Cell Broadband Engine Registers

Version 1.5

April 2, 2007

Printed in the United States of America April 2007

IBM Systems and Technology Group
2070 Route 52, Bldg. 330
Hopewell Junction, NY 12533-6351

The IBM home page can be found at **ibm.com**®

The IBM Semiconductor solutions home page can be found at **ibm.com**/chips

Version 1.5
April 2, 2007

Contents

List of Tables

Revision Log

Each release of this document supersedes all previously released versions. The revision log lists all significant changes made to the document since its initial release. In the rest of the document, change bars in the margin indicate that the adjacent text was modified from the previous release of this document.

Revision Date	Contents of Modification
April 2, 2007	Version 1.5 • This revision includes designs up through CMOS SOI 65 nm, DD 1.1. • Updated the document references in *Section Related Publications* on page 15. • Edited *Section 1.2 MMIO Access Rules for 32-Bit and 64-Bit Registers* on page 20 to clarify MMIO access rules. • Added the BIU_FIR and CIU_FIR Register addresses to *Table 2-1 PPE Privilege MMIO Memory Map* on page 24. • Added *Section 2.2.1 CIU Fault Isolation, Error Mask, Checkstop Enable Registers* on page 34. • Corrected the bit diagram for *Section 2.3.1 NCU Mode Setup Register (NCU_ModeSetup)* on page 40. • Corrected the initial POR value *Section 3.2.11.4 MFC Transfer Class ID Register (MFC_TClassID)* on page 86. • Corrected the MFC_CQ_SR Register read/write field in *Table 3-2 SPE Privilege 2 Memory Map* on page 49 and *Section 3.3.2.1 MFC Command Queue Context Save/Restore Register (MFC_CQ_SR)* on page 100. • Updated the SPU_ECC_Cntl[59:60] bit description in *Section 3.2.14.1 SPU ECC Control Register (SPU_ECC_Cntl)* on page 91. • Edited the note in *Section 3.4.2.1 MFC Local Store Address Register (MFC_LSA)* on page 112 • Corrected the address range for IIC Registers in *Section 6 Internal Interrupt Controller (IIC) MMIO Registers* on page 145. • Added the IOC_FIR Register addresses to *Table 6-1 IIC Memory Map* on page 145. • Corrected the initial POR value for the MIC_FIR reset/set fields in *Section 7.7.1 MIC FIR/Checkstop Enable/Reset/Set Registers (MIC_FIR)* on page 203. • Corrected the MIC_FIR Register example and bit description [0:14] in *Section 7.7.1 MIC FIR/Checkstop Enable/Reset/Set Registers (MIC_FIR)* on page 203 • Edited the EIB_Int[LLD_Ind] reference description in *Section 10.2 EIB Interrupt Register (EIB_Int)* on page 230. • Modified the performance monitor interval timer description in *Section 11.4.3 Performance Monitor Interval Register (pm_interval)* on page 255.
July 5. 2006	Version 1.4 • Added a note clarifying updates to the MFC_VR, SPU_VR, BP_VR, PVR Register values. • Corrected minor typographical errors.
June 2, 2006	Version 1.3 Updated the description for the following registers: • Updated the description for the CIU _ModeSetup Register. • Updated the description for the IOC_IOST_Origin Register. • Correction to the BP_VR Register.
April 28, 2006	Version 1.2 • Added Version Register values for CMOS SOI 10KE0 DD 3.2 and CMOS SOI 11S0 DD1.0 (MFC_VR, SPU_VR, BP_VR, PVR).

Revision Date	Contents of Modification
April 7, 2006	Version 1.1. • Added the following registers: – L2_FIR, L2_FIR_Set, L2_FIR_Reset, L2_FIR_Err, L2_FIR_Err_Set, L2_FIR_Err_Reset, L2_FIR_ChkStopEnbl. – MIC_Aux_Trc_Base, MIC_Aux_Trc_Cur_Addr, MIC_Aux_Trc_Grf_Addr, MIC_Aux_Trc_Grf_Data, MIC_Aux_Trc_Max_Addr, MIC_Calibration_Addr_n, MIC_Cmd_Dur_n, MIC_Cmd_Spc_n, MIC_Ctl_Cnfg_n, MIC_Ctl_Cnfg2, MIC_Dev_Cfg_n, MIC_DF_Config, MIC_DF_Ctl_n, MIC_Ecc_Addr_n, MIC_Exc, MIC_FIR, MIC_FIR_Debug, MIC_Mem_Cfg_n, MIC_Mnt_Cfg, MIC_PTCal_Adr_n, MIC_Que_BurstSize_n, MIC_Ref_Scb, MIC_Slow_Fast_Timer_n, MIC_Slow_Next_Timer_n, MIC_TM_Threshold_n, MIC_XIO_PTCAL_DATA_n, MIC_Trcd_Pchg_n, MIC_Yreg_Stat_n, Yreg_Init_Cnts_n, Yreg_Init_Ctl_n, Yreg_YDRAM_Dta_n, Yreg_YRAC_Dta_n. – checkstop_fir, recoverable_fir, SPE_available, serial_number, group_control, debug_bus_control, TS_CTSR1, TS_CTSR2, TS_MTSR2, TS_GITR, TM_CR1, TM_CR2, TM_TPR, TM_STR1, TM_STR2, TM_TSR, TBR. – EIB_AC0_CTL, EIB_Cfg – L2_ModeSetup1, NCU_ModeSetup, BIU_ModeSetup1, BIU_ModeSetup2. – MFC_LPID, INT_Route, RA_Group_ID, RA_Enable, DMAC_PMCR, SPU_ECC_Cntl, SPU_ECC_Stat, SPU_ECC_Addr, SPU_ERR_Mask. – IOC_IOIF0_QueThshld, IOC_IOIF1_QueThshld. – TKM_MBAR, TKM_IOIF0_AR, TKM_IOIF1_AR, TKM_PR, TKM_PMCR. • Changes to the following registers/sections: L2_Machchk_en, MFC_CNTL, Ext_tr_timer. • BED I/O MMIO Registers defined as 64-bit registers.
November 9, 2005	Initial release.

Preface

This document describes the fields of the Cell Broadband Engine™ (CBE) registers. This document should be used in conjunction with the *Cell Broadband Engine Architecture* (CBEA) and other supporting documents listed in *Related Publications*.

Who Should Read This Manual

This manual is intended for designers who plan to develop products using the CBE implementation of the CBEA.

Related Publications

A list of related materials follows.

Title	Version	Date
Cell Broadband Engine Architecture	1.01	October 2006
PowerPC User Instruction Set Architecture, Book I	2.02	January 2005
PowerPC Virtual Environment Architecture, Book II	2.02	January 2005
PowerPC Operating Environment Architecture, Book III	2.02	January 2005
Synergistic Processor Unit Instruction Set Architecture	1.11	October 2006

Conventions and Notation

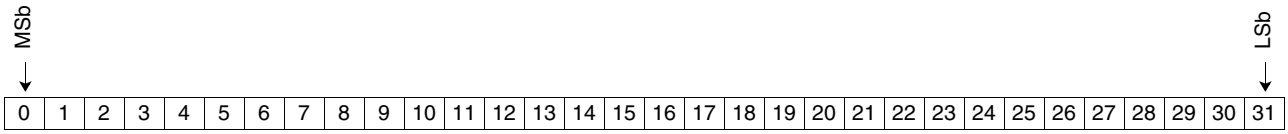
Byte Ordering

Throughout this document, standard IBM big-endian notation is used, meaning that bytes are numbered in ascending order from left to right. Big-endian and little-endian byte ordering are described in the *Cell Broadband Engine Architecture* document.

