Returns TRUE if a specified RETURN_STATUS code is an error code.

This function returns TRUE if *StatusCode* has the high bit set. Otherwise, FALSE is returned.

Parameters

<i>StatusCode</i>	The status code value to evaluate.
-------------------	------------------------------------

Return values

<i>TRUE</i>	The high bit of <i>StatusCode</i> is set.
<i>FALSE</i>	The high bit of <i>StatusCode</i> is clear.

Definition at line 997 of file Base.h.

12.1.2.22 `#define RETURNS_TWICE`

Tell the code optimizer that the function will return twice.

This prevents wrong optimizations which can cause bugs. Tell the code optimizer that the function will return twice.

This prevents wrong optimizations which can cause bugs.

Definition at line 178 of file Base.h.

12.1.2.23 `#define SIGNATURE_16(A, B) ((A) | (B << 8))`

Returns a 16-bit signature built from 2 ASCII characters.

This macro returns a 16-bit value built from the two ASCII characters specified by *A* and *B*.

Parameters

<i>A</i>	The first ASCII character.
<i>B</i>	The second ASCII character.

Returns

A 16-bit value built from the two ASCII characters specified by *A* and *B*.

Definition at line 1220 of file Base.h.

12.1.2.24 `#define SIGNATURE_32(A, B, C, D) (SIGNATURE_16(A, B) | (SIGNATURE_16(C, D) << 16))`

Returns a 32-bit signature built from 4 ASCII characters.

This macro returns a 32-bit value built from the four ASCII characters specified by *A*, *B*, *C*, and *D*.

Parameters

<i>A</i>	The first ASCII character.
<i>B</i>	The second ASCII character.
<i>C</i>	The third ASCII character.
<i>D</i>	The fourth ASCII character.

Returns

A 32-bit value built from the two ASCII characters specified by *A*, *B*, *C* and *D*.

Definition at line 1237 of file Base.h.

12.1.2.25 `#define SIGNATURE_64(A, B, C, D, E, F, G, H) (SIGNATURE_32 (A, B, C, D) | ((UINT64) (SIGNATURE_32 (E, F, G, H)) << 32))`

Returns a 64-bit signature built from 8 ASCII characters.

This macro returns a 64-bit value built from the eight ASCII characters specified by A, B, C, D, E, F, G, and H.

Parameters

<i>A</i>	The first ASCII character.
<i>B</i>	The second ASCII character.
<i>C</i>	The third ASCII character.
<i>D</i>	The fourth ASCII character.
<i>E</i>	The fifth ASCII character.
<i>F</i>	The sixth ASCII character.
<i>G</i>	The seventh ASCII character.
<i>H</i>	The eighth ASCII character.

Returns

A 64-bit value built from the two ASCII characters specified by A, B, C, D, E, F, G and H.

Definition at line 1258 of file Base.h.

12.1.2.26 `#define STATIC_ASSERT _Static_assert`

Portable definition for compile time assertions.

Equivalent to C11 `static_assert` macro from `assert.h`.

Parameters

<i>Expression</i>	Boolean expression.
<i>Message</i>	Raised compiler diagnostic message when expression is false.

Definition at line 805 of file Base.h.

12.1.2.27 `#define TRUE ((BOOLEAN)(1==1))`

Boolean true value.

UEFI Specification defines this value to be 1, but this form is more portable.

Definition at line 310 of file Base.h.

12.1.2.28 `#define UNREACHABLE()`

Signal compilers and analyzers that this call is not reachable.

It is up to the compiler to remove any code past that point.

Definition at line 78 of file Base.h.

12.1.2.29 `#define VA_ARG(Marker, TYPE) (*(TYPE *) ((Marker += _INT_SIZE_OF (TYPE)) - _INT_SIZE_OF (TYPE)))`

Returns an argument of a specified type from a variable argument list and updates the pointer to the variable argument list to point to the next argument.

This function returns an argument of the type specified by `TYPE` from the beginning of the variable argument list specified by `Marker`. `Marker` is then updated to point to the next argument in the variable argument list. The method for computing the pointer to the next argument in the argument list is CPU-specific following the EFI API ABI.

Parameters

<i>Marker</i>	VA_LIST used to traverse the list of arguments.
<i>TYPE</i>	The type of argument to retrieve from the beginning of the variable argument list.

Returns

An argument of the type specified by TYPE.

Definition at line 710 of file Base.h.

12.1.2.30 `#define VA_COPY(Dest, Start) ((void)((Dest) = (Start)))`

Initializes a VA_LIST as a copy of an existing VA_LIST.

This macro initializes Dest as a copy of Start, as if the VA_START macro had been applied to Dest followed by the same sequence of uses of the VA_ARG macro as had previously been used to reach the present state of Start.

Parameters

<i>Dest</i>	VA_LIST used to traverse the list of arguments.
<i>Start</i>	VA_LIST used to traverse the list of arguments.

Definition at line 735 of file Base.h.

12.1.2.31 `#define VA_END(Marker) (Marker = (VA_LIST) 0)`

Terminates the use of a variable argument list.

This function initializes Marker so it can no longer be used with [VA_ARG\(\)](#). After this macro is used, the only way to access the variable argument list is by using [VA_START\(\)](#) again.

Parameters

<i>Marker</i>	VA_LIST used to traverse the list of arguments.
---------------	---

Definition at line 722 of file Base.h.

12.1.2.32 `#define VA_START(Marker, Parameter) (Marker = (VA_LIST) ((UINTN) & (Parameter) + _INT_SIZE_OF (Parameter)))`

Retrieves a pointer to the beginning of a variable argument list, based on the name of the parameter that immediately precedes the variable argument list.

This function initializes Marker to point to the beginning of the variable argument list that immediately follows Parameter. The method for computing the pointer to the next argument in the argument list is CPU-specific following the EFI API ABI.

Parameters

<i>Marker</i>	The VA_LIST used to traverse the list of arguments.
<i>Parameter</i>	The name of the parameter that immediately precedes the variable argument list.

Returns

A pointer to the beginning of a variable argument list.

Definition at line 692 of file Base.h.

12.1.3 Typedef Documentation

12.1.3.1 typedef UINTN* BASE_LIST

Pointer to the start of a variable argument list stored in a memory buffer.

Same as `UINT8 *`.

Definition at line 742 of file `Base.h`.

12.1.3.2 typedef CHAR8* VA_LIST

Variable used to traverse the list of arguments.

This type can vary by implementation and could be an array or structure.

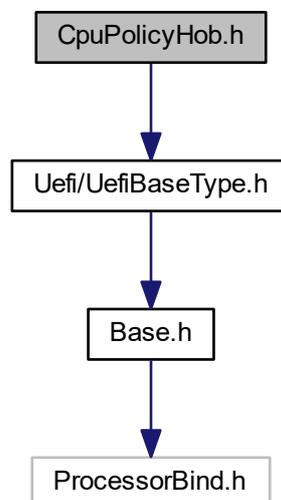
Definition at line 674 of file `Base.h`.

12.2 CpuPolicyHob.h File Reference

CPU Policy HOB.

```
#include <Uefi/UefiBaseType.h>
```

Include dependency graph for `CpuPolicyHob.h`:



Classes

- struct [CPU_POLICY_HOB](#)
CPU Initialization Policy Options.

12.2.1 Detailed Description

CPU Policy HOB.

Copyright

Copyright 2017 - 2021 Intel Corporation.

SPDX-License-Identifier: BSD-2-Clause-Patent

12.3 FspFixedPcds.h File Reference

This file lists all FixedAtBuild PCDs referenced in FSP integration guide.

Macros

- #define [PcdFspAreaBaseAddress](#) 0xFFD00000
FspAreaBaseAddress.
- #define [PcdFspImageldString](#) \$ICX-SP\$
FspImageldString.
- #define [PcdGlobalDataPointerAddress](#) 0xFED00148
GlobalDataPointerAddress.
- #define [PcdTemporaryRamBase](#) 0xFE800000
TemporaryRamBase.
- #define [PcdTemporaryRamSize](#) 0x00200000
TemporaryRamSize.
- #define [PcdFspReservedBufferSize](#) 0x00000100
FspReservedBufferSize.

12.3.1 Detailed Description

This file lists all FixedAtBuild PCDs referenced in FSP integration guide.

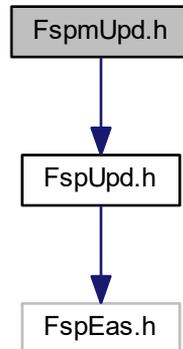
Those value may vary in different FSP revision to meet different requirements.

12.4 FspmUpd.h File Reference

Copyright (c) 2021, Intel Corporation.

```
#include <FspUpd.h>
```

Include dependency graph for FspmUpd.h:



Classes

- struct [FSPM_CONFIG](#)
FSP-M Configuration.
- struct [FSPM_UPD](#)
Fsp M UPD Configuration.

12.4.1 Detailed Description

Copyright (c) 2021, Intel Corporation.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

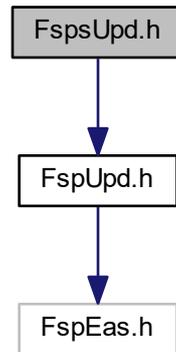
This file is automatically generated. Please do NOT modify !!!

12.5 FspUpd.h File Reference

Copyright (c) 2021, Intel Corporation.

```
#include <FspUpd.h>
```

Include dependency graph for FspUpd.h:



Classes

- struct [FSPS_CONFIG](#)
FSP-S Configuration.
- struct [FSPS_UPD](#)
Fsp S UPD Configuration.

12.5.1 Detailed Description

Copyright (c) 2021, Intel Corporation.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

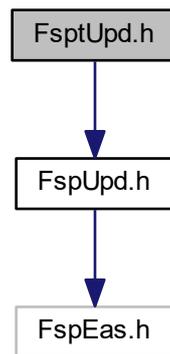
This file is automatically generated. Please do NOT modify !!!

12.6 FsptUpd.h File Reference

Copyright (c) 2021, Intel Corporation.

```
#include <FsptUpd.h>
```

Include dependency graph for FsptUpd.h:



Classes

- struct [FSPT_CORE_UPD](#)
FSP-T Core UPD.
- struct [FSPT_CONFIG](#)
FSP-T Configuration.
- struct [FSPT_UPD](#)
Fsp T UPD Configuration.

12.6.1 Detailed Description

Copyright (c) 2021, Intel Corporation.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

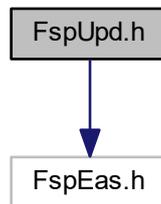
This file is automatically generated. Please do NOT modify !!!

12.7 FspUpd.h File Reference

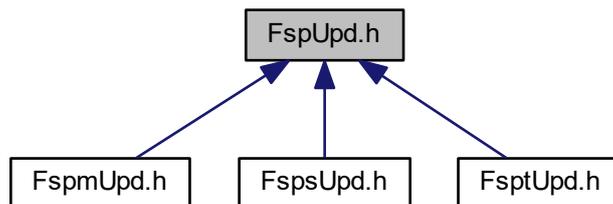
Copyright (c) 2021, Intel Corporation.

```
#include <FspEas.h>
```

Include dependency graph for FspUpd.h:



This graph shows which files directly or indirectly include this file:



12.7.1 Detailed Description

Copyright (c) 2021, Intel Corporation.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

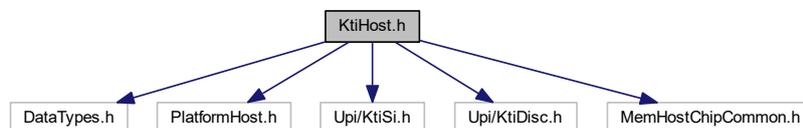
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This file is automatically generated. Please do NOT modify !!!

12.8 KtiHost.h File Reference

```
#include "DataTypes.h"
#include "PlatformHost.h"
#include <Upi/KtiSi.h>
#include <Upi/KtiDisc.h>
#include "MemHostChipCommon.h"
Include dependency graph for KtiHost.h:
```



Classes

- struct [PER_LANE_EPARAM_LINK_INFO](#)
Per Lane PHY Configuration.
- struct [ALL_LANES_EPARAM_LINK_INFO](#)
All Lanes PHY Configuration.
- struct [KTI_HOST_IN](#)
KTIRC input structure.

12.8.1 Detailed Description

Copyright

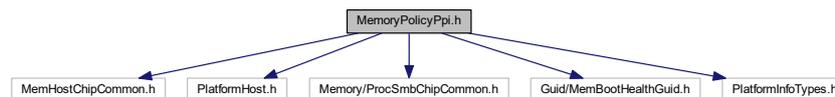
Copyright 2004 - 2021 Intel Corporation.

12.9 MemoryPolicyPpi.h File Reference

Header file defining MEMORY_POLICY_PPI, which is for platform code to set platform specific configurations of memory reference code.

```
#include <MemHostChipCommon.h>
#include <PlatformHost.h>
#include <Memory/ProcSmbChipCommon.h>
#include <Guid/MemBootHealthGuid.h>
#include <PlatformInfoTypes.h>
```

Include dependency graph for MemoryPolicyPpi.h:



Classes

- struct [memTiming](#)
Memory Timings Settings.
- struct [ddrDimmSetup](#)
DIMM enable/disable information.
- struct [ddrChannelSetup](#)
Channel setup structure declaration.
- struct [PPR_ADDR](#)
PPR DRAM Address.
- struct [PPR_ADDR_MRC_SETUP](#)
PPR Address, buffer to hold DRAM Address that need to be repaired.
- struct [ddrSocketSetup](#)
Socket setup structure declaration.
- union [AdvMemTestRankData](#)
Define AdvMemTest Rank List item The input format is defined as follows: Rank number in bits[3:0] DIMM number in bits[7:4] Channel number in the MC in bits[11:8] MC number in bits[15:12] Socket number in bits [19:16] bits [31:20] are reserved For example: To test MC 0, CH 1, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x100 To test MC 1, CH 0, DIMM 0, RANK 0 on Socket 0, you need to enter a value of 0x1000.
- struct [memSetup](#)
Host memory setup structure declaration.
- struct [commonSetup](#)
Common Platform Settings of MRC.
- struct [sysSetup](#)
Platform Setting for MRC.
- struct [_MEMORY_POLICY_PPI](#)
Memory Policy PPI Definition.

Macros

- #define [MAX_B2P_MAILBOX_GROUPS](#) 32
Number of group of BIOS-to-Pcode Mailbox command.
- #define [MEMORY_POLICY_PPI_REVISION](#) 0x00000001
Revision of MEMORY_POLICY_PPI.

Typedefs

- typedef struct [sysSetup](#) [SYS_SETUP](#)
Platform Setting for MRC.
- typedef struct [_MEMORY_POLICY_PPI](#) [MEMORY_POLICY_PPI](#)
Memory Policy PPI Definition.

12.9.1 Detailed Description

Header file defining [MEMORY_POLICY_PPI](#), which is for platform code to set platform specific configurations of memory reference code.

Copyright

Copyright 2018 - 2021 Intel Corporation.

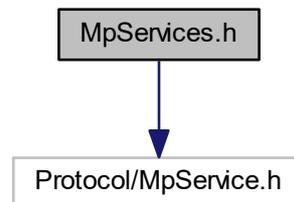
SPDX-License-Identifier: BSD-2-Clause-Patent

12.10 MpServices.h File Reference

This file declares UEFI PI Multi-processor PPI.

```
#include <Protocol/MpService.h>
```

Include dependency graph for MpServices.h:



Classes

- struct [_EFI_PEI_MP_SERVICES_PPI](#)
This PPI is installed by some platform or chipset-specific PEIM that abstracts handling multiprocessor support.

Typedefs

- typedef [EFI_STATUS](#)(* [EFI_PEI_MP_SERVICES_GET_NUMBER_OF_PROCESSORS](#)) ([IN CONST](#) [EFI_PEI_SERVICES](#) **PeiServices, [IN](#) [EFI_PEI_MP_SERVICES_PPI](#) *This, [OUT](#) [UINTN](#) *NumberOfProcessors, [OUT](#) [UINTN](#) *NumberOfEnabledProcessors)
Get the number of CPU's.
-

- typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_GET_PROCESSOR_INFO)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, IN `UINTN ProcessorNumber`, OUT `EFI_PROCESSOR_INFORMATION *ProcessorInfoBuffer`)
Get information on a specific CPU.
- typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_STARTUP_ALL_APS)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, IN `EFI_AP_PROCEDURE Procedure`, IN `BOOLEAN SingleThread`, IN `UINTN TimeoutInMicroSeconds`, IN `VOID *ProcedureArgument` OPTIONAL)
- typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_STARTUP_THIS_AP)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, IN `EFI_AP_PROCEDURE Procedure`, IN `UINTN ProcessorNumber`, IN `UINTN TimeoutInMicroseconds`, IN `VOID *ProcedureArgument` OPTIONAL)
Activate a specific application processor.
- typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_SWITCH_BSP)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, IN `UINTN ProcessorNumber`, IN `BOOLEAN EnableOldBSP`)
Switch the boot strap processor.
- typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_ENABLEDISABLEAP)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, IN `UINTN ProcessorNumber`, IN `BOOLEAN EnableAP`, IN `UINT32 *HealthFlag` OPTIONAL)
Enable or disable an application processor.
- typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_WHOAMI)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, OUT `UINTN *ProcessorNumber`)
Identify the currently executing processor.

12.10.1 Detailed Description

This file declares UEFI PI Multi-processor PPI.

This PPI is installed by some platform or chipset-specific PEIM that abstracts handling multiprocessor support.

Copyright (c) 2015 - 2017, Intel Corporation. All rights reserved.

SPDX-License-Identifier: BSD-2-Clause-Patent

Revision Reference:

This PPI is introduced in PI Version 1.4.

12.10.2 Typedef Documentation

- 12.10.2.1 typedef `EFI_STATUS(* EFI_PEI_MP_SERVICES_ENABLEDISABLEAP)` (IN CONST `EFI_PEI_SERVICES **PeiServices`, IN `EFI_PEI_MP_SERVICES_PPI *This`, IN `UINTN ProcessorNumber`, IN `BOOLEAN EnableAP`, IN `UINT32 *HealthFlag` OPTIONAL)

Enable or disable an application processor.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	A pointer to the <code>EFI_PEI_MP_SERVICES_PPI</code> instance.
in	<i>ProcessorNumber</i>	The handle number of the AP. The range is from 0 to the total number of logical processors minus 1. The total number of logical processors can be retrieved by <code>EFI_PEI_MP_SERVICES_PPI.GetNumberOfProcessors()</code> .

in	<i>EnableAP</i>	Specifies the new state for the processor for enabled, FALSE for disabled.
in	<i>HealthFlag</i>	If not NULL, a pointer to a value that specifies the new health status of the AP. This flag corresponds to StatusFlag defined in EFI_PEI_MP_SERVICE↔S_PPI.GetProcessorInfo(). Only the PROCESSOR_HEALTH_STATUS_BIT is used. All other bits are ignored. If it is NULL, this parameter is ignored.

Return values

<i>EFI_SUCCESS</i>	The specified AP was enabled or disabled successfully.
<i>EFI_UNSUPPORTED</i>	Enabling or disabling an AP cannot be completed prior to this service returning.
<i>EFI_UNSUPPORTED</i>	Enabling or disabling an AP is not supported.
<i>EFI_DEVICE_ERROR</i>	The calling processor is an AP.
<i>EFI_NOT_FOUND</i>	Processor with the handle specified by ProcessorNumber does not exist.
<i>EFI_INVALID_PARAMETER</i>	ProcessorNumber specifies the BSP.

Definition at line 230 of file MpServices.h.

12.10.2.2 `typedef EFI_STATUS(* EFI_PEI_MP_SERVICES_GET_NUMBER_OF_PROCESSORS) (IN CONST EFI_PEI_SERVICES **PeiServices, IN EFI_PEI_MP_SERVICES_PPI *This, OUT UINTN *NumberOfProcessors, OUT UINTN *NumberOfEnabledProcessors)`

Get the number of CPU's.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	Pointer to this instance of the PPI.
out	<i>NumberOfProcessors</i>	Pointer to the total number of logical processors in the system, including the BSP and disabled APs.
out	<i>NumberOfEnabledProcessors</i>	Number of processors in the system that are enabled.

Return values

<i>EFI_SUCCESS</i>	The number of logical processors and enabled logical processors was retrieved.
<i>EFI_DEVICE_ERROR</i>	The calling processor is an AP.
<i>EFI_INVALID_PARAMETER</i>	NumberOfProcessors is NULL. NumberOfEnabledProcessors is NULL.

Definition at line 45 of file MpServices.h.

12.10.2.3 `typedef EFI_STATUS(* EFI_PEI_MP_SERVICES_GET_PROCESSOR_INFO) (IN CONST EFI_PEI_SERVICES **PeiServices, IN EFI_PEI_MP_SERVICES_PPI *This, IN UINTN ProcessorNumber, OUT EFI_PROCESSOR_INFORMATION *ProcessorInfoBuffer)`

Get information on a specific CPU.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	Pointer to this instance of the PPI.
in	<i>ProcessorNumber</i>	Pointer to the total number of logical processors in the system, including the BSP and disabled APs.

out	<i>ProcessorInfo</i> ↔ <i>Buffer</i>	Number of processors in the system that are enabled.
-----	---	--

Return values

<i>EFI_SUCCESS</i>	Processor information was returned.
<i>EFI_DEVICE_ERROR</i>	The calling processor is an AP.
<i>EFI_INVALID_PARAMETER</i> ↔	ProcessorInfoBuffer is NULL.
<i>EFI_NOT_FOUND</i>	The processor with the handle specified by ProcessorNumber does not exist in the platform.

Definition at line 70 of file MpServices.h.

12.10.2.4 **typedef EFI_STATUS(* EFI_PEI_MP_SERVICES_STARTUP_ALL_APS) (IN CONST EFI_PEI_SERVICES **PeiServices, IN EFI_PEI_MP_SERVICES_PPI *This, IN EFI_AP_PROCEDURE Procedure, IN BOOLEAN SingleThread, IN UINTN TimeoutInMicroSeconds, IN VOID *ProcedureArgument OPTIONAL)**

Activate all of the application processors.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	A pointer to the EFI_PEI_MP_SERVICES_PPI instance.
in	<i>Procedure</i>	A pointer to the function to be run on enabled APs of the system.
in	<i>SingleThread</i>	If TRUE, then all the enabled APs execute the function specified by Procedure one by one, in ascending order of processor handle number. If FALSE, then all the enabled APs execute the function specified by Procedure simultaneously.
in	<i>TimeoutIn</i> ↔ <i>MicroSeconds</i>	Indicates the time limit in microseconds for APs to return from Procedure, for blocking mode only. Zero means infinity. If the timeout expires before all APs return from Procedure, then Procedure on the failed APs is terminated. All enabled APs are available for next function assigned by EFI_PEI_MP_SERVICES_PPI.StartupAllAPs() or EFI_PEI_MP_SERVICES_PPI.StartupThisAP(). If the timeout expires in blocking mode, BSP returns EFI_TIMEOUT.
in	<i>Procedure</i> ↔ <i>Argument</i>	The parameter passed into Procedure for all APs.

Return values

<i>EFI_SUCCESS</i>	In blocking mode, all APs have finished before the timeout expired.
<i>EFI_DEVICE_ERROR</i>	Caller processor is AP.
<i>EFI_NOT_STARTED</i>	No enabled APs exist in the system.
<i>EFI_NOT_READY</i>	Any enabled APs are busy.
<i>EFI_TIMEOUT</i>	In blocking mode, the timeout expired before all enabled APs have finished.
<i>EFI_INVALID_PARAMETER</i> ↔ <i>ER</i>	Procedure is NULL.

Definition at line 113 of file MpServices.h.

12.10.2.5 **typedef EFI_STATUS(* EFI_PEI_MP_SERVICES_STARTUP_THIS_AP) (IN CONST EFI_PEI_SERVICES **PeiServices, IN EFI_PEI_MP_SERVICES_PPI *This, IN EFI_AP_PROCEDURE Procedure, IN UINTN ProcessorNumber, IN UINTN TimeoutInMicroSeconds, IN VOID *ProcedureArgument OPTIONAL)**

Activate a specific application processor.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	A pointer to the EFI_PEI_MP_SERVICES_PPI instance.
in	<i>Procedure</i>	A pointer to the function to be run on enabled APs of the system.
in	<i>Processor↔ Number</i>	The handle number of the AP. The range is from 0 to the total number of logical processors minus 1. The total number of logical processors can be retrieved by EFI_PEI_MP_SERVICES_PPI.GetNumberOfProcessors().
in	<i>TimeoutIn↔ MicroSeconds</i>	Indicates the time limit in microseconds for APs to return from Procedure, for blocking mode only. Zero means infinity. If the timeout expires before all A↔Ps return from Procedure, then Procedure on the failed APs is terminated. All enabled APs are available for next function assigned by EFI_PEI_MP_SERV↔ICES_PPI.StartupAllAPs() or EFI_PEI_MP_SERVICES_PPI.StartupThisAP(). If the timeout expires in blocking mode, BSP returns EFI_TIMEOUT.
in	<i>Procedure↔ Argument</i>	The parameter passed into Procedure for all APs.

Return values

<i>EFI_SUCCESS</i>	In blocking mode, specified AP finished before the timeout expires.
<i>EFI_DEVICE_ERROR</i>	The calling processor is an AP.
<i>EFI_TIMEOUT</i>	In blocking mode, the timeout expired before the specified AP has finished.
<i>EFI_NOT_FOUND</i>	The processor with the handle specified by ProcessorNumber does not exist.
<i>EFI_INVALID_PARAMET↔ ER</i>	ProcessorNumber specifies the BSP or disabled AP.
<i>EFI_INVALID_PARAMET↔ ER</i>	Procedure is NULL.

Definition at line 158 of file MpServices.h.

12.10.2.6 typedef EFI_STATUS(* EFI_PEI_MP_SERVICES_SWITCH_BSP) (IN CONST EFI_PEI_SERVICES **PeiServices, IN EFI_PEI_MP_SERVICES_PPI *This, IN UINTN ProcessorNumber, IN BOOLEAN EnableOldBSP)

Switch the boot strap processor.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	A pointer to the EFI_PEI_MP_SERVICES_PPI instance.
in	<i>Processor↔ Number</i>	The handle number of the AP. The range is from 0 to the total number of logical processors minus 1. The total number of logical processors can be retrieved by EFI_PEI_MP_SERVICES_PPI.GetNumberOfProcessors().
in	<i>EnableOldBSP</i>	If TRUE, then the old BSP will be listed as an enabled AP. Otherwise, it will be disabled.

Return values

<i>EFI_SUCCESS</i>	BSP successfully switched.
<i>EFI_UNSUPPORTED</i>	Switching the BSP cannot be completed prior to this service returning.
<i>EFI_UNSUPPORTED</i>	Switching the BSP is not supported.
<i>EFI_DEVICE_ERROR</i>	The calling processor is an AP.
<i>EFI_NOT_FOUND</i>	The processor with the handle specified by ProcessorNumber does not exist.
<i>EFI_INVALID_PARAMET↔ ER</i>	ProcessorNumber specifies the current BSP or a disabled AP.