Note: In this document, storage units are defined as they are defined in the *PowerPC Architecture*. Quadwords are 128 bits, doublewords are 64 bits, words are 32 bits, halfwords are 16 bits, and bytes are 8 bits.

Bit Ordering

Bits are numbered in ascending order from left to right with bit 0 representing the most significant bit (MSb) and bit 31 the least significant bit (LSb).



Bit Encoding

Numbers are generally shown in decimal format, unless designated as follows:

- Hexadecimal values are preceded by an “x” and enclosed in single quotation marks.
  For example: x‘0A00’.
- Binary values in sentences are shown in single quotation marks.
  For example: ‘1010’.

The binary point for fixed-point format data is at the right end of the field or value. Operations are performed with the binary points aligned, even if the fields are of different widths.

Software Documentation Conventions

The following software documentation conventions are used in this manual:

- Commands and instruction names are written in **bold** type. For example: **put**.
- Transactions are capitalized and are not bold to distinguish them from instructions. For example: The intent of the enforce in-order execution of I/O (EIEIO) transaction is to act as a barrier for two groups of transactions in support of the PowerPC Architecture **eieio** instruction.
- Variables are written in italic type. Required parameters are enclosed in angle brackets. Optional parameters are enclosed in brackets. For example: **get*<f,b>[s]***.
- I/O signal names are in upper case.

Referencing Registers, Fields, and Bit Ranges

Registers are referenced by their full name or by their short name (also called the register mnemonic). Fields are referenced by their field name or by their bit position. The following table describes how registers, fields, and bit ranges are referenced in this document and provides examples of the references.

Type of Reference	Format	Example
Reference to a specific register and a specific field using the register short name and the field name.	Register_Short_Name[Field_Name]	MSR[R]
Reference to a field using the field name.	[Field_Name]	[R]
Reference to a specific register and to multiple fields using the register short name and the field names.	Register_Short_Name[Field_Name1, Field_Name2]	MSR[FE0, FE1]
Reference to a specific register and to multiple fields using the register short name and the bit positions.	Register_Short_Name[Bit_Number, Bit_Number]	MSR[52, 55]
Reference to a specific register and to a field using the register short name and the bit position or the bit range.	Register_Short_Name[Bit_Number]	MSR[52]
	Register_Short_Name[Starting_Bit_Number:Ending_Bit_Number]	MSR[39:44]
A field name followed by an equal sign (=) and a value indicates the value for that field.	Register_Short_Name[Field_Name] = *n*[1]	MSR[FE0] = ‘1’ MSR[FE] = x‘1’
	Register_Short_Name[Bit_Number] = *n*[1]	MSR[52] = ‘0’ MSR[52] = x‘0’
	Register_Short_Name[Starting_Bit_Number:Ending_Bit_Number] = *n*[1]	MSR[39:43] = ‘10010’ MSR[39:43] = x‘11’

1. Where *n* is the binary or hexadecimal value for the field or bits specified in the brackets.

1. Cell Broadband Engine Memory-Mapped I/O Registers

This section defines the memory map for the memory-mapped I/O (MMIO) registers in the Cell Broadband Engine™ (CBE).

While the *Cell Broadband Engine Architecture* (CBEA) defines one base register (BP_Base, which is known as BE_MMIO_Base in the implementation) for relocating the internal registers, the CBE implements BE_MMIO_Base as several base registers that replicate this relocation function for the units, as shown in *Table 1-1*. These base register values are initialized from the configuration ring during the power-on reset (POR) sequence.

The number of bits in these configuration ring fields is also shown in *Table 1-1*. In all cases, these bits correspond to the most significant bit of the 42-bit real-address implemented in the CBE. The most significant 19 bits of all these configuration ring fields should be set to the same value. If a configuration ring field has more than 19 bits, these additional bits should be set to a value consistent with the settings in *Table 1-3* on page 20 for the starting address of that unit. Each Synergistic Processor Element (SPE) memory flow controller (MFC) unit has its own BE_MMIO_Base in the configuration ring, but each unit should be initialized to the same value. The Input/Output Controller (IOC) unit contains one configuration ring field that defines the most significant 22 bits of the MMIO space for multiple units as shown in *Table 1-1*.

The value of BE_MMIO_Base is relocatable, and the value of the most significant 19 bits is not specified in this document.

Table 1-1. Registers That Are Replicated Forms of BE_MMIO_Base

Configuration Ring Field	Sets BE_MMIO_Base for the Units Below
SPE BE_MMIO_Base Address (19 bits)	SPE0-7
PPE BE_MMIO_Base Address (30 bits)	PPE
MIC BE_MMIO _Base Address (30 bits)	MIC
PRV BE_MMIO_Base Address (30 bits)	Pervasive
BEI BE_MMIO_Base Address (22 bits)	IIC, IOC Address Translation, IOC,BIC, and EIB

1.1 Classification of Registers

Registers in the MMIO memory map are classified as either Privilege 1, Privilege 2, or Problem State. These designations relate to a suggested hierarchy of privileged access. Privilege 1 registers are the most privileged. They are intended to be accessed by a hypervisor or firmware operating in the HV = ‘1’ and PR = ‘0’ mode, usually when supporting logical partitioning. Privilege 2 registers are intended for privileged operating system code running in HV = ‘0’ and PR = ‘0’ mode. When no hypervisor is present, firmware and the privileged operating system typically combine Privilege 1 and Privilege 2 resources into one privilege level. Problem State registers may be directly accessible by applications operating at the HV = ‘0’ and PR = ‘1’ modes, should an operating system so choose. Access to MMIO registers in these modes is not directly enforced by hardware. Enforcement of these accesses is left to the operating system and hypervisor developers’ use of the translation facilities of the PPE memory management unit (MMU) when data relocation is active.

1.2 MMIO Access Rules for 32-Bit and 64-Bit Registers

The CBE 32-bit registers must be accessed 32 bits at a time. No accesses are allowed on fewer than 32 bits. In addition, 64-bit access to an address range that includes a 32-bit register is not allowed, unless explicitly specified otherwise.

The CBE 64-bit registers must be accessed either 64 bits or, if allowed, 32 bits at a time. No accesses are allowed on fewer than 32 bits.

Table 1-2 lists the access rules for the 64-bit registers.

Table 1-2. 64-Bit Register Access Rules

Address Space	Doubleword Read (bits [0:63])	High Word Read (bits [0:31])	Low Word Read (bits [32:63])	Doubleword Write (bits [0:63])	High Word Write (bits [0:31])	Low Word Write (bits [32:63])
Problem Space	yes	yes	yes	yes	yes	yes
Privilege with high word reserved and low word defined	yes	no	yes	yes	no	yes
Privilege with high word defined and low word reserved	yes	yes	no	yes	yes	no
Privilege space with both high and low word defined	yes	no	no	yes	no	no

1.3 The MMIO Memory Map

Reserved areas of the MMIO memory map within assigned units, reserved registers, and reserved bits operate in the same way: writes have no effect and reads return zeros. No error flags are set when reserved locations that are assigned to units are written or read. Reserved areas of the MMIO memory map that are not assigned to any unit should not be read from or written to, as doing so causes serious errors in software.

Table 1-3. CBE Memory Map (Page 1 of 3)

BE_MMIO_Base + Offset Range		Area	Size in Hexadecimal	Size in Decimal	Additional Information
Start	End				
x‘0’	x‘03FFFF’	SPE(0) Local Store	x‘40000’	262144	
x‘040000’	x‘05FFFF’	SPE(0) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘060000’	x‘07FFFF’	SPE(0) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘080000’	x‘0BFFFF’	SPE(1) Local Store	x‘40000’	262144	
x‘0C0000’	x‘0DFFFF’	SPE(1) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘0E0000’’	x‘0FFFFF’	SPE(1) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘100000’	x‘13FFFF’	SPE(2) Local Store	x‘40000’	262144	

Table 1-3. CBE Memory Map (Page 2 of 3)

BE_MMIO_Base + Offset Range		Area	Size in Hexadecimal	Size in Decimal	Additional Information
Start	End				
x‘140000’	x‘15FFFF’	SPE(2) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘160000’	x‘17FFFF’	SPE(2) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘180000’	x‘1BFFFF’	SPE(3) Local Store	x‘40000’	262144	
x‘1C0000’	x‘1DFFFF’	SPE(3) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘1E0000’	x‘1FFFFF’	SPE(3) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘200000’	x‘23FFFF’	SPE(4) Local Store	x‘40000’	262144	
x‘240000’	x‘25FFFF’	SPE(4) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘260000’	x‘27FFFF’	SPE(4) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘280000’	x‘2BFFFF’	SPE(5) Local Store	x‘40000’	262144	
x‘2C0000’	x‘2DFFFF’	SPE(5) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘2E0000’	x‘2FFFFF’	SPE(5) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘300000’	x‘33FFFF	SPE(6) Local Store	x‘40000’	262144	
x‘340000’	x‘35FFFF’	SPE(6) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘360000’	x‘37FFFF’	SPE(6) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘380000’	x‘3BFFFF’	SPE(7) Local Store	x‘40000’	262144	
x‘3C0000’	x‘3DFFFF’	SPE (7) Problem State	x‘20000’	131072	*Table 3-3 SPE Problem State Memory Map* on page 50
x‘3E0000’	x‘3FFFFF’	SPE(7) Privilege 2	x‘20000’	131072	*Table 3-2 SPE Privilege 2 Memory Map* on page 49
x‘400000’	x‘401FFF’	SPE(0) Privilege 1	x‘2000’	8192	*Table 3-1 SPE Privilege 1 Memory Map* on page 46
x‘402000’	x‘403FFF’	SPE(1) Privilege 1	x‘2000’	8192	
x‘404000’	x‘405FFF’	SPE(2) Privilege 1	x‘2000’	8192	
x‘406000’	x‘407FFF’	SPE(3) Privilege 1	x‘2000’	8192	
x‘408000’	x‘409FFF’	SPE(4) Privilege 1	x‘2000’	8192	
x‘40A000’	x‘40BFFF’	SPE(5) Privilege 1	x‘2000’	8192	
x‘40C000’	x‘40DFFF’	SPE(6) Privilege 1	x‘2000’	8192	
x‘40E000’	x‘40FFFF’	SPE(7) Privilege 1	x‘2000’	8192	
x‘500000’	x‘500FFF’	PPE Privilege	x‘1000’	4096	*Table 2-1 PPE Privilege MMIO Memory Map* on page 24
x‘501000’	x‘507FFF’	Reserved			

Table 1-3. CBE Memory Map (Page 3 of 3)

BE_MMIO_Base + Offset Range		Area	Size in Hexadecimal	Size in Decimal	Additional Information
Start	End				
x‘508000’	x‘508FFF’	IIC	x‘1000’	4096	*Table 6-1 IIC Memory Map* on page 145
x‘509000’	x‘5093FF’	Reserved			
x‘509400’	x‘5097FF’	Pervasive: Performance Monitor	x‘400’	1024	*Table 11-1 Pervasive Registers* on page 237
x‘509800’	x‘509BFF’	Pervasive: Thermal and Power Management	x‘400’	1024	*Table 11-1 Pervasive Registers* on page 237
x‘509C00’	x‘509FFF’	Pervasive: RAS	x‘400’	1024	*Table 11-1* on page 237
x‘50A000’	x‘50AFFF’	MIC and TKM	x‘1000’	4096	*Table 8-1 TKM MMIO Memory Map* on page 207
x‘50B000’	x‘50FFFF’	Reserved			
x‘510000’	x‘510FFF’	IOC Address Translation	x‘1000’	4096	*Table 5-1 IOC Address Translation MMIO Memory Map* on page 133
x‘511000’	x‘5113FF’	BIC 0 NClk	x‘400’	1024	
x‘511400’	x‘5117FF’	BIC 1 NClk	x‘400’	1024	
x‘511800’	x‘511BFF’	EIB	x‘400’	1024	*Table 10-1 EIB MMIO Memory Map* on page 227
x‘511C00’	x‘511FFF’	IOC I/O Command	x‘400’	1024	*Table 4-1 BEI IOC MMIO Memory Map* on page 121
x‘512000’	x‘512FFF’	BIC 0 BClk	x‘1000’	4096	
x‘513000’	x‘513FFF’	BIC 1 BClk	x‘1000’	4096	
x‘514000’	x‘514FFF’	Reserved	x‘1000’	4096	
x‘515000’	x‘7FFFFF’	Reserved			

2. PowerPC Processor Element (PPE) MMIO Registers

This section describes the PPE memory-mapped I/O (MMIO) registers. *Table 2-1* on page 24 shows the PPE MMIO memory map and lists the PPE registers. The PPE register space starts at x‘500 000’ and ends at x‘500 FFF’. Offsets are from the start of the PPE privilege area. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Registers that have no implementation-specific bit definitions (all bits are as defined in the architecture) and have initial power-on reset (POR) values scanned to all zeros show only a cross-reference to *Appendix A Registers Defined in the CBEA* on page 335.
- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

Table 2-1. PPE Privilege MMIO Memory Map (Page 1 of 2)

Hexadecimal Offset (x'500 *nnn*')	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
Level 2 (L2) Cache MMIO Registers				
x'300'	*L2 RMT Index Register (L2_RMT_Index)*	64	R/W	Not implemented
x'310'	*L2 RMT Data Register (L2_RMT_Data)*	64	R/W	*Section 2.1.1* on page 26
x'800' x'810' x'820' x'808' x'818' x'828' x'830'	*L2 Fault Isolation Register (L2_FIR)* *L2 Fault Isolation Register Set (L2_FIR_Set)* *L2 Fault Isolation Register Reset (L2_FIR_Reset)* *L2 Fault Isolation Register Error Mask (L2_FIR_Err)* *L2 Fault Isolation Register Error Mask Set (L2_FIR_Err_Set)* *L2 Fault Isolation Register Error Mask Reset (L2_FIR_Err_Reset)* *L2 Fault Isolation Register Checkstop Enable (L2_FIR_ChkStopEnbl)*	64	R/W	*Section 2.1.2* on page 27
x'838'	Reserved			
x'858'	*L2 Mode Setup Register 1 (L2_ModeSetup1)*	64	R/W	*Section 2.1.3* on page 31
x'858'	Reserved			
x'870'	*L2 Machine Check Enable Register (L2_Machchk_en)*	64	R/W	*Section 2.1.4* on page 33
x'878'	Reserved			
Core Interface Unit (CIU) MMIO Registers				
x'900' x'910' x'920' x'908' x'918' x'928' x'930'	*CIU Fault Isolation Register (CIU_FIR)* *CIU Fault Isolation Register Set (CIU_FIR_Set)* *CIU Fault Isolation Register Reset (CIU_FIR_Reset)* *CIU Fault Isolation Register Error Mask (CIU_FIR_Err)* *CIU Fault Isolation Register Error Mask Set (CIU_FIR_Err_Set)* *CIU Fault Isolation Register Error Mask Reset (CIU_FIR_Err_Reset)* *CIU Fault Isolation Register Checkstop Enable (CIU_FIR_ChkStpEnbl)*	64	R/W	*Section 2.2.1* on page 34
x'938'	*CIU Enable Recoverable Error Register (CIU_ERE)*	64	R/W	*Section 2.2.2* on page 36
x'940'	*CIU Local Recoverable Error Counter Register (CIU_REC)*	64	R/W	*Section 2.2.3* on page 37
x'948'	*CIU Mode Setup Register (CIU_ModeSetup)*	64	R/W	*Section 2.2.4* on page 38
x'958' – x'A60'	Reserved			
Noncacheable Unit (NCU) MMIO Registers				
x'A48'	*NCU Mode Setup Register (NCU_ModeSetup)*	64	R/W	*Section 2.3.1* on page 40
x'A58' – x'A60'	Reserved			