<code>EFI_NOT_READY</code>	The specified AP is busy.
----------------------------	---------------------------

Definition at line 193 of file MpServices.h.

12.10.2.7 `typedef EFI_STATUS(* EFI_PEI_MP_SERVICES_WHOAMI) (IN CONST EFI_PEI_SERVICES **PeiServices, IN EFI_PEI_MP_SERVICES_PPI *This, OUT UINTN *ProcessorNumber)`

Identify the currently executing processor.

Parameters

in	<i>PeiServices</i>	An indirect pointer to the PEI Services Table published by the PEI Foundation.
in	<i>This</i>	A pointer to the <code>EFI_PEI_MP_SERVICES_PPI</code> instance.
out	<i>ProcessorNumber</i>	The handle number of the AP. The range is from 0 to the total number of logical processors minus 1. The total number of logical processors can be retrieved by <code>EFI_PEI_MP_SERVICES_PPI.GetNumberOfProcessors()</code> .

Return values

<code>EFI_SUCCESS</code>	The current processor handle number was returned in <code>ProcessorNumber</code> .
<code>EFI_INVALID_PARAMETER</code>	<code>ProcessorNumber</code> is NULL.

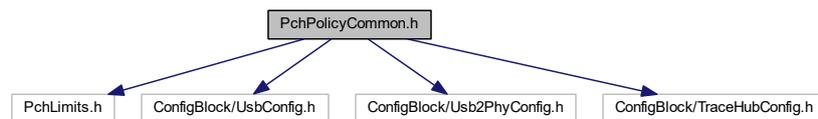
Definition at line 255 of file MpServices.h.

12.11 PchPolicyCommon.h File Reference

PCH configuration based on PCH policy.

```
#include "PchLimits.h"
#include <ConfigBlock/UsbConfig.h>
#include <ConfigBlock/Usb2PhyConfig.h>
#include <ConfigBlock/TraceHubConfig.h>
```

Include dependency graph for PchPolicyCommon.h:



Classes

- struct [PCH_GENERAL_CONFIG](#)
PCH General Configuration.
- struct [PCH_PCIE_EQ_LANE_PARAM](#)
Represent lane specific PCIe Gen3 equalization parameters.
- struct [PCH_PCIE_ROOT_PORT_CONFIG](#)
The PCH_PCI_EXPRESS_ROOT_PORT_CONFIG describe the feature and capability of each PCH PCIe root port.
- struct [PCH_PCIE_CONFIG](#)
The PCH_PCIE_CONFIG block describes the expected configuration of the PCH PCI Express controllers.
- struct [PCH_PCIE_CONFIG2](#)

- The *PCH_PCIE_CONFIG2* block describes the additional configuration of the PCH PCI Express controllers.
- struct [PCH_HSIO_PCIE_LANE_CONFIG](#)
 - The *PCH_HSIO_PCIE_LANE_CONFIG* describes HSIO settings for PCIe lane.
- struct [PCH_HSIO_PCIE_CONFIG](#)
 - The *PCH_HSIO_PCIE_CONFIG* block describes the configuration of the HSIO for PCIe lanes.
- struct [PCH_HSIO_PCIE_WM20_CONFIG](#)
 - The *PCH_HSIO_PCIE_WM20_CONFIG* block describes the configuration of the HSIO for PCIe lanes.
- struct [PCH_SATA_PORT_CONFIG](#)
 - This structure configures the features, property, and capability for each SATA port.
- struct [PCH_SATA_RST_CONFIG](#)
 - Rapid Storage Technology settings.
- struct [PCH_RST_PCIE_STORAGE_CONFIG](#)
 - This structure describes the details of Intel RST for PCIe Storage remapping Note: In order to use this feature, Intel RST Driver is required.
- struct [PCH_SATA_CONFIG](#)
 - The *PCH_SATA_CONFIG* block describes the expected configuration of the SATA controllers.
- struct [PCH_HSIO_SATA_PORT_LANE](#)
 - The *PCH_HSIO_SATA_PORT_LANE* describes HSIO settings for SATA Port lane.
- struct [PCH_HSIO_SATA_CONFIG](#)
 - The *PCH_HSIO_SATA_CONFIG* block describes the HSIO configuration of the SATA controller.
- struct [PCH_IOAPIC_CONFIG](#)
 - The *PCH_IOAPIC_CONFIG* block describes the expected configuration of the PCH IO APIC, it's optional and PCH code would ignore it if the BdfValid bit is not TRUE.
- struct [PCH_HPET_CONFIG](#)
 - The *PCH_HPET_CONFIG* block passes the bus/device/function value for HPET.
- struct [PCH_HDAUDIO_CONFIG](#)
 - This structure contains the policies which are related to HD Audio device (cAVS).
- struct [PCH_LAN_CONFIG](#)
 - PCH intergrated LAN controller configuration settings.
- struct [PCH_SMBUS_CONFIG](#)
 - The *SMBUS_CONFIG* block lists the reserved addresses for non-ARP capable devices in the platform.
- struct [PCH_LOCK_DOWN_CONFIG](#)
 - The *PCH_LOCK_DOWN_CONFIG* block describes the expected configuration of the PCH for security requirement.
- struct [THERMAL_THROTTLE_LEVELS](#)
 - This structure lists PCH supported throttling register setting for customization.
- struct [DMI_HW_WIDTH_CONTROL](#)
 - This structure allows to customize DMI HW Autonomous Width Control for Thermal and Mechanical spec design.
- struct [SATA_THERMAL_THROTTLE](#)
 - This structure lists PCH supported SATA thermal throttling register setting for customization.
- struct [PCH_THERMAL_THROTTLING](#)
 - This structure decides the settings of PCH Thermal throttling.
- struct [TS_GPIO_PIN_SETTING](#)
 - This structure configures PCH memory throttling thermal sensor GPIO PIN settings.
- struct [PCH_MEMORY_THROTTLING](#)
 - This structure supports an external memory thermal sensor (TS-on-DIMM or TS-on-Board).
- struct [PCH_THERMAL_CONFIG](#)
 - The *PCH_THERMAL_CONFIG* block describes the expected configuration of the PCH for Thermal.
- struct [PCH_POWER_RESET_STATUS](#)
 - This *PCH_POWER_RESET_STATUS* Specifies which Power/Reset bits need to be cleared by the PCH Init Driver.
- union [PCH_GBL2HOST_EN](#)
 - This *PCH_GBL2HOST_EN* specifes enable bits related to the "Convert Global Resets to Host Resets" (G2H) feature.

- struct [PCH_WAKE_CONFIG](#)
This structure allows to customize PCH wake up capability from S5 or DeepSx by WOL, LAN, PCIE wake events.
 - struct [PCH_PM_CONFIG](#)
The [PCH_PM_CONFIG](#) block describes expected miscellaneous power management settings.
 - struct [PCH_DMI_CONFIG](#)
The [PCH_DMI_CONFIG](#) block describes the expected configuration of the PCH for DMI.
 - struct [PCH_LPC_SIRQ_CONFIG](#)
The [PCH_LPC_SIRQ_CONFIG](#) block describes the expected configuration of the PCH for Serial IRQ.
 - struct [PCH_PORT61H_SMM_CONFIG](#)
This structure is used for the emulation feature for Port61h read.
 - struct [PCH_DEVICE_INTERRUPT_CONFIG](#)
The [PCH_DEVICE_INTERRUPT_CONFIG](#) block describes interrupt pin, IRQ and interrupt mode for PCH device.
 - struct [PCH_INTERRUPT_CONFIG](#)
The [PCH_INTERRUPT_CONFIG](#) block describes interrupt settings for PCH.
 - struct [PCH_TRACE_HUB_CONFIG](#)
The [PCH_TRACE_HUB_CONFIG](#) block describes TraceHub settings for PCH.
 - struct [PCH_SKYCAM_CIO2_FLS_CONFIG](#)
The [PCH_SKYCAM_CIO2_FLS_CONFIG](#) block describes SkyCam CIO2 FLS registers configuration.
 - struct [PCH_USB20_PORT_CONFIG](#)
This structure configures per USB2 port physical settings.
 - struct [PCH_USB30_PORT_CONFIG](#)
This structure describes whether the USB3 Port N of PCH is enabled by platform modules.
 - struct [PCH_XHCI_SSIIC_PORT](#)
These members describe some settings which are related to the SSIIC ports.
 - struct [PCH_SSIIC_CONFIG](#)
These members describe some settings which are related to the SSIIC ports.
 - struct [PCH_XDCI_CONFIG](#)
The [PCH_XDCI_CONFIG](#) block describes the configurations of the xDCI Usb Device controller.
 - struct [PCH_USB_CONFIG](#)
This member describes the expected configuration of the PCH USB controllers, Platform modules may need to refer Setup options, schematic, BIOS specification to update this field.
 - struct [PROTECTED_RANGE](#)
The PCH provides a method for blocking writes and reads to specific ranges in the SPI flash when the Protected Ranges are enabled.
 - struct [PCH_FLASH_PROTECTION_CONFIG](#)
PCH Flash Protection Configuration.
 - struct [PCH_WDT_CONFIG](#)
This policy clears status bits and disable watchdog, then lock the WDT registers.
 - struct [PCH_P2SB_CONFIG](#)
This structure contains the policies which are related to P2SB device.
 - struct [PCH_DCI_CONFIG](#)
This structure contains the policies which are related to Direct Connection Interface (DCI).
 - struct [PCH_LPC_CONFIG](#)
This structure contains the policies which are related to LPC.
 - struct [PCH_SPI_CONFIG](#)
This structure contains the policies which are related to SPI.
 - struct [_PCH_POLICY](#)
The PCH Policy allows the platform code to publish a set of configuration information that the PCH drivers will use to configure the PCH hardware.
-

Macros

- #define `PCH_HDAUDIO_AUTO` 2
The `PCH_HDAUDIO_CONFIG` block describes the expected configuration of the Intel HD Audio feature.
- #define `PCH_MAX_DEVICE_INTERRUPT_CONFIG` 64
Number of all PCH devices.
- #define `PCH_MAX_PXRC_CONFIG` 8
Number of PXRC registers in ITSS.
- #define `PCH_POLICY_REVISION` 15
PCH Policy revision number Any backwards compatible changes to this structure will result in an update in the revision number.

Enumerations

- enum `PCH_RESERVED_PAGE_ROUTE`
- enum `PCH_PCIE_ASPM_CONTROL`
The values before AutoConfig match the setting of PCI Express Base Specification 1.1, please be careful for adding new feature.
- enum `PCH_PCIE_L1SUBSTATES_CONTROL`
Refer to PCH EDS for the PCH implementation values corresponding to below PCI-E spec defined ranges.
- enum `PCH_PCIE_EQ_METHOD`
- enum `PCH_HDAUDIO_IO_BUFFER_OWNERSHIP`
- enum `PCH_SLP_S4_MIN_ASSERT`
- enum `PCH_START_FRAME_PULSE`
Refer to PCH EDS for the details of Start Frame Pulse Width in Continuous and Quiet mode.
- enum `PCH_INT_PIN`
- enum `PCH_USB_PORT_LOCATION`
The location of the USB connectors.

12.11.1 Detailed Description

PCH configuration based on PCH policy.

Copyright

Copyright 2009 - 2021 Intel Corporation.

SPDX-License-Identifier: BSD-2-Clause-Patent

12.11.2 Enumeration Type Documentation

12.11.2.1 enum `PCH_HDAUDIO_IO_BUFFER_OWNERSHIP`

Enumerator

PchHdaloBufOwnerHdaLink HD-Audio link owns all the I/O buffers.

PchHdaloBufOwnerHdaLinkI2sPort HD-Audio link owns 4 and I2S port owns 4 of the I/O buffers.

PchHdaloBufOwnerI2sPort I2S0 and I2S1 ports own all the I/O buffers.

Definition at line 744 of file PchPolicyCommon.h.

12.11.2.2 enum PCH_INT_PIN

Enumerator

PchNoInt No Interrupt Pin.

Definition at line 1404 of file PchPolicyCommon.h.

12.11.2.3 enum PCH_PCIE_EQ_METHOD

Enumerator

PchPcieEqDefault Use reference code default (software margining)

PchPcieEqHardware Hardware equalization (experimental), note this requires PCH-LP C0 or PCH-H D0 or newer.

PchPcieEqSoftware Use software margining flow.

PchPcieEqStaticCoeff Fixed equalization (requires Coefficient settings per lane)

Definition at line 123 of file PchPolicyCommon.h.

12.11.2.4 enum PCH_RESERVED_PAGE_ROUTE

Enumerator

PchReservedPageToLpc Port 80h cycles are sent to LPC.

PchReservedPageToPcie Port 80h cycles are sent to PCIe.

Definition at line 58 of file PchPolicyCommon.h.

12.11.2.5 enum PCH_SLP_S4_MIN_ASSERT

Enumerator

PchSlpS4PchTime The time defined in PCH EDS Power Sequencing and Reset Signal Timings table.

Definition at line 1191 of file PchPolicyCommon.h.

12.11.2.6 enum PCH_USB_PORT_LOCATION

The location of the USB connectors.

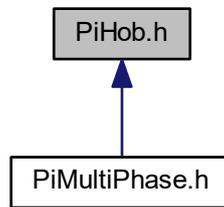
This information is use to decide eye diagram tuning value for Usb 2.0 motherboard trace.

Definition at line 1495 of file PchPolicyCommon.h.

12.12 PiHob.h File Reference

HOB related definitions in PI.

This graph shows which files directly or indirectly include this file:



Classes

- struct [EFI_HOB_GENERIC_HEADER](#)
Describes the format and size of the data inside the HOB.
- struct [EFI_HOB_HANDOFF_INFO_TABLE](#)
Contains general state information used by the HOB producer phase.
- struct [EFI_HOB_MEMORY_ALLOCATION_HEADER](#)
EFI_HOB_MEMORY_ALLOCATION_HEADER describes the various attributes of the logical memory allocation.
- struct [EFI_HOB_MEMORY_ALLOCATION](#)
Describes all memory ranges used during the HOB producer phase that exist outside the HOB list.
- struct [EFI_HOB_MEMORY_ALLOCATION_STACK](#)
Describes the memory stack that is produced by the HOB producer phase and upon which all post-memory-installed executable content in the HOB producer phase is executing.
- struct [EFI_HOB_MEMORY_ALLOCATION_BSP_STORE](#)
Defines the location of the boot-strap processor (BSP) BSPStore ("Backing Store Pointer Store").
- struct [EFI_HOB_MEMORY_ALLOCATION_MODULE](#)
Defines the location and entry point of the HOB consumer phase.
- struct [EFI_HOB_RESOURCE_DESCRIPTOR](#)
Describes the resource properties of all fixed, nonrelocatable resource ranges found on the processor host bus during the HOB producer phase.
- struct [EFI_HOB_GUID_TYPE](#)
Allows writers of executable content in the HOB producer phase to maintain and manage HOBs with specific GUID.
- struct [EFI_HOB_FIRMWARE_VOLUME](#)
Details the location of firmware volumes that contain firmware files.
- struct [EFI_HOB_FIRMWARE_VOLUME2](#)
Details the location of a firmware volume that was extracted from a file within another firmware volume.
- struct [EFI_HOB_FIRMWARE_VOLUME3](#)
Details the location of a firmware volume that was extracted from a file within another firmware volume.
- struct [EFI_HOB_CPU](#)
Describes processor information, such as address space and I/O space capabilities.
- struct [EFI_HOB_MEMORY_POOL](#)
Describes pool memory allocations.
- struct [EFI_HOB_UEFI_CAPSULE](#)
Each UEFI capsule HOB details the location of a UEFI capsule.
- union [EFI_PEI_HOB_POINTERS](#)
Union of all the possible HOB Types.

Macros

- `#define EFI_HOB_HANDOFF_TABLE_VERSION 0x0009`
Value of version in `EFI_HOB_HANDOFF_INFO_TABLE`.

Typedefs

- typedef UINT32 `EFI_RESOURCE_TYPE`
The resource type.
- typedef UINT32 `EFI_RESOURCE_ATTRIBUTE_TYPE`
A type of recount attribute type.

12.12.1 Detailed Description

HOB related definitions in PI.

Copyright (c) 2006 - 2017, Intel Corporation. All rights reserved.
 SPDX-License-Identifier: BSD-2-Clause-Patent

Revision Reference:

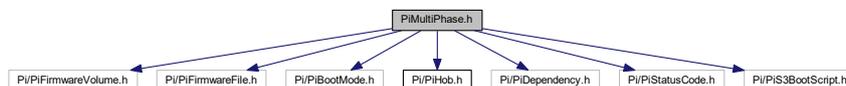
PI Version 1.6

12.13 PiMultiPhase.h File Reference

Include file matches things in PI for multiple module types.

```
#include <Pi/PiFirmwareVolume.h>
#include <Pi/PiFirmwareFile.h>
#include <Pi/PiBootMode.h>
#include <Pi/PiHob.h>
#include <Pi/PiDependency.h>
#include <Pi/PiStatusCode.h>
#include <Pi/PiS3BootScript.h>
```

Include dependency graph for PiMultiPhase.h:



Classes

- struct `EFI_MMRAM_DESCRIPTOR`
Structure describing a MMRAM region and its accessibility attributes.
- struct `_EFI_MM_RESERVED_MMRAM_REGION`
Structure describing a MMRAM region which cannot be used for the MMRAM heap.

Macros

- #define `DXE_ERROR`(StatusCode) (MAX_BIT | (MAX_BIT >> 2) | StatusCode)
Produces an error code in the range reserved for use by the Platform Initialization Architecture Specification.
- #define `EFI_REQUEST_UNLOAD_IMAGE_DXE_ERROR` (1)
If this value is returned by an EFI image, then the image should be unloaded.
- #define `EFI_NOT_AVAILABLE_YET_DXE_ERROR` (2)
If this value is returned by an API, it means the capability is not yet installed/available/ready to use.
- #define `PI_ENCODE_WARNING`(a) ((MAX_BIT >> 2) | (a))
Success and warning codes reserved for use by PI.
- #define `PI_ENCODE_ERROR`(a) (MAX_BIT | (MAX_BIT >> 2) | (a))
Error codes reserved for use by PI.
- #define `EFI_INTERRUPT_PENDING_PI_ENCODE_ERROR` (0)
Return status codes defined in SMM CIS.
- #define `EFI_MMRAM_OPEN` 0x00000001
MMRAM states and capabilities.

- #define `EFI_AUTH_STATUS_PLATFORM_OVERRIDE` 0x01
Bitmask of values for Authentication Status.

Typedefs

- typedef struct `_EFI_MM_RESERVED_MMRAM_REGION` `EFI_MM_RESERVED_MMRAM_REGION`
Structure describing a MMRAM region which cannot be used for the MMRAM heap.
- typedef `VOID`(* `EFI_AP_PROCEDURE`) (`IN OUT VOID` *Buffer)
The function prototype for invoking a function on an Application Processor.
- typedef `EFI_STATUS`(* `EFI_AP_PROCEDURE2`) (`IN VOID` *ProcedureArgument)
The function prototype for invoking a function on an Application Processor.

12.13.1 Detailed Description

Include file matches things in PI for multiple module types.

Copyright (c) 2006 - 2018, Intel Corporation. All rights reserved.

SPDX-License-Identifier: BSD-2-Clause-Patent

Revision Reference:

These elements are defined in UEFI Platform Initialization Specification 1.2.

12.13.2 Macro Definition Documentation

12.13.2.1 #define `DXE_ERROR`(*StatusCode*) (MAX_BIT | (MAX_BIT >> 2) | *StatusCode*)

Produces an error code in the range reserved for use by the Platform Initialization Architecture Specification.

The supported 32-bit range is 0xA0000000-0xBFFFFFFF The supported 64-bit range is 0xA000000000000000-0xBFFFFFFF00000000

Parameters

<i>StatusCode</i>	The status code value to convert into a warning code. StatusCode must be in the range 0x00000000..0x1FFFFFFF.
-------------------	---

Returns

The value specified by StatusCode in the PI reserved range.

Definition at line 36 of file PiMultiPhase.h.

12.13.2.2 #define EFI_AUTH_STATUS_PLATFORM_OVERRIDE 0x01

Bitmask of values for Authentication Status.

Authentication Status is returned from EFI_GUIDED_SECTION_EXTRACTION_PROTOCOL and the EFI_PEI_↔ GUIDED_SECTION_EXTRACTION_PPI

xx00 Image was not signed. xxx1 Platform security policy override. Assumes the same meaning as 0010 (the image was signed, the signature was tested, and the signature passed authentication test). 0010 Image was signed, the signature was tested, and the signature passed authentication test. 0110 Image was signed and the signature was not tested. 1010 Image was signed, the signature was tested, and the signature failed the authentication test.

Definition at line 84 of file PiMultiPhase.h.

12.13.2.3 #define PI_ENCODE_ERROR(a)(MAX_BIT | (MAX_BIT >> 2) | (a))

Error codes reserved for use by PI.

Supported 32-bit range is 0xa0000000-0xbfffffff. Supported 64-bit range is 0xa000000000000000-0xbfffffffffffffff.

Definition at line 61 of file PiMultiPhase.h.

12.13.2.4 #define PI_ENCODE_WARNING(a)((MAX_BIT >>> 2) | (a))

Success and warning codes reserved for use by PI.

Supported 32-bit range is 0x20000000-0x3fffffff. Supported 64-bit range is 0x2000000000000000-0x3fffffffffffffff.

Definition at line 54 of file PiMultiPhase.h.

12.13.3 Typedef Documentation**12.13.3.1 typedef VOID(* EFI_AP_PROCEDURE) (IN OUT VOID *Buffer)**

The function prototype for invoking a function on an Application Processor.

This definition is used by the UEFI MP Serices Protocol, and the PI SMM System Table.

Parameters

<i>in, out</i>	<i>Buffer</i>	The pointer to private data buffer.
----------------	---------------	-------------------------------------

Definition at line 191 of file PiMultiPhase.h.

12.13.3.2 typedef EFI_STATUS(* EFI_AP_PROCEDURE2) (IN VOID *ProcedureArgument)

The function prototype for invoking a function on an Application Processor.

This definition is used by the UEFI MM MP Serices Protocol.

Parameters

<code>in</code>	<i>Procedure↔ Argument</i>	The pointer to private data buffer.
-----------------	--------------------------------	-------------------------------------

Return values

<code>EFI_SUCCESS</code>	Excutive the procedure successfully
--------------------------	-------------------------------------

Definition at line 207 of file PiMultiPhase.h.

12.14 PsmiPolicyHob.h File Reference

PSMI Policy HOB.

Classes

- struct [TRACE_INFO](#)
Trace Info.
- struct [PSMI_POLICY_DATA_HOB](#)
PSMI policy.

12.14.1 Detailed Description

PSMI Policy HOB.

Copyright

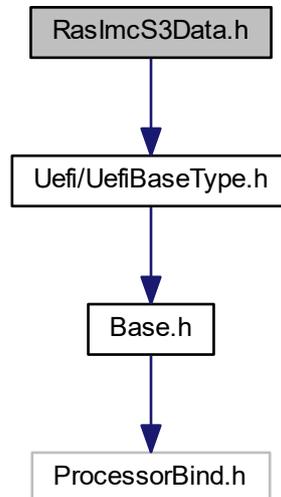
Copyright 2018 - 2021 Intel Corporation.

SPDX-License-Identifier: BSD-2-Clause-Patent

12.15 RasImcS3Data.h File Reference

RAS IMC S3 Data Load PPI.

```
#include <Uefi/UefiBaseType.h>
Include dependency graph for RasImcS3Data.h:
```



Classes

- struct [_RAS_IMC_S3_DATA_PPI](#)
RAS IMC S3 Data PPI.

Typedefs

- typedef [EFI_STATUS](#)(* [RAS_IMC_S3_DATA_PPI_GET_IMC_S3_RAS_DATA](#)) (IN CONST [RAS_IMC_S3_DATA_PPI](#) *This, IN OUT UINT32 *DataSize, OUT VOID *Data)
Retrieves data for S3 saved memory RAS features from non-volatile storage.

12.15.1 Detailed Description

RAS IMC S3 Data Load PPI.

Copyright

Copyright 2021 Intel Corporation.

SPDX-License-Identifier: BSD-2-Clause-Patent

12.15.2 Typedef Documentation

- 12.15.2.1 typedef [EFI_STATUS](#)(* [RAS_IMC_S3_DATA_PPI_GET_IMC_S3_RAS_DATA](#)) (IN CONST [RAS_IMC_S3_DATA_PPI](#) *This, IN OUT UINT32 *DataSize, OUT VOID *Data)

Retrieves data for S3 saved memory RAS features from non-volatile storage.

If the Data buffer is too small to hold the contents of the NVS data, the error `EFI_BUFFER_TOO_SMALL` is returned and `DataSize` is set to the required buffer size to obtain the data.

Parameters

in	<i>This</i>	A pointer to this instance of the RAS_IMC_S3_DATA_PPI.
in, out	<i>DataSize</i>	On entry, points to the size in bytes of the Data buffer. On return, points to the size of the data returned in Data.
out	<i>Data</i>	Points to the buffer which will hold the returned data.

Return values

<i>EFI_SUCCESS</i>	The NVS data was read successfully.
<i>EFI_NOT_FOUND</i>	The NVS data does not exist.
<i>EFI_BUFFER_TOO_SMALL</i>	The DataSize is too small for the NVS data. DataSize is updated with the size required for the NVS data.
<i>EFI_INVALID_PARAMETER</i>	DataSize or Data is NULL.
<i>EFI_DEVICE_ERROR</i>	The NVS data could not be retrieved because of a device error.
<i>EFI_UNSUPPORTED</i>	This platform does not support the save/restore of S3 memory data

Definition at line 41 of file RasImcS3Data.h.

12.16 RasRcPolicyPpi.h File Reference

RAS Policy PPI header file.

Classes

- struct [RAS_RC_POLICY_PPI](#)

RAS policy being requested of RC.

12.16.1 Detailed Description

RAS Policy PPI header file.

Copyright

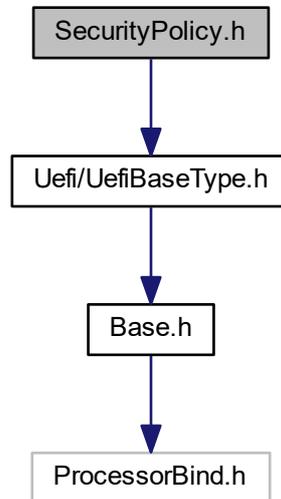
Copyright 2018 - 2021 Intel Corporation.