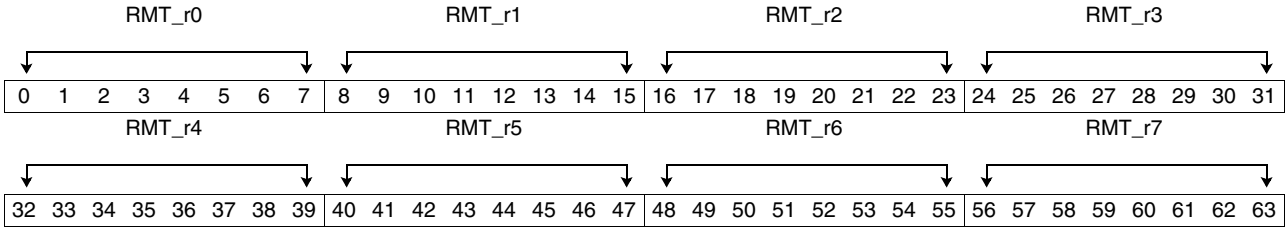
Table 2-1. PPE Privilege MMIO Memory Map (Page 2 of 2)

Hexadecimal Offset (x‘500 *nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
Bus Interface Unit (BIU) MMIO Registers				
x‘B00’ x‘B10’ x‘B20’ x‘B08’ x‘B18’ x‘B28’ x‘B30’	*BIU Fault Isolation Register (BIU_FIR)* *BIU Fault Isolation Register Set (BIU_FIR_Set)* *BIU Fault Isolation Register Reset (BIU_FIR_Reset)* *BIU Fault Isolation Register Error Mask (BIU_FIR_Err)* *BIU Fault Isolation Register Error Mask Set (BIU_FIR_Err_Set)* *BIU Fault Isolation Register Error Mask Reset (BIU_FIR_Err_Reset)* *BIU Fault Isolation Register Checkstop Enable (BIU_FIR_ChkStpEnbl)*	64	R/W	*Section A.11* on page 345
x‘B48’	*BIU Mode Setup Register 1 (BIU_ModeSetup1)*	64	R/W	*Section 2.4.1* on page 41
x‘B50’	*BIU Mode Setup Register 2 (BIU_ModeSetup2)*	64	R/W	*Section 2.4.2* on page 42
x‘B60’ x‘B68’ x‘B70’	*BIU Reserved Registers 1-3 (BIU_Reserved_n)*	64	R/W	*Section 2.4.3* on page 43

2.1 L2 MMIO Registers

2.1.1 L2 RMT Data Register (L2_RMT_Data)

Register Short Name	L2_RMT_Data	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500310’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	L2



Bits	Field Name	Description
0:7	RMT_r0	Replacement management table (RMT) row 0 (classID = 000)
8:15	RMT_r1	RMT row 1 (classID = 001)
16:23	RMT_r2	RMT row 2 (classID = 010)
24:31	RMT_r3	RMT row 3 (classID = 011)
32:39	RMT_r4	RMT row 4 (classID = 100)
40:47	RMT_r5	RMT row 5 (classID = 101)
48:55	RMT_r6	RMT row 6 (classID = 110)
56:63	RMT_r7	RMT row 7 (classID = 111)

Programming Note: For each RMT row, a field value of ‘00000000’ is treated logically as ‘11111111’. In other words, disabling all sets in an RMT row results in those sets being treated as if they were enabled. This is not allowed.

2.1.2 L2 Fault Isolation, Error Mask, Checkstop Enable Registers

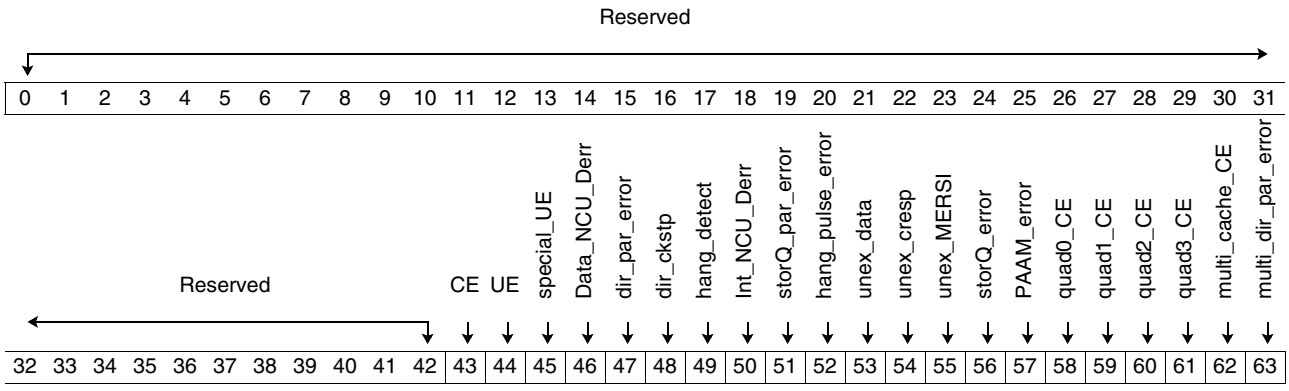
This section describes the L2 fault isolation registers (FIRs).

Note: The L2_FIR_Reset and the L2_FIR_Err_Reset Registers have a value at initial POR set to x'00000000_001FFFFF'. The value listed in the table below is for the remaining L2 Fault Isolation, Error Mask, and Checkstop Enable Registers.