SPDX-License-Identifier: BSD-2-Clause-Patent

12.17 SecurityPolicy.h File Reference

Provides data structure information used by ServerSecurity features in Mtkme etc.

```
#include <Uefi/UefiBaseType.h>  
Include dependency graph for SecurityPolicy.h:
```



Classes

- struct [SECURITY_POLICY](#)

Security Policy.

12.17.1 Detailed Description

Provides data structure information used by ServerSecurity features in Mtkme etc.

Copyright

Copyright 2018 - 2021 Intel Corporation.

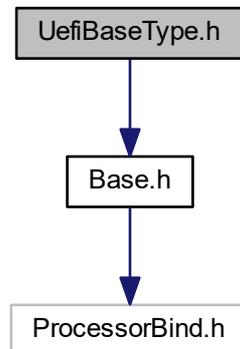
SPDX-License-Identifier: BSD-2-Clause-Patent

12.18 UefiBaseType.h File Reference

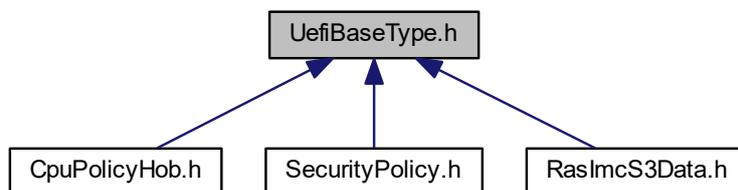
Defines data types and constants introduced in UEFI.

```
#include <Base.h>
```

Include dependency graph for UefiBaseType.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [EFI_TIME](#)
EFI Time Abstraction: Year: 1900 - 9999 Month: 1 - 12 Day: 1 - 31 Hour: 0 - 23 Minute: 0 - 59 Second: 0 - 59 Nanosecond: 0 - 999,999,999 TimeZone: -1440 to 1440 or 2047.
- struct [EFI_MAC_ADDRESS](#)
32-byte buffer containing a network Media Access Control address.
- union [EFI_IP_ADDRESS](#)
16-byte buffer aligned on a 4-byte boundary.

Macros

- #define [EFIERR\(_a\) ENCODE_ERROR\(_a\)](#)
Define macro to encode the status code.
- #define [EFI_SIZE_TO_PAGES\(Size\) \(\(\(Size\) >> EFI_PAGE_SHIFT\) + \(\(\(Size\) & EFI_PAGE_MASK\) ? 1 : 0\)\)](#)

- Macro that converts a size, in bytes, to a number of EFI_PAGESs.*

 - #define [EFI_PAGES_TO_SIZE](#)(Pages) ((Pages) << EFI_PAGE_SHIFT)
- Macro that converts a number of EFI_PAGES to a size in bytes.*

 - #define [EFI_IMAGE_MACHINE_IA32](#) 0x014C
 - PE32+ Machine type for IA32 UEFI images.*
 - #define [EFI_IMAGE_MACHINE_IA64](#) 0x0200
 - PE32+ Machine type for IA64 UEFI images.*
 - #define [EFI_IMAGE_MACHINE_EBC](#) 0x0EBC
 - PE32+ Machine type for EBC UEFI images.*
 - #define [EFI_IMAGE_MACHINE_X64](#) 0x8664
 - PE32+ Machine type for X64 UEFI images.*
 - #define [EFI_IMAGE_MACHINE_ARMTHUMB_MIXED](#) 0x01C2
 - PE32+ Machine type for ARM mixed ARM and Thumb/Thumb2 images.*
 - #define [EFI_IMAGE_MACHINE_AARCH64](#) 0xAA64
 - PE32+ Machine type for AARCH64 A64 images.*
 - #define [EFI_IMAGE_MACHINE_RISCV32](#) 0x5032
 - PE32+ Machine type for RISC-V 32/64/128.*
- #define [EFI_SUCCESS](#) RETURN_SUCCESS
 - Enumeration of EFI_STATUS.*
- #define [EFI_NETWORK_UNREACHABLE](#) EFIERR(100)
 - ICMP error definitions.*
- #define [EFI_CONNECTION_FIN](#) EFIERR(104)
 - Tcp connection status definitions.*

Typedefs

- typedef [GUID](#) [EFI_GUID](#)
 - 128-bit buffer containing a unique identifier value.*
- typedef RETURN_STATUS [EFI_STATUS](#)
 - Function return status for EFI API.*
- typedef VOID * [EFI_HANDLE](#)
 - A collection of related interfaces.*
- typedef VOID * [EFI_EVENT](#)
 - Handle to an event structure.*
- typedef UINTN [EFI_TPL](#)
 - Task priority level.*
- typedef UINT64 [EFI_LBA](#)
 - Logical block address.*
- typedef UINT64 [EFI_PHYSICAL_ADDRESS](#)
 - 64-bit physical memory address.*
- typedef UINT64 [EFI_VIRTUAL_ADDRESS](#)
 - 64-bit virtual memory address.*
- typedef IPv4_ADDRESS [EFI_IPv4_ADDRESS](#)
 - 4-byte buffer.*
- typedef IPv6_ADDRESS [EFI_IPv6_ADDRESS](#)
 - 16-byte buffer.*

12.18.1 Detailed Description

Defines data types and constants introduced in UEFI.

Copyright (c) 2006 - 2021, Intel Corporation. All rights reserved.

Portions copyright (c) 2011 - 2016, ARM Ltd. All rights reserved.

Copyright (c) 2020, Hewlett Packard Enterprise Development LP. All rights reserved.

SPDX-License-Identifier: BSD-2-Clause-Patent

12.18.2 Macro Definition Documentation

12.18.2.1 #define EFI_PAGES_TO_SIZE(*Pages*) ((*Pages*) << EFI_PAGE_SHIFT)

Macro that converts a number of EFI_PAGES to a size in bytes.

Parameters

<i>Pages</i>	The number of EFI_PAGES. This parameter is assumed to be type UINTN. Passing in a parameter that is larger than UINTN may produce unexpected results.
--------------	---

Returns

The number of bytes associated with the number of EFI_PAGES specified by *Pages*.

Definition at line 212 of file UefiBaseType.h.

12.18.2.2 #define EFI_SIZE_TO_PAGES(*Size*) (((*Size*) >> EFI_PAGE_SHIFT) + (((*Size*) & EFI_PAGE_MASK) ? 1 : 0))

Macro that converts a size, in bytes, to a number of EFI_PAGESs.

Parameters

<i>Size</i>	A size in bytes. This parameter is assumed to be type UINTN. Passing in a parameter that is larger than UINTN may produce unexpected results.
-------------	---

Returns

The number of EFI_PAGESs associated with the number of bytes specified by *Size*.

Definition at line 199 of file UefiBaseType.h.

12.18.3 Typedef Documentation

12.18.3.1 typedef IPv4_ADDRESS EFI_IPv4_ADDRESS

4-byte buffer.

An IPv4 internet protocol address.

Definition at line 85 of file UefiBaseType.h.

12.18.3.2 typedef IPv6_ADDRESS EFI_IPv6_ADDRESS

16-byte buffer.

An IPv6 internet protocol address.

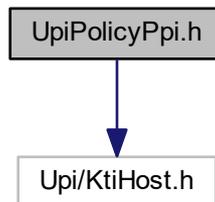
Definition at line 90 of file UefiBaseType.h.

12.19 UpiPolicyPpi.h File Reference

Silicon Policy PPI is used for specifying platform related Intel silicon information and policy setting.

```
#include <Upi/KtiHost.h>
```

Include dependency graph for UpiPolicyPpi.h:



Classes

- struct [_UPI_POLICY_PPI](#)
UPI Policy Structure.

Macros

- #define [UPI_POLICY_PPI_REVISION](#) 0x1
PPI revision information This PPI will be extended in a backwards compatible manner over time Added interfaces should be documented here with the revisions added Revision 1: Initial revision.

12.19.1 Detailed Description

Silicon Policy PPI is used for specifying platform related Intel silicon information and policy setting.

This PPI is consumed by the silicon PEI modules and carried over to silicon DXE modules.

Copyright

Copyright 2017 - 2021 Intel Corporation.

SPDX-License-Identifier: BSD-2-Clause-Patent

Index

- [_BASE_INT_SIZE_OF](#)
 - [Base.h, 192](#)
- [_EFI_MM_RESERVED_MMRAM_REGION, 31](#)
 - [MmramReservedSize, 31](#)
 - [MmramReservedStart, 31](#)
- [_EFI_PEI_MP_SERVICES_PPI, 32](#)
- [_INT_SIZE_OF](#)
 - [Base.h, 192](#)
- [_LIST_ENTRY, 32](#)
- [_MEMORY_POLICY_PPI, 33](#)
- [_PCH_POLICY, 33](#)
 - [IoApicConfig, 36](#)
 - [Revision, 36](#)
 - [SataConfig, 41](#)
 - [TempMemBaseAddr, 41](#)
 - [TempPciBusMin, 41](#)
- [_RAS_IMC_S3_DATA_PPI, 41](#)
- [_UPI_POLICY_PPI, 42](#)
 - [Revision, 42](#)
- [ABS](#)
 - [Base.h, 192](#)
- [ADRDataSaveMode](#)
 - [memSetup, 101](#)
- [ADREn](#)
 - [memSetup, 101](#)
- [ALIGN_POINTER](#)
 - [Base.h, 192](#)
- [ALIGN_VALUE](#)
 - [Base.h, 193](#)
- [ALIGN_VARIABLE](#)
 - [Base.h, 193](#)
- [ALL_LANES_EPARAM_LINK_INFO, 43](#)
- [ANALYZER_NORETURN](#)
 - [Base.h, 193](#)
- [ANALYZER_UNREACHABLE](#)
 - [Base.h, 193](#)
- [ARRAY_SIZE](#)
 - [Base.h, 193](#)
- [AdvMemTestCondPause](#)
 - [memSetup, 101](#)
- [AdvMemTestCondTrefi](#)
 - [memSetup, 101](#)
- [AdvMemTestCondTwr](#)
 - [memSetup, 101](#)
- [AdvMemTestRankData, 43](#)
- [AdvMemTestRankListNumEntries](#)
 - [memSetup, 101](#)
- [AdvMemTestResetList](#)
 - [memSetup, 101](#)
- [AepNotSupportedException](#)
 - [memSetup, 102](#)
- [AllowNoLtrIccPIIShutdown](#)
 - [PCH_PCIE_CONFIG, 147](#)
- [ApdEn](#)
 - [memSetup, 102](#)
- [AppDirectMemoryHole](#)
 - [memSetup, 102](#)
- [BASE_ARG](#)
 - [Base.h, 195](#)
- [BASE_CR](#)
 - [Base.h, 195](#)
- [BASE_LIST](#)
 - [Base.h, 201](#)
- [Base.h, 187](#)
 - [_BASE_INT_SIZE_OF, 192](#)
 - [_INT_SIZE_OF, 192](#)
 - [ABS, 192](#)
 - [ALIGN_POINTER, 192](#)
 - [ALIGN_VALUE, 193](#)
 - [ALIGN_VARIABLE, 193](#)
 - [ANALYZER_NORETURN, 193](#)
 - [ANALYZER_UNREACHABLE, 193](#)
 - [ARRAY_SIZE, 193](#)
 - [BASE_ARG, 195](#)
 - [BASE_CR, 195](#)
 - [BASE_LIST, 201](#)
 - [ENCODE_ERROR, 195](#)
 - [ENCODE_WARNING, 196](#)
 - [FALSE, 196](#)
 - [MAX, 196](#)
 - [MIN, 196](#)
 - [NORETURN, 197](#)
 - [OFFSET_OF, 197](#)
 - [RETURN_ADDRESS, 197](#)
 - [RETURN_BUFFER_TOO_SMALL, 197](#)
 - [RETURN_ERROR, 197](#)
 - [RETURNS_TWICE, 198](#)
 - [SIGNATURE_16, 198](#)
 - [SIGNATURE_32, 198](#)
 - [SIGNATURE_64, 198](#)
 - [STATIC_ASSERT, 199](#)
 - [TRUE, 199](#)
 - [UNREACHABLE, 199](#)
 - [VA_ARG, 199](#)
 - [VA_COPY, 200](#)
 - [VA_END, 200](#)
 - [VA_LIST, 201](#)
 - [VA_START, 200](#)

- BaseAddress
 - EFI_HOB_UEFI_CAPSULE, 66
- BifurcationPcie0
 - FSPS_CONFIG, 84
- BifurcationPcie1
 - FSPS_CONFIG, 84
- BiosInterface
 - PCH_LOCK_DOWN_CONFIG, 141
- BiosLock
 - PCH_LOCK_DOWN_CONFIG, 141
- Blockgnt2cmd1cyc
 - memSetup, 102
- BoardTypeBitmask
 - FSPM_CONFIG, 74
 - KTI_HOST_IN, 91
- BusRatio
 - FSPM_CONFIG, 74
 - KTI_HOST_IN, 92
- CFRImagePtr
 - KTI_HOST_IN, 92
- CPU_POLICY_HOB, 46
 - flexRatioNext, 48
- CacheMemType
 - memSetup, 102
- CapsuleResetType
 - PCH_PM_CONFIG, 155
- chInter
 - memSetup, 103
- check_platform_detect
 - memSetup, 103
- check_pm_sts
 - memSetup, 103
- CkMode
 - memSetup, 104
- CkeldleTimer
 - memSetup, 103
- CkeProgramming
 - memSetup, 103
- ckeThrottling
 - memSetup, 103
- ClkReqDetect
 - PCH_PCIE_ROOT_PORT_CONFIG, 152
- ClkReqNumber
 - PCH_PCIE_ROOT_PORT_CONFIG, 152
- cmdSetupPercentOffset
 - memSetup, 104
- CmilnitOption
 - memSetup, 104
- CmsEnableDramPm
 - memSetup, 104
- ColdResetRequestEnd
 - KTI_HOST_IN, 92
- ColdResetRequestStart
 - KTI_HOST_IN, 92
- commonSetup, 44
 - ddrtXactor, 45
 - debugJumper, 45
 - maxAddrMem, 45
 - options, 45
 - serialDebugMsgLvl, 45
 - SocketConfig, 46
- ComplianceTestMode
 - PCH_PCIE_CONFIG, 147
- CpuPolicyHob.h, 201
- CpuStart
 - EFI_MMRAM_DESCRIPTOR, 68
- CrashLogClear
 - RAS_RC_POLICY_PPI, 179
- CrashLogReArm
 - RAS_RC_POLICY_PPI, 179
- Grid
 - PCH_GENERAL_CONFIG, 130
- D2KCreditConfig
 - FSPM_CONFIG, 75
- DMI_HW_WIDTH_CONTROL, 50
- DXE_ERROR
 - PiMultiPhase.h, 221
- DataBufferDfe
 - memSetup, 104
- DataDIIOff
 - memSetup, 105
- DciAutoDetect
 - PCH_DCI_CONFIG, 127
- DciEn
 - PCH_DCI_CONFIG, 127
- DcpmmAveragePowerLimit
 - memSetup, 105
- DcpmmAveragePowerTimeConstant
 - memSetup, 105
- DcpmmMbbAveragePowerTimeConstant
 - memSetup, 105
- DcpmmMbbFeature
 - memSetup, 105
- DcpmmMbbMaxPowerLimit
 - memSetup, 106
- DdrCacheSize
 - memSetup, 106
- ddrChannelSetup, 48
- ddrDimmSetup, 49
- ddrFreqLimit
 - memSetup, 106
- ddrSocketSetup, 49
- DdrtCkeEn
 - memSetup, 106
- DdrtQosMode
 - FSPM_CONFIG, 75
- ddrtXactor
 - commonSetup, 45
- DeEmphasis
 - FSPM_CONFIG, 75
- debugJumper
 - commonSetup, 45
- DebugPrintLevel
 - FSPM_CONFIG, 75
- Degrade4SPreference
 - FSPM_CONFIG, 75

- DegradePrecedence
 - FSPM_CONFIG, 75
- DetectTimeoutMs
 - PCH_PCIE_CONFIG, 147
- DeviceResetDelay
 - PCH_RST_PCIE_STORAGE_CONFIG, 159
- DeviceResetPad
 - PCH_PCIE_ROOT_PORT_CONFIG, 152
- DfxDnTxPreset
 - FSPM_CONFIG, 75
- DfxRxPreset
 - FSPM_CONFIG, 76
- DfxUpTxPreset
 - FSPM_CONFIG, 76
- dimmTypeSupport
 - memSetup, 106
- DisableComplianceMode
 - PCH_USB_CONFIG, 172
- DisableDirForAppDirect
 - memSetup, 106
- DisableDsxAcPresentPulldown
 - PCH_PM_CONFIG, 155
- DisableEnergyReport
 - PCH_PM_CONFIG, 155
- DisableNativePowerButton
 - PCH_PM_CONFIG, 155
- DisableRootPortClockGating
 - PCH_PCIE_CONFIG, 147
- Disddrtopprd
 - memSetup, 107
- DmiAspm
 - PCH_DMI_CONFIG, 128
- DramRaplEnable
 - memSetup, 107
- dramraplRefreshBase
 - memSetup, 107
- DspEndpointDmic
 - PCH_HDAUDIO_CONFIG, 131
- DspPpModuleMask
 - PCH_HDAUDIO_CONFIG, 131
- EFI_AP_PROCEDURE
 - PiMultiPhase.h, 222
- EFI_AP_PROCEDURE2
 - PiMultiPhase.h, 222
- EFI_AUTH_STATUS_PLATFORM_OVERRIDE
 - PiMultiPhase.h, 222
- EFI_HOB_CPU, 51
 - Header, 51
- EFI_HOB_FIRMWARE_VOLUME, 52
 - Header, 52
- EFI_HOB_FIRMWARE_VOLUME2, 52
 - Header, 53
- EFI_HOB_FIRMWARE_VOLUME3, 54
 - ExtractedFv, 54
 - FileName, 54
 - FvName, 55
 - Header, 55
- EFI_HOB_GENERIC_HEADER, 55
- EFI_HOB_GUID_TYPE, 56
 - Header, 56
- EFI_HOB_HANDOFF_INFO_TABLE, 56
 - EfiMemoryTop, 57
 - Header, 57
 - Version, 58
- EFI_HOB_MEMORY_ALLOCATION, 58
 - Header, 59
- EFI_HOB_MEMORY_ALLOCATION_BSP_STORE, 59
 - Header, 60
- EFI_HOB_MEMORY_ALLOCATION_HEADER, 60
 - MemoryBaseAddress, 61
 - MemoryType, 61
 - Name, 61
- EFI_HOB_MEMORY_ALLOCATION_MODULE, 61
 - Header, 62
- EFI_HOB_MEMORY_ALLOCATION_STACK, 62
 - Header, 63
- EFI_HOB_MEMORY_POOL, 63
 - Header, 64
- EFI_HOB_RESOURCE_DESCRIPTOR, 64
 - Header, 65
 - Owner, 65
- EFI_HOB_UEFI_CAPSULE, 66
 - BaseAddress, 66
- EFI_IP_ADDRESS, 67
- EFI_IPv4_ADDRESS
 - UefiBaseType.h, 230
- EFI_IPv6_ADDRESS
 - UefiBaseType.h, 230
- EFI_MAC_ADDRESS, 67
- EFI_MMRAM_DESCRIPTOR, 68
 - CpuStart, 68
 - PhysicalStart, 68
 - RegionState, 68
- EFI_PAGES_TO_SIZE
 - UefiBaseType.h, 230
- EFI_PEI_HOB_POINTERS, 69
- EFI_PEI_MP_SERVICES_ENABLEDISABLEAP
 - MpServices.h, 210
- EFI_PEI_MP_SERVICES_GET_NUMBER_OF_PROCESSORS
 - MpServices.h, 211
- EFI_PEI_MP_SERVICES_GET_PROCESSOR_INFO
 - MpServices.h, 211
- EFI_PEI_MP_SERVICES_STARTUP_ALL_APS
 - MpServices.h, 212
- EFI_PEI_MP_SERVICES_STARTUP_THIS_AP
 - MpServices.h, 212
- EFI_PEI_MP_SERVICES_SWITCH_BSP
 - MpServices.h, 213
- EFI_PEI_MP_SERVICES_WHOAMI
 - MpServices.h, 214
- EFI_SIZE_TO_PAGES
 - UefiBaseType.h, 230
- EFI_TIME, 69
- ENCODE_ERROR
 - Base.h, 195

- ENCODE_WARNING
 - Base.h, 196
- eSATASpeedLimit
 - PCH_SATA_CONFIG, 160
- EfiMemoryTop
 - EFI_HOB_HANDOFF_INFO_TABLE, 57
- EliminateDirectoryInFarMemory
 - memSetup, 107
- Enable
 - PCH_HDAUDIO_CONFIG, 131
 - PCH_HPET_CONFIG, 132
 - PCH_LAN_CONFIG, 140
 - PCH_MEMORY_THROTTLING, 144
 - PCH_RST_PCIE_STORAGE_CONFIG, 159
 - PCH_SATA_CONFIG, 160
 - PCH_SATA_PORT_CONFIG, 162
 - PCH_SMBUS_CONFIG, 165
 - PCH_XDCI_CONFIG, 175
- EnableGbE
 - FSPS_CONFIG, 84
- EnablePeerMemoryWrite
 - PCH_PCIE_CONFIG, 147
- EnablePort8xhDecode
 - PCH_PCIE_CONFIG, 148
- enforcePOR
 - memSetup, 108
- EnforcePopulationPor
 - memSetup, 107
- EnhancePort8xhDecoding
 - PCH_LPC_CONFIG, 143
- EqPh3LaneParam
 - PCH_PCIE_CONFIG, 148
- ExtendedADDDCEn
 - memSetup, 108
- ExtendedType17
 - memSetup, 108
- ExtractedFv
 - EFI_HOB_FIRMWARE_VOLUME3, 54
- FALSE
 - Base.h, 196
- FSPM_CONFIG, 70
 - BoardTypeBitmask, 74
 - BusRatio, 74
 - D2KCreditConfig, 75
 - DdrtQosMode, 75
 - DeEmphasis, 75
 - DebugPrintLevel, 75
 - Degrade4SPreference, 75
 - DegradePrecedence, 75
 - DfxDnTxPreset, 75
 - DfxRxPreset, 76
 - DfxUpTxPreset, 76
 - IIOpcieMaxPayload, 76
 - IIOpciePortLinkSpeed, 76
 - IoDcMode, 76
 - IrqThreshold, 76
 - IsKtiNvramDataReady, 76
 - KtiCpuSktHotPlugTopology, 76
 - KtiCrcMode, 77
 - KtiFailoverEn, 77
 - KtiLinkL0pEn, 77
 - KtiLinkL1En, 77
 - KtiLinkSpeed, 77
 - KtiLinkSpeedMode, 77
 - LegacyVgaSoc, 77
 - LegacyVgaStack, 78
 - MbeBwCal, 78
 - mmCfgBase, 78
 - mmCfgSize, 78
 - mmiohBase, 78
 - mmiohSize, 78
 - NtbBarSizeEmBarSZ1, 78
 - NtbBarSizeEmBarSZ2, 79
 - NtbBarSizeEmBarSZ2_0, 79
 - NtbBarSizeEmBarSZ2_1, 79
 - NtbBarSizeImBar1, 79
 - NtbBarSizeImBar2, 79
 - NtbBarSizeImBar2_0, 79
 - NtbBarSizeImBar2_1, 79
 - NtbPpd, 79
 - NtbXlinkCtlOverride, 80
 - PchPciePIISsc, 80
 - PchSirqMode, 80
 - PcieCommonClock, 80
 - SerialUartDebugEnable, 80
 - SerialUartDebugloBase, 80
 - SnoopAllCores, 80
 - SnoopThrottleConfig, 81
 - SplitLock, 81
 - StaleAtoSOptEn, 81
 - ThermalDeviceEnable, 81
 - TorThresLoctoremEmpty, 81
 - TorThresLoctoremNorm, 81
 - TscSyncEn, 81
 - UmaClustering, 82
 - WaitTimeForPSBP, 82
 - XptPrefetchEn, 82
- FSPM_UPD, 82
- FSPS_CONFIG, 83
 - BifurcationPcie0, 84
 - BifurcationPcie1, 84
 - EnableGbE, 84
- FSPS_UPD, 85
- FSPT_CONFIG, 85
- FSPT_CORE_UPD, 86
- FSPT_UPD, 86
- FastGoConfig
 - memSetup, 108
- FileName
 - EFI_HOB_FIRMWARE_VOLUME3, 54
- flexRatioNext
 - CPU_POLICY_HOB, 48
- ForcePxclnit
 - memSetup, 109
- FspFixedPcds.h, 202
- FspUpd.h, 206

- FspmUpd.h, 202
- FspsUpd.h, 204
- FsptUpd.h, 205
- FvName
 - EFI_HOB_FIRMWARE_VOLUME3, 55
- GUID, 87
- Gen3EqPh3Method
 - PCH_PCIE_ROOT_PORT_CONFIG, 152
- GlobalSmi
 - PCH_LOCK_DOWN_CONFIG, 141
- Gp27WakeFromDeepSx
 - PCH_WAKE_CONFIG, 173
- GpioLockDown
 - PCH_LOCK_DOWN_CONFIG, 142
- Header
 - EFI_HOB_CPU, 51
 - EFI_HOB_FIRMWARE_VOLUME, 52
 - EFI_HOB_FIRMWARE_VOLUME2, 53
 - EFI_HOB_FIRMWARE_VOLUME3, 55
 - EFI_HOB_GUID_TYPE, 56
 - EFI_HOB_HANDOFF_INFO_TABLE, 57
 - EFI_HOB_MEMORY_ALLOCATION, 59
 - EFI_HOB_MEMORY_ALLOCATION_BSP_STORE, 60
 - EFI_HOB_MEMORY_ALLOCATION_MODULE, 62
 - EFI_HOB_MEMORY_ALLOCATION_STACK, 63
 - EFI_HOB_MEMORY_POOL, 64
 - EFI_HOB_RESOURCE_DESCRIPTOR, 65
- highGap
 - KTI_HOST_IN, 92
- HsioRxEqBoostMagAd
 - PCH_SATA_PORT_CONFIG, 162
- HsioRxEqBoostMagAdEnable
 - PCH_SATA_PORT_CONFIG, 162
- HsioRxSetCtle
 - PCH_PCIE_ROOT_PORT_CONFIG, 152
- HsioRxSetCtleEnable
 - PCH_PCIE_ROOT_PORT_CONFIG, 152
- HsioTxGen1DownscaleAmp
 - PCH_SATA_PORT_CONFIG, 162
- HsioTxGen1DownscaleAmpEnable
 - PCH_SATA_PORT_CONFIG, 162
- HsioTxGen2DownscaleAmp
 - PCH_SATA_PORT_CONFIG, 162
- HsioTxGen2DownscaleAmpEnable
 - PCH_SATA_PORT_CONFIG, 162
- IIOpcieMaxPayload
 - FSPM_CONFIG, 76
- IIOpciePortLinkSpeed
 - FSPM_CONFIG, 76
- IPv4_ADDRESS, 88
- IPv6_ADDRESS, 88
- imcBclk
 - memSetup, 109
- IoApicConfig
 - _PCH_POLICY, 36
- IoDcMode
 - FSPM_CONFIG, 76
- IrqThreshold
 - FSPM_CONFIG, 76
- IsKtiNvramDataReady
 - FSPM_CONFIG, 76
- KTI_HOST_IN, 88
 - BoardTypeBitmask, 91
 - BusRatio, 92
 - CFRImagePtr, 92
 - ColdResetRequestEnd, 92
 - ColdResetRequestStart, 92
 - highGap, 92
 - lowGap, 92
 - OemCheckCpuPartsChangeSwap, 93
 - OemGetAdaptedEqSettings, 93
 - SplitLock, 93
- KtiCpuSktHotPlugTopology
 - FSPM_CONFIG, 76
- KtiCrcMode
 - FSPM_CONFIG, 77
- KtiFailoverEn
 - FSPM_CONFIG, 77
- KtiHost.h, 207
- KtiLinkL0pEn
 - FSPM_CONFIG, 77
- KtiLinkL1En
 - FSPM_CONFIG, 77
- KtiLinkSpeed
 - FSPM_CONFIG, 77
- KtiLinkSpeedMode
 - FSPM_CONFIG, 77
- LatchSystemShutdownState
 - memSetup, 109
- LegacyADRMdModeEn
 - memSetup, 109
- LegacyVgaSoc
 - FSPM_CONFIG, 77
- LegacyVgaStack
 - FSPM_CONFIG, 78
- lowGap
 - KTI_HOST_IN, 92
- LsxImplementation
 - memSetup, 110
- MAX
 - Base.h, 196
- MIN
 - Base.h, 196
- maxAddrMem
 - commonSetup, 45
- MbeBwCal
 - FSPM_CONFIG, 78
- MdIIOffEn
 - memSetup, 110
- memFlows