Register Short Name	Register Name
L2_FIR	L2 Fault Isolation Register
L2_FIR_Set	L2 Fault Isolation Register Set
L2_FIR_Reset	L2 Fault Isolation Register Reset
L2_FIR_Err	L2 Fault Isolation Register Error Mask
L2_FIR_Err_Set	L2 Fault Isolation Register Error Mask Set
L2_FIR_Err_Reset	L2 Fault Isolation Register Error Mask Reset
L2_FIR_ChkStpEnbl	L2 Fault Isolation Register Checkstop Enable

Register Short Name	See table above	**Privilege Type**	Privilege 1
Access Type	See table below	**Width**	64 bits
Hex Offset From BE_MMIO_Base	See table below	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	L2

Register Short Name	Hex Offset From BE_MMIO_Base	Access Type	FIR Function
L2_FIR	x'500800'	MMIO Read Only	FIR Read
L2_FIR_Set	x'500810'	MMIO Write Only	FIR Set
L2_FIR_Reset	x'500820'	MMIO Write Only	FIR Reset
L2_FIR_Err	x'500808'	MMIO Read Only	Error Mask Read
L2_FIR_Err_Set	x'500818'	MMIO Write Only	Error Mask Set
L2_FIR_Err_Reset	x'500828'	MMIO Write Only	Error Mask Reset
L2_FIR_ChkStpEnbl	x'500830'	MMIO Read/Write	Checkstop Enable



Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
0:42	Reserved	Bits are not implemented; all bits read back zero.	0	0
43	CE	L2 Cache correctable error (CE). When a correctable data error is detected on a processor request (instruction or data type of transfer), the data is corrected before it is forwarded to the processor. The corrected data is written back to the L2 cache. When a correctable data error is detected on a castout request as the result of a flush, a read with intent to modify (RWITM) transaction, or the victim of a replacement, the data is corrected before it is forwarded to the bus interface unit (BIU).	0	0
44	UE	L2 Cache uncorrectable error (UE). When an uncorrectable error is detected in the L2 cache in response to a processor load request, a UE response is sent to the requesting PowerPC processor unit (PPU). When a store hits in an L2 cache line that contains the UE data, the store is merged into the line (timing does not permit discarding the store). An altered uncorrectable error (AUE) error correction code (ECC) is written to distinguish a data error written back to the L2 cache from an error passed from another source. When an uncorrectable error is detected in the L2 cache in response to a castout operation, a data error (DERR) is sent to the BIU, and a special UE (SUE) is sent to memory.	0	1
45	special_UE	L2 cache special UE. BIU-to-L2 DERR. Cacheable side (I = ‘0’). Load miss data from the BIU has DERR active. The L2 updates the cache with an SUE ECC. For a PPU load request, the uncorrectable data is passed to the PPU with a UE indicator. (Pass Through Error) **Note:** The caching-Inhibited (I) attribute bit for the instruction or data address in memory indicates whether requests are cacheable (I =’0') and sent to the L2 or noncacheable (I = ‘1’) and sent to the NCU.	0	1
46	Data_NCU_Derr	Noncacheable side (I = ‘1’) load data request (versus instruction). BIU-to-L2 DERR. The NCU load data from the BIU has DERR active. This error condition is normally a machine check case, but this FIR bit allows it to be a checkstop case if enabled using Checkstop Enable. (Pass Through Error)	1	0
47	dir_par_error	L2 directory parity error. The failing directory is refreshed with the contents of the other directory.	0	0
48	dir_ckstp	L2 directory checkstop. When set, indicates that an L2 directory parity error has been detected on both halves of the directory (snoop and processor). (System Checkstop)	0	1
49	hang_detect	L2 hang detection of various finite state machines (FSMs). The FSMs include: L2 read/claim (RC), castout (CO), snoop (SNP), and noncacheable control (Ncctl) hang detection. When set, indicates that there was no response to a memory read request. **Livelock Resolution Mode:** Error Mask = 0, Checkstop = 0	0	1
50	Int_NCU_Derr	Noncacheable side (I = ‘1’) load instruction request (versus data). BIU-to-L2 DERR. An NCU instruction load from the BIU has DERR active. The L2 passes the uncorrectable data to the PPU. (Pass Through Error)	0	1

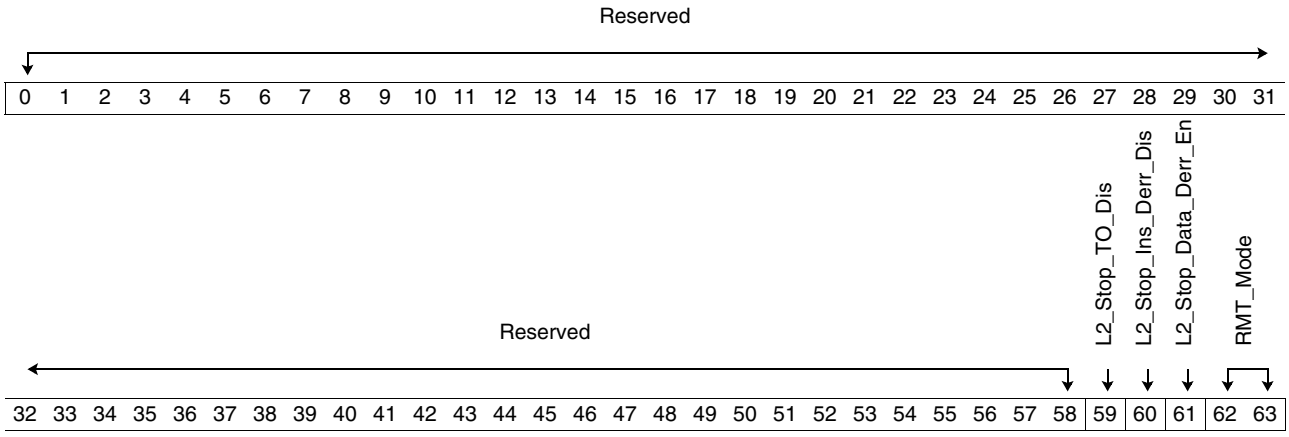
Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
51	storQ_par_error	L2 store queue data parity error in the growable register file (GRF). There are eight parity error signals (data[0-7]_par_err): one for each doubleword of the 64-byte store request to the Read/Claim (RC) machine. This bit is set whenever one of the parity error signals is set.	0	1
52	hang_pulse_error	L2 RC or Ncctl hang waiting for data from the BIU. The L2 RC or Ncctl was waiting for a data tag match and detected two hang pulses. **Livelock Resolution Mode:** Error Mask = 0, Checkstop Enable = 0	0	1
53	unex_data	L2 RC or Ncctl unexpected data. The L2 RC or Ncctl received a data tag match that the RC or Ncctl machine was not expecting (for example, too many data valids [DVALs]).	0	1
54	unex_cresp	L2 RC unexpected combined response (cresp). The L2 RC received an invalid cresp for an RC bus operation.	0	1
55	unex_MERSI	L2 RC unexpected MERSI state. The L2 RC machine read an invalid Modified, Exclusive, Recent, Shared, Invalid (MERSI) state out of the L2 directory.	0	1
56	storQ_error	L2 store queue internal control error. This bit is set as a result of the following types of errors: • Store queue overflow: sqctl_cfg_stq_overflow_err_dis • Incorrect first or second quadword command sequence: sqctl_cfg_stq_stqw_seq_err_dis	0	1
57	PAAM_error	L2 Snoop PAAM error. A new snoop request violated the previous adjacent address match (PAAM) window of an active snoop request or of a busy local RC waiting for a combined response. This bit is set as a result of the following types of errors: • rcx_paam_violation: sndsp_cfg_rc_paam_err_dis • snp_collided_paam: sndsp_cfg_snp_paam_err_dis **Note:** If the EIB_AC0CTL configuration bits [13:15] are set to 0, 1, 2, 3, 6, or 7, then the Element Interconnect Bus (EIB) can reflect commands inside the PAAM window and force them to be retried. (The values 4 and 5 are both DD1 mode-no PAAM violations period.) The default value is 6. For EIB modes 0, 1, 2, 3, 6, and 7, the L2 PAAM violation FIR bit must be masked, and the Checkstop Enable must be '0'. For EIB modes 4 and 5, the L2 PAAM violation FIR bit must not be masked, and the Checkstop Enable must be set to '1'. For support of default EIB settings, Error Mask = 1, Checkstop Enable = 0.	1	0
58	quad0_CE	Quadrant0 CE threshold. Multiple cache CEs were detected during a hang pulse duration.	1	0
59	quad1_CE	Quadrant1 CE threshold. Multiple cache CEs were detected during a hang pulse duration.	1	0
60	quad2_CE	Quadrant2 CE threshold. Multiple cache CEs were detected during a hang pulse duration.	1	0
61	quad3_CE	Quadrant3 CE threshold. Multiple cache CEs were detected during a hang pulse duration.	1	0
62	multi_cache_CE	Multiple cache CEs. Multiple cache CEs were detected during a hang pulse duration from more than one quadrant.	1	0

Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
63	multi_dir_par_error	Multiple directory parity errors. Multiple directory parity errors were detected within a period of two hang pulses. This indicates that the array error is not an intermittent fault. Because the memory flow controller (MFC) cannot make forward progress with a stuck fault in the directory, the system is checkstopped.	0	1

2.1.3 L2 Mode Setup Register 1 (L2_ModeSetup1)

This register is used to set up and control various parts of the L2. L2_ModeSetup1[62:63] are used to select the RMT mode.

Register Short Name	L2_ModeSetup1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'500848'	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	L2

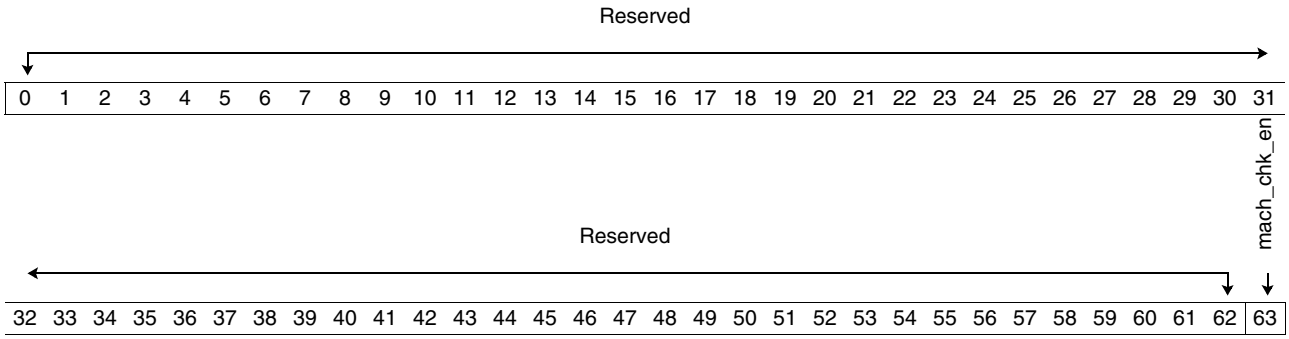


Bits	Field Name	Description
0:58	Reserved	Bits are not implemented; all bits read back zero.
59	L2_Stop_TO_Dis	L2 stops on noncacheable unit (NCU) data timeout. When this stop occurs, the L2_FIR register bit [52] gets set corresponding to the NCU data timeout, and a checkstop occurs. If this bit is set to '1', then the L2 goes through recovery, sets the L2_FIR bit [46], and sends a machine check. 0 Enabled. L2 stops and holds its internal state if an NCU data timeout occurs. 1 Disabled. L2 state machine continues as normal on an NCU data timeout.
60	L2_Stop_Ins_Derr_Dis	L2 stops on a Data_err on an NCU instruction fetch. When this stop occurs, the L2_FIR register bit [50] is set corresponding to the NCU instruction fetch on a Data_err, and a checkstop occurs. If this bit is set to '1', then the L2 goes through recovery and sets the L2_FIR bit [50]. 0 Enabled. L2 stops and holds its internal state on a Data_err on an NCU instruction fetch. 1 Disabled. L2 state machine continues normal operation on a noncacheable instruction fetch Data_err.
61	L2_Stop_Data_Derr_En	The default setting for this bit is for the L2 to continue normal operation when a Data_err on an NCU data fetch occurs, set the L2_FIR register bit [46], and a machine check occurs. If this bit is set to '1', then the L2_FIR register bit [46] corresponding to a Data_err on an NCU data fetch gets set, a machine check occurs, and the L2 stops (nonrecoverable error). 0 Disabled. L2 state machine continues normal operation on a Data_err for an NCU Data fetch. 1 Enabled. L2 stops and holds the internal state when a Data_err for an NCU Data fetch occurs.

Bits	Field Name	Description
62:63	RMT_Mode	Select the RMT Mode. 00 Pseudo-RMT Mode. 01 Not allowed. Produces undefined results. 10 Binary least recently used (LRU) RMT mode. 11 Direct-mapped mode.

2.1.4 L2 Machine Check Enable Register (L2_Machchk_en)

Register Short Name	L2_Machchk_en	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500870’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	L2



Bits	Field Name	Description
0:62	Reserved	Bits are not implemented; all bits read back zero.
63	mach_chk_en	Machine check enable. 0 Recoverable error is reported if L2_FIR[46] (Data_NCU_Derr bit) is set. 1 Machine check occurs if L2_FIR[46] (Data_NCU_Derr bit) is set. **Note:** If L2_FIR[46] is set, a machine check interrupt (PPU interrupt vector x‘200’) occurs regardless of the setting of this bit.

2.2 CIU MMIO Registers

2.2.1 CIU Fault Isolation, Error Mask, Checkstop Enable Registers

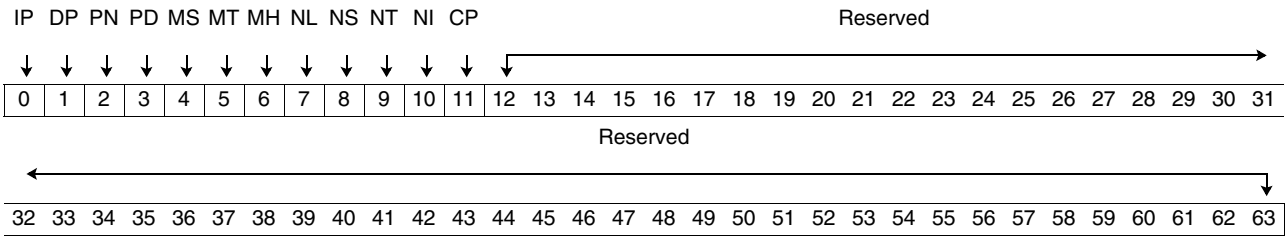
This section describes the CIU Fault Isolation Registers (FIRs).

Note: The CIU_FIR_Reset and the CIU_FIR_Err_Reset Registers have a value at initial POR set to x'FFF00000_00000000. The value listed in the table below is for the remaining CIU Fault Isolation, Error Mask, and Checkstop Enable Registers.