- memSetup, 110
- memFlowsExt
 - memSetup, 111
- MemHotOutputAssertThreshold
 - memSetup, 111
- memSetup, 93
 - ADRDDataSaveMode, 101
 - ADREn, 101
 - AdvMemTestCondPause, 101
 - AdvMemTestCondTrefi, 101
 - AdvMemTestCondTwr, 101
 - AdvMemTestRankListNumEntries, 101
 - AdvMemTestResetList, 101
 - AepNotSupportedException, 102
 - ApdEn, 102
 - AppDirectMemoryHole, 102
 - Blockgnt2cmd1cyc, 102
 - CacheMemType, 102
 - chInter, 103
 - check_platform_detect, 103
 - check_pm_sts, 103
 - CkMode, 104
 - CkIdleTimer, 103
 - CkeProgramming, 103
 - ckeThrottling, 103
 - cmdSetupPercentOffset, 104
 - CmilInitOption, 104
 - CmsEnableDramPm, 104
 - DataBufferDfe, 104
 - DataDIIOff, 105
 - DcpmmAveragePowerLimit, 105
 - DcpmmAveragePowerTimeConstant, 105
 - DcpmmMbbAveragePowerTimeConstant, 105
 - DcpmmMbbFeature, 105
 - DcpmmMbbMaxPowerLimit, 106
 - DdrCacheSize, 106
 - ddrFreqLimit, 106
 - DdrtCkeEn, 106
 - dimmTypeSupport, 106
 - DisableDirForAppDirect, 106
 - Disddrtopprd, 107
 - DramRaplEnable, 107
 - dramraplRefreshBase, 107
 - EliminateDirectoryInFarMemory, 107
 - enforcePOR, 108
 - EnforcePopulationPor, 107
 - ExtendedADDDCEn, 108
 - ExtendedType17, 108
 - FastGoConfig, 108
 - ForcePxclnit, 109
 - imcBclk, 109
 - LatchSystemShutdownState, 109
 - LegacyADRMModeEn, 109
 - LsxImplementation, 110
 - MdIIOffEn, 110
 - memFlows, 110
 - memFlowsExt, 111
 - MemHotOutputAssertThreshold, 111
 - MemoryConnectorType, 112
 - MemoryTopology, 112
 - MinNormalMemSize, 112
 - NfitPublishMailboxStructsDisable, 112
 - normOpplntvl, 112
 - NvDimmEnergyPolicy, 112
 - NvdimmSmbusMaxAccessTime, 113
 - NvdimmSmbusReleaseDelay, 113
 - NvmMediaStatusException, 113
 - NvmQos, 114
 - NvmdimmPerfConfig, 113
 - NvmdimmPowerCyclePolicy, 113
 - olttPeakBWLIMITPercent, 114
 - OppSrefEn, 114
 - options, 114
 - optionsExt, 115
 - optionsNgn, 116
 - PanicWm, 116
 - partialMirrorUEFI, 117
 - partialmirrorpercent, 116
 - partialmirrorsad0, 117
 - partialmirrors, 117
 - patrolScrubAddrMode, 117
 - PdaModeX16, 117
 - PeriodicRcomp, 118
 - PeriodicRcompInterval, 118
 - PkgcSrefEn, 118
 - PpdEn, 118
 - pprAddrSetup, 118
 - pprType, 119
 - readPreamble, 119
 - RxDfeEn, 119
 - setSecureEraseAllDIMMs, 119
 - setSecureEraseSktCh, 120
 - smartTestKey, 120
 - spareErrTh, 120
 - SpdPrintEn, 120
 - SpdPrintLength, 120
 - SpdSmbSpeed, 121
 - SrefProgramming, 121
 - TempRefreshOption, 121
 - thermalThrottlingOptions, 121
 - ThrottlingMidOnTempLo, 121
 - TrainingCompOptions, 122
 - trainingResultOffsetFunctionEnable, 122
 - TxRiseFallSlewRate, 122
 - UseSmbusForMrwEarly, 122
 - VirtualNumaEnable, 122
 - volMemMode, 123
 - writePreamble, 123
 - memTiming, 123
 - nCL, 124
 - nCMDRate, 124
 - nFAW, 124
 - nRAS, 125
 - nRC, 125
 - nRCD, 125
 - nRFC, 125

- nRP, 125
- nRRD, 125
- nRTP, 126
- nWR, 126
- nWTR, 126
- MemoryBaseAddress
 - EFI_HOB_MEMORY_ALLOCATION_HEADER, 61
- MemoryConnectorType
 - memSetup, 112
- MemoryPolicyPpi.h, 208
- MemoryTopology
 - memSetup, 112
- MemoryType
 - EFI_HOB_MEMORY_ALLOCATION_HEADER, 61
- MinNormalMemSize
 - memSetup, 112
- mmCfgBase
 - FSPM_CONFIG, 78
- mmCfgSize
 - FSPM_CONFIG, 78
- mmiohBase
 - FSPM_CONFIG, 78
- mmiohSize
 - FSPM_CONFIG, 78
- MmramReservedSize
 - _EFI_MM_RESERVED_MMRAM_REGION, 31
- MmramReservedStart
 - _EFI_MM_RESERVED_MMRAM_REGION, 31
- MpServices.h, 209
 - EFI_PEI_MP_SERVICES_ENABLEDISABLEAP, 210
 - EFI_PEI_MP_SERVICES_GET_NUMBER_OF_PROCESSORS, 211
 - EFI_PEI_MP_SERVICES_GET_PROCESSOR_INFO, 211
 - EFI_PEI_MP_SERVICES_STARTUP_ALL_APS, 212
 - EFI_PEI_MP_SERVICES_STARTUP_THIS_AP, 212
 - EFI_PEI_MP_SERVICES_SWITCH_BSP, 213
 - EFI_PEI_MP_SERVICES_WHOAMI, 214
- nCL
 - memTiming, 124
- nCMDRate
 - memTiming, 124
- nFAW
 - memTiming, 124
- NORETURN
 - Base.h, 197
- nRAS
 - memTiming, 125
- nRC
 - memTiming, 125
- nRCD
 - memTiming, 125
- nRFC
 - memTiming, 125
- nRP
 - memTiming, 125
- nRRD
 - memTiming, 125
- nRTP
 - memTiming, 126
- nWR
 - memTiming, 126
- nWTR
 - memTiming, 126
- Name
 - EFI_HOB_MEMORY_ALLOCATION_HEADER, 61
- NfitPublishMailboxStructsDisable
 - memSetup, 112
- normOppIntvl
 - memSetup, 112
- NtbBarSizeEmBarSZ1
 - FSPM_CONFIG, 78
- NtbBarSizeEmBarSZ2
 - FSPM_CONFIG, 79
- NtbBarSizeEmBarSZ2_0
 - FSPM_CONFIG, 79
- NtbBarSizeEmBarSZ2_1
 - FSPM_CONFIG, 79
- NtbBarSizeImBar1
 - FSPM_CONFIG, 79
- NtbBarSizeImBar2
 - FSPM_CONFIG, 79
- NtbBarSizeImBar2_0
 - FSPM_CONFIG, 79
- NtbBarSizeImBar2_1
 - FSPM_CONFIG, 79
- NtbPpd
 - FSPM_CONFIG, 79
- NtbXlinkCtlOverride
 - FSPM_CONFIG, 80
- NvDimmEnergyPolicy
 - memSetup, 112
- NvdimmSmbusMaxAccessTime
 - memSetup, 113
- NvdimmSmbusReleaseDelay
 - memSetup, 113
- NvmMediaStatusException
 - memSetup, 113
- NvmQos
 - memSetup, 114
- NvmdimmPerfConfig
 - memSetup, 113
- NvmdimmPowerCyclePolicy
 - memSetup, 113
- OFFSET_OF
 - Base.h, 197
- OemCheckCpuPartsChangeSwap
 - KTI_HOST_IN, 93
- OemGetAdaptedEqSettings
 - KTI_HOST_IN, 93

- olttPeakBWLIMITPercent
 - memSetup, 114
- OppSrefEn
 - memSetup, 114
- options
 - commonSetup, 45
 - memSetup, 114
- optionsExt
 - memSetup, 115
- optionsNgn
 - memSetup, 116
- OverCurrentPin
 - PCH_USB20_PORT_CONFIG, 170
 - PCH_USB30_PORT_CONFIG, 171
- Owner
 - EFI_HOB_RESOURCE_DESCRIPTOR, 65
- P2SbReveal
 - PCH_P2SB_CONFIG, 145
- PCH_DCI_CONFIG, 126
 - DciAutoDetect, 127
 - DciEn, 127
- PCH_DEVICE_INTERRUPT_CONFIG, 127
- PCH_DMI_CONFIG, 127
 - DmiAspm, 128
- PCH_FLASH_PROTECTION_CONFIG, 128
- PCH_GBL2HOST_EN, 129
- PCH_GENERAL_CONFIG, 129
 - Crid, 130
 - SubSystemVendorId, 130
- PCH_HDAUDIO_CONFIG, 130
 - DspEndpointDmic, 131
 - DspPpModuleMask, 131
 - Enable, 131
- PCH_HDAUDIO_IO_BUFFER_OWNERSHIP
 - PchPolicyCommon.h, 217
- PCH_HPET_CONFIG, 132
 - Enable, 132
- PCH_HSIO_PCIE_CONFIG, 132
- PCH_HSIO_PCIE_LANE_CONFIG, 133
 - RsvdBits2, 135
 - RsvdBits3, 135
- PCH_HSIO_PCIE_WM20_CONFIG, 135
- PCH_HSIO_SATA_CONFIG, 136
- PCH_HSIO_SATA_PORT_LANE, 136
- PCH_INT_PIN
 - PchPolicyCommon.h, 217
- PCH_INTERRUPT_CONFIG, 138
- PCH_IOAPIC_CONFIG, 139
- PCH_LAN_CONFIG, 140
 - Enable, 140
- PCH_LOCK_DOWN_CONFIG, 140
 - BiosInterface, 141
 - BiosLock, 141
 - GlobalSmi, 141
 - GpioLockDown, 142
 - RtcLock, 142
 - SpiEiss, 142
 - TcoLock, 142
- PCH_LPC_CONFIG, 142
 - EnhancePort8xhDecoding, 143
- PCH_LPC_SIRQ_CONFIG, 143
- PCH_MEMORY_THROTTLING, 144
 - Enable, 144
 - TsGpioPinSetting, 144
- PCH_P2SB_CONFIG, 145
 - P2SbReveal, 145
 - PsfUnlock, 145
 - SbiUnlock, 145
- PCH_PCIE_CONFIG, 145
 - AllowNoLtrLccPllShutdown, 147
 - ComplianceTestMode, 147
 - DetectTimeoutMs, 147
 - DisableRootPortClockGating, 147
 - EnablePeerMemoryWrite, 147
 - EnablePort8xhDecode, 148
 - EqPh3LaneParam, 148
 - RpFunctionSwap, 148
- PCH_PCIE_CONFIG2, 148
- PCH_PCIE_EQ_LANE_PARAM, 149
- PCH_PCIE_EQ_METHOD
 - PchPolicyCommon.h, 218
- PCH_PCIE_ROOT_PORT_CONFIG, 150
 - ClkReqDetect, 152
 - ClkReqNumber, 152
 - DeviceResetPad, 152
 - Gen3EqPh3Method, 152
 - HsioRxSetCtle, 152
 - HsioRxSetCtleEnable, 152
 - PcieSpeed, 153
 - SlotImplemented, 153
- PCH_PM_CONFIG, 153
 - CapsuleResetType, 155
 - DisableDsxAcPresentPulldown, 155
 - DisableEnergyReport, 155
 - DisableNativePowerButton, 155
 - PchPwrCycDur, 155
 - PciClockRun, 155
 - PciePllSsc, 156
 - PmcReadDisable, 156
 - PowerResetStatusClear, 156
 - PwrBtnOverridePeriod, 156
 - RsvdBits0, 156
 - SlpLanLowDc, 156
 - SlpS0Enable, 156
- PCH_PORT61H_SMM_CONFIG, 157
- PCH_POWER_RESET_STATUS, 157
- PCH_RESERVED_PAGE_ROUTE
 - PchPolicyCommon.h, 218
- PCH_RST_PCIE_STORAGE_CONFIG, 158
 - DeviceResetDelay, 159
 - Enable, 159
 - RstPcieStoragePort, 159
- PCH_SATA_CONFIG, 159
 - eSATA SpeedLimit, 160
 - Enable, 160
 - SataMode, 160

- PCH_SATA_PORT_CONFIG, 161
 - Enable, 162
 - HsioRxEqBoostMagAd, 162
 - HsioRxEqBoostMagAdEnable, 162
 - HsioTxGen1DownscaleAmp, 162
 - HsioTxGen1DownscaleAmpEnable, 162
 - HsioTxGen2DownscaleAmp, 162
 - HsioTxGen2DownscaleAmpEnable, 162
 - ZpOdd, 162
- PCH_SATA_RST_CONFIG, 163
- PCH_SKYCAM_CIO2_FLS_CONFIG, 164
- PCH_SLP_S4_MIN_ASSERT
 - PchPolicyCommon.h, 218
- PCH_SMBUS_CONFIG, 165
 - Enable, 165
- PCH_SPI_CONFIG, 166
 - ShowSpiController, 166
- PCH_SSI_CONFIG, 166
- PCH_THERMAL_CONFIG, 167
 - PchHotLevel, 168
 - ThermalDeviceEnable, 168
 - ThermalThrottling, 168
- PCH_THERMAL_THROTTLING, 168
- PCH_TRACE_HUB_CONFIG, 169
- PCH_USB20_PORT_CONFIG, 169
 - OverCurrentPin, 170
- PCH_USB30_PORT_CONFIG, 170
 - OverCurrentPin, 171
- PCH_USB_CONFIG, 171
 - DisableComplianceMode, 172
 - PortUsb20, 172
 - UsbPrecondition, 172
- PCH_USB_PORT_LOCATION
 - PchPolicyCommon.h, 218
- PCH_WAKE_CONFIG, 173
 - Gp27WakeFromDeepSx, 173
 - PmeB0S5Dis, 173
- PCH_WDT_CONFIG, 174
- PCH_XDCI_CONFIG, 174
 - Enable, 175
- PCH_XHCI_SSI_PORT, 175
- PER_LANE_EPARAM_LINK_INFO, 175
- PI_ENCODE_ERROR
 - PiMultiPhase.h, 222
- PI_ENCODE_WARNING
 - PiMultiPhase.h, 222
- PPR_ADDR, 176
- PPR_ADDR_MRC_SETUP, 176
- PROTECTED_RANGE, 177
- PSMI_POLICY_DATA_HOB, 177
- PanicWm
 - memSetup, 116
- partialMirrorUEFI
 - memSetup, 117
- partialMirrorpercent
 - memSetup, 116
- partialMirrorsad0
 - memSetup, 117
- partialmirrorsts
 - memSetup, 117
- patrolScrubAddrMode
 - memSetup, 117
- PchCrossThrottling
 - THERMAL_THROTTLE_LEVELS, 184
- PchHdaloBufOwnerHdaLink
 - PchPolicyCommon.h, 217
- PchHdaloBufOwnerHdaLinkI2sPort
 - PchPolicyCommon.h, 217
- PchHdaloBufOwnerI2sPort
 - PchPolicyCommon.h, 217
- PchHotLevel
 - PCH_THERMAL_CONFIG, 168
- PchNoInt
 - PchPolicyCommon.h, 218
- PchPcieEqDefault
 - PchPolicyCommon.h, 218
- PchPcieEqHardware
 - PchPolicyCommon.h, 218
- PchPcieEqSoftware
 - PchPolicyCommon.h, 218
- PchPcieEqStaticCoeff
 - PchPolicyCommon.h, 218
- PchPciePIISsc
 - FSPM_CONFIG, 80
- PchPolicyCommon.h, 214
 - PCH_HDAUDIO_IO_BUFFER_OWNERSHIP, 217
 - PCH_INT_PIN, 217
 - PCH_PCIE_EQ_METHOD, 218
 - PCH_RESERVED_PAGE_ROUTE, 218
 - PCH_SLP_S4_MIN_ASSERT, 218
 - PCH_USB_PORT_LOCATION, 218
 - PchHdaloBufOwnerHdaLink, 217
 - PchHdaloBufOwnerHdaLinkI2sPort, 217
 - PchHdaloBufOwnerI2sPort, 217
 - PchNoInt, 218
 - PchPcieEqDefault, 218
 - PchPcieEqHardware, 218
 - PchPcieEqSoftware, 218
 - PchPcieEqStaticCoeff, 218
 - PchReservedPageToLpc, 218
 - PchReservedPageToPcie, 218
 - PchSlpS4PchTime, 218
- PchPwrCycDur
 - PCH_PM_CONFIG, 155
- PchReservedPageToLpc
 - PchPolicyCommon.h, 218
- PchReservedPageToPcie
 - PchPolicyCommon.h, 218
- PchSirqMode
 - FSPM_CONFIG, 80
- PchSlpS4PchTime
 - PchPolicyCommon.h, 218
- PciClockRun
 - PCH_PM_CONFIG, 155
- PcieCommonClock
 - FSPM_CONFIG, 80

- PciePllSsc
 - PCH_PM_CONFIG, 156
- PcieSpeed
 - PCH_PCIE_ROOT_PORT_CONFIG, 153
- PdaModeX16
 - memSetup, 117
- PeriodicRcomp
 - memSetup, 118
- PeriodicRcompInterval
 - memSetup, 118
- PhysicalStart
 - EFI_MMRAM_DESCRIPTOR, 68
- PiHob.h, 218
- PiMultiPhase.h, 220
 - DXE_ERROR, 221
 - EFI_AP_PROCEDURE, 222
 - EFI_AP_PROCEDURE2, 222
 - EFI_AUTH_STATUS_PLATFORM_OVERRIDE, 222
 - PI_ENCODE_ERROR, 222
 - PI_ENCODE_WARNING, 222
- PkgcSrefEn
 - memSetup, 118
- PmcReadDisable
 - PCH_PM_CONFIG, 156
- PmeB0S5Dis
 - PCH_WAKE_CONFIG, 173
- PmsyncEnable
 - TS_GPIO_PIN_SETTING, 185
- PortUsb20
 - PCH_USB_CONFIG, 172
- PowerResetStatusClear
 - PCH_PM_CONFIG, 156
- PpdEn
 - memSetup, 118
- pprAddrSetup
 - memSetup, 118
- pprType
 - memSetup, 119
- PsfUnlock
 - PCH_P2SB_CONFIG, 145
- PsmiPolicyHob.h, 223
- PwrBtnOverridePeriod
 - PCH_PM_CONFIG, 156
- RAS_IMC_S3_DATA_PPI_GET_IMC_S3_RAS_DATA
 - RasImcS3Data.h, 224
- RAS_RC_POLICY_PPI, 178
 - CrashLogClear, 179
 - CrashLogReArm, 179
- RETURN_ADDRESS
 - Base.h, 197
- RETURN_BUFFER_TOO_SMALL
 - Base.h, 197
- RETURN_ERROR
 - Base.h, 197
- RETURNS_TWICE
 - Base.h, 198
- RasImcS3Data.h, 223
 - RAS_IMC_S3_DATA_PPI_GET_IMC_S3_RAS_DATA, 224
- RasRcPolicyPpi.h, 226
- readPreamble
 - memSetup, 119
- RegionState
 - EFI_MMRAM_DESCRIPTOR, 68
- Revision
 - _PCH_POLICY, 36
 - _UPI_POLICY_PPI, 42
- RpFunctionSwap
 - PCH_PCIE_CONFIG, 148
- RstPcieStoragePort
 - PCH_RST_PCIE_STORAGE_CONFIG, 159
- RsvdBits0
 - PCH_PM_CONFIG, 156
- RsvdBits2
 - PCH_HSIO_PCIE_LANE_CONFIG, 135
- RsvdBits3
 - PCH_HSIO_PCIE_LANE_CONFIG, 135
- RtcLock
 - PCH_LOCK_DOWN_CONFIG, 142
- RxDfeEn
 - memSetup, 119
- SATA_THERMAL_THROTTLE, 179
- SECURITY_POLICY, 180
 - SgxDebugMode, 181
 - SgxSinitDataFromTpm, 181
 - SgxSinitNvsData, 181
- SIGNATURE_16
 - Base.h, 198
- SIGNATURE_32
 - Base.h, 198
- SIGNATURE_64
 - Base.h, 198
- STATIC_ASSERT
 - Base.h, 199
- SataConfig
 - _PCH_POLICY, 41
- SataMode
 - PCH_SATA_CONFIG, 160
- SbiUnlock
 - PCH_P2SB_CONFIG, 145
- SecurityPolicy.h, 226
- serialDebugMsgLvl
 - commonSetup, 45
- SerialUartDebugEnable
 - FSPM_CONFIG, 80
- SerialUartDebugIoBase
 - FSPM_CONFIG, 80
- setSecureEraseAllDIMMs
 - memSetup, 119
- setSecureEraseSktCh
 - memSetup, 120
- SgxDebugMode
 - SECURITY_POLICY, 181
- SgxSinitDataFromTpm
 - SECURITY_POLICY, 181

- SgxSinitNvsData
 - SECURITY_POLICY, 181
- ShowSpiController
 - PCH_SPI_CONFIG, 166
- SlotImplemented
 - PCH_PCIE_ROOT_PORT_CONFIG, 153
- SlpLanLowDc
 - PCH_PM_CONFIG, 156
- SlpS0Enable
 - PCH_PM_CONFIG, 156
- smartTestKey
 - memSetup, 120
- SnoopAllCores
 - FSPM_CONFIG, 80
- SnoopThrottleConfig
 - FSPM_CONFIG, 81
- SocketConfig
 - commonSetup, 46
- spareErrTh
 - memSetup, 120
- SpdPrintEn
 - memSetup, 120
- SpdPrintLength
 - memSetup, 120
- SpdSmbSpeed
 - memSetup, 121
- SpiEiss
 - PCH_LOCK_DOWN_CONFIG, 142
- SplitLock
 - FSPM_CONFIG, 81
 - KTI_HOST_IN, 93
- SrefProgramming
 - memSetup, 121
- StaleAtoSOptEn
 - FSPM_CONFIG, 81
- SubSystemVendorId
 - PCH_GENERAL_CONFIG, 130
- sysSetup, 182
 - WFRWAEEnable, 183
- THERMAL_THROTTLE_LEVELS, 183
 - PchCrossThrottling, 184
 - TTLock, 184
 - TTState13Enable, 184
- TRACE_INFO, 184
- TRUE
 - Base.h, 199
- TS_GPIO_PIN_SETTING, 185
 - PmsyncEnable, 185
- TTLock
 - THERMAL_THROTTLE_LEVELS, 184
- TTState13Enable
 - THERMAL_THROTTLE_LEVELS, 184
- TcoLock
 - PCH_LOCK_DOWN_CONFIG, 142
- TempMemBaseAddr
 - _PCH_POLICY, 41
- TempPciBusMin
 - _PCH_POLICY, 41
- TempRefreshOption
 - memSetup, 121
- ThermalDeviceEnable
 - FSPM_CONFIG, 81
 - PCH_THERMAL_CONFIG, 168
- ThermalThrottling
 - PCH_THERMAL_CONFIG, 168
- thermalThrottlingOptions
 - memSetup, 121
- ThrottlingMidOnTempLo
 - memSetup, 121
- TorThresLoctoremEmpty
 - FSPM_CONFIG, 81
- TorThresLoctoremNorm
 - FSPM_CONFIG, 81
- TrainingCompOptions
 - memSetup, 122
- trainingResultOffsetFunctionEnable
 - memSetup, 122
- TsGpioPinSetting
 - PCH_MEMORY_THROTTLING, 144
- TscSyncEn
 - FSPM_CONFIG, 81
- TxRiseFallSlewRate
 - memSetup, 122
- UNREACHABLE
 - Base.h, 199
- UefiBaseType.h, 227
 - EFI_IPv4_ADDRESS, 230
 - EFI_IPv6_ADDRESS, 230
 - EFI_PAGES_TO_SIZE, 230
 - EFI_SIZE_TO_PAGES, 230
- UmaClustering
 - FSPM_CONFIG, 82
- UpiPolicyPpi.h, 231
- UsbPrecondition
 - PCH_USB_CONFIG, 172
- UseSmbusForMrwEarly
 - memSetup, 122
- VA_ARG
 - Base.h, 199
- VA_COPY
 - Base.h, 200
- VA_END
 - Base.h, 200
- VA_LIST
 - Base.h, 201
- VA_START
 - Base.h, 200
- Version
 - EFI_HOB_HANDOFF_INFO_TABLE, 58
- VirtualNumaEnable
 - memSetup, 122
- volMemMode
 - memSetup, 123
- WFRWAEEnable

sysSetup, [183](#)
WaitTimeForPSBP
 FSPM_CONFIG, [82](#)
writePreamble
 memSetup, [123](#)

XptPrefetchEn
 FSPM_CONFIG, [82](#)

ZpOdd
 PCH_SATA_PORT_CONFIG, [162](#)