Register Short Name	Register Name
CIU_FIR	CIU Fault Isolation Register
CIU_FIR_Set	CIU Fault Isolation Register Set
CIU_FIR_Reset	CIU Fault Isolation Register Reset
CIU_FIR_Err	CIU Fault Isolation Register Error Mask
CIU_FIR_Err_Set	CIU Fault Isolation Register Error Mask Set
CIU_FIR_Err_Reset	CIU Fault Isolation Register Error Mask Reset
CIU_FIR_ChkStpEnbl	CIU Fault Isolation Register Checkstop Enable

Register Short Name	See table above	**Privilege Type**	Privilege 1
Access Type	See table below	**Width**	64 bits
Hex Offset From BE_MMIO_Base	See table below	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	CIU

Register Short Name	Hex Offset From BE_MMIO_Base	Access Type	FIR Function
CIU_FIR	x'500900'	MMIO Read Only	FIR Read
CIU_FIR_Set	x'500910'	MMIO Write Only	FIR Set
CIU_FIR_Reset	x'500920'	MMIO Write Only	FIR Reset
CIU_FIR_Err	x'500908'	MMIO Read Only	Error Mask Read
CIU_FIR_Err_Set	x'500918'	MMIO Write Only	Error Mask Set
CIU_FIR_Err_Reset	x'500928'	MMIO Write Only	Error Mask Reset
CIU_FIR_ChkStpEnbl	x'500930'	MMIO Read/Write	Checkstop Enable



Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
0	IP	PowerPC processor unit (PPU) instruction cache parity error Recoverable instruction cache parity error	0	0
1	DP	PPU data cache parity error Recoverable data cache parity error	0	0
2	PN	PPU nonrecoverable error PPE nonrecoverable error	0	1
3	PD	PPU debug checkstop IU debug checkstop **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	1
4	MS	Memory management unit (MMU) segment lookaside buffer (SLB) parity error MMU nonrecoverable error An SLB translate or read operation has encountered a parity error.	0	1
5	MT	MMU translation lookaside buffer (TLB) parity error MMU nonrecoverable error A TLB translate or read operation has encountered a parity error.	0	1
6	MH	MMU load or store hung MMU nonrecoverable error A page table entry (PTE) load or store request has been pending without a response for too long. **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	1
7	NL	NCU load time out The load-done signal has not been asserted for an outstanding caching-inhibited load or caching-inhibited fetch during a hang-pulse duration. **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	1
8	NS	NCU store time out One of the store-done signals has not been asserted for an outstanding caching-inhibited store, SYNC, EIEIO, TLBIE, ICBI, or TLBSYNC during a hang-pulse duration. **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	1
9	NT	NCU TLBIQ time out One of the TLBIQ state machines has been stacked during a hang-pulse duration. **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	1
10	NI	NCU ICBIQ time out One of the ICBIQ state machines has been stacked during a hang-pulse duration. **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	1
11	CP	CIU data prefetch time out The data-prefetch-done signal has not been asserted for an outstanding data prefetch during a hang-pulse duration. **Livelock resolution mode:** Error Mask = 0, Checkstop Enable = 0	0	0
12:63	Reserved	Bits are not implemented; all bits read back zero.	0	0

2.2.2 CIU Enable Recoverable Error Register (CIU_ERE)

Register Short Name	CIU_ERE	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500938’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	CIU

Bits	Field Name	Description
0	IP	Enable a recoverable instruction-cache parity error to increment the local recoverable-error counter.
1	DP	Enable a recoverable data-cache parity error to increment the local recoverable-error counter.
2:63	Reserved	Bits are not implemented; all bits read back zero.

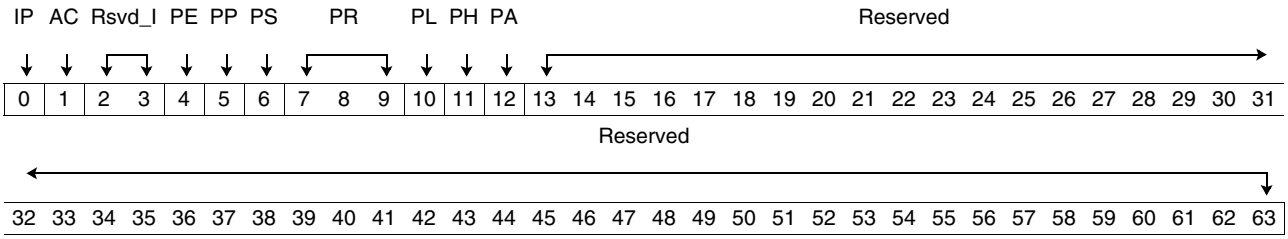
2.2.3 CIU Local Recoverable Error Counter Register (CIU_REC)

Register Short Name	CIU_REC	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500940’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	CIU

Bits	Field Name	Description
0:5	RC	Local recoverable-error counter for the PPU Counts the PPU recoverable errors (L1 instruction-cache parity errors and L1 data-cache parity errors) until the counter reaches its maximum value (all ones). At that time, the local carry signal is asserted for the test control unit (TCU). The error signals are converted to NClk/2 signals and ORed so that they can be counted. Thus, there can be a miscount. If there is a checkstop error, the counter freezes. The TCU can reset this counter to zero. The register counts up to 63 errors and freezes there until reset by the TCU or preset by the MMIO.
6:63	Reserved	Bits are not implemented; all bits read back zero.

2.2.4 CIU Mode Setup Register (CIU_ModeSetup)

Register Short Name	CIU_ModeSetup	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500948’	**Memory Map Area**	PPE Privilege
Value at Initial POR	x‘01000000_00000000’	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	CIU



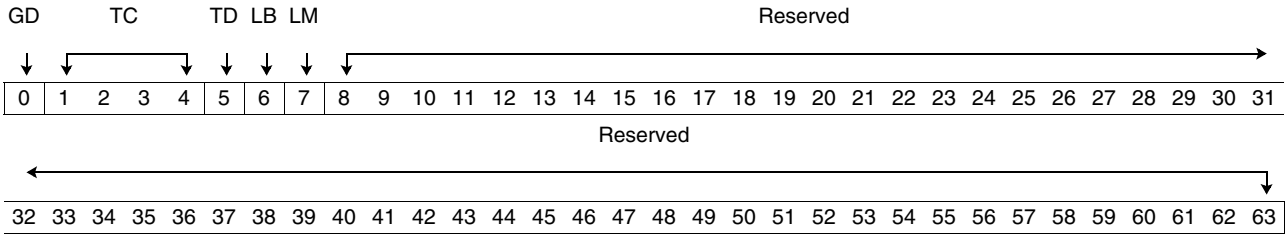
Bits	Field Name	Description
0	IP	CIU Load Queue (CLQ) Instruction High-Priority Mode. The CIU arbiter normally gives a higher priority to data load requests. 0 The arbiter grants data loads higher priority than instruction fetches. 1 The arbiter grants instruction fetches higher priority than data loads. **Note:** This bit must be set to ‘0’ if both threads on the processor run in caching-inhibited space and both threads issue caching-inhibited instruction fetches.
1	AC	CLQ Always-Correct Mode When this bit is set, the CIU forces active the always-correct signal to the L2 for each request, which puts the L2 into Always-Correct mode. 0 Pass to L2 from requester 1 Force active to L2
2:3	Rsvd_I	Reserved. Latch bits are implemented; value read is the value written.
4	PE	Data Prefetch Enable 0 Treat data prefetch as a no-op. 1 Enable data prefetch To disable data prefetch, deactivate all data prefetch streams before resetting this bit.
5	PP	Data Prefetch 4 KB Page Size 0 Set the read size (0 to 1) from the PPE. 1 Limit data prefetch size to a maximum of 4 KB and stop at the 4 KB boundary.
6	PS	Data Prefetch 4/8 Disable The prefetch queue has eight entries to support the maximum number of streams. The entries are divided to support two threads. When both threads are running, each gets four prefetches. If only one thread is running, it gets eight prefetches (both halves). When this bit is set, only four entries are allocated per thread, regardless of the number of active threads. 0 Eight entries for a single thread 1 Four entries for a single thread
7:9	PR	Prefetch Outstanding Request Limit [0:2] This sets the limit for how many outstanding instruction or data prefetch requests can be in the L2. It sets the maximum numbers of L2 RC state machine resources for instruction or data prefetch. The default is ‘100’ (4). If the Prefetch Outstanding Request Limit 4 is set (CIU_ModeSetup[10] = ‘1’), then setting this bit has no effect. The values from ‘001’ (1) to ‘110’ (6) are valid for this limit number. In Data Prefetch Permanent High-Priority Mode (CIU_ModeSetup[11] = ‘1’), this limit value must be less than ‘110’ (6).

Bits	Field Name	Description
10	PL	Prefetch Outstanding Request Limit 4 0 Enables the prefetch outstanding request limit. 1 Disables the prefetch outstanding request limit and forces the count to four.
11	PH	Data Prefetch Permanent High-Priority Mode If this bit is set, the priorities of load-type requests for L2 are changed to give higher priority to the data prefetch request. 0 Data prefetch low-priority mode: MMU > Demand Load > Demand Fetch > Instruction Prefetch > Data Prefetch. If the instruction high-priority mode is set, Demand Load and Demand Fetch are swapped. 1 Data prefetch high-priority mode: MMU > Data Prefetch > Demand Load > Demand Fetch > Instruction Prefetch. If the instruction high-priority mode is set, Demand Load and Demand Fetch are swapped. In the data prefetch permanent high-priority mode, the prefetch outstanding request limit must be less than '110' (6).
12	PA	Data Prefetch Periodical High-Priority Mode This bit has no effect if PH (CIU_ModeSetup[11] = '1') is set. If the PA (CIU_ModeSetup[12] = '1') is set, the priorities of load-type requests for L2 are changed to give higher priority to the data prefetch request. This only occurs after the CIU issues 31 complete load-type requests to the L2 (except for the data prefetch from the last completed request of data prefetch) and the L2 acknowledges these requests. The changed priorities for PA (CIU_ModeSetup[12] = '1') are as follows: MMU > Data Prefetch > Demand Load > Demand Fetch > Instruction Prefetch
13:63	Reserved	Bits are not implemented; all bits read back zero.

2.3 NCU MMIO Registers

2.3.1 NCU Mode Setup Register (NCU_ModeSetup)

Register Short Name	NCU_ModeSetup	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500A48’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	NCU

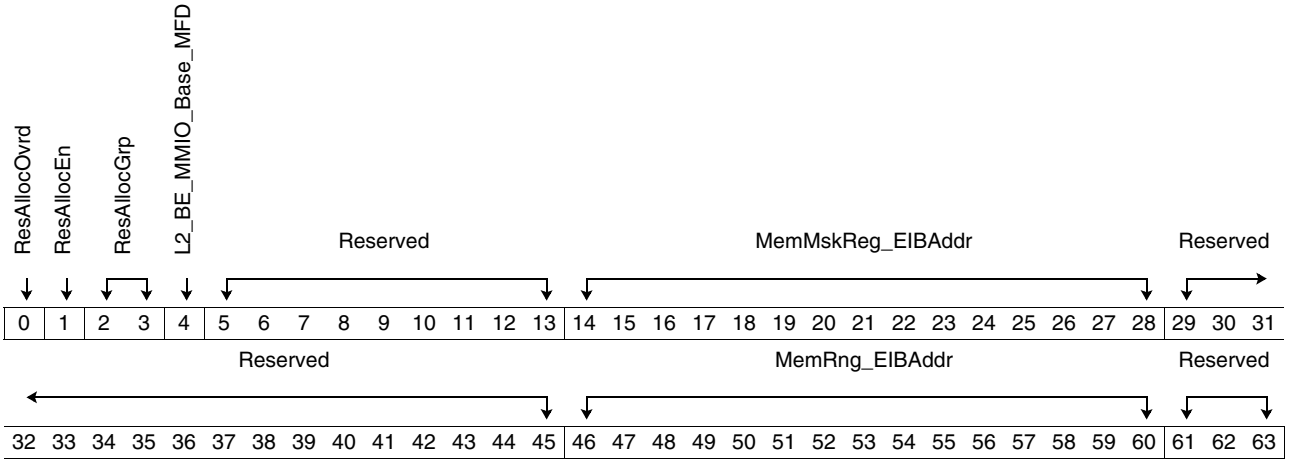


Bits	Field Name	Description
0	GD	Store-gather disable 0 Enable store gathering 1 Disable store gathering
1:4	TC	Store-gather timeout programmable count (4 bits) These are the higher-order bits of the preset value of the store-gather timeout counter.
5	TD	Store-gather timeout disable 0 Enable store-gather timeout 1 Disable store-gather timeout
6	LB	**lwsync** bus operation 0 NCU does not issue any bus operations to the element interconnect bus (EIB) for **lwsync**. 1 NCU issues either SYNC or EIEIO to the EIB for **lwsync**.
7	LM	**lwsync** mapping If the **lwsync** bus operation LB bit (NCU_ModeSetup[6]) is not set, this bit does not take effect. 0 NCU issues a SYNC to the EIB for **lwsync**. 1 NCU issues an EIEIO to the EIB for **lwsync**.
8:63	Reserved	Bits are not implemented; all bits read back zero.

2.4 BIU MMIO Registers

2.4.1 BIU Mode Setup Register 1 (BIU_ModeSetup1)

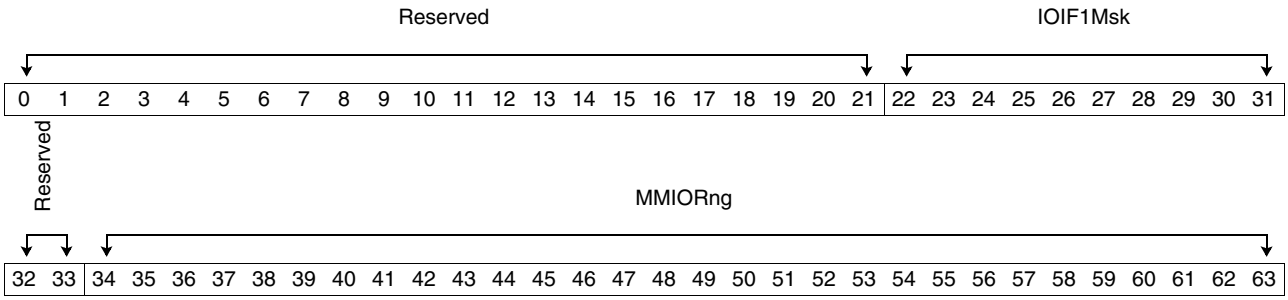
Register Short Name	BIU_ModeSetup1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500B48’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BIU



Bits	Field Name	Description
0	ResAllocOvrd	Resource Allocation Override. Read only. **Note:** This bit is a copy of bzf_bzl_ra_override in the BIU Configuration Scan Register.
1	ResAllocEn	Resource Allocation Enable 0 Disable 1 Enable
2:3	ResAllocGrp	Resource Allocation Group ID
4	L2_BE_MMIO_Base_MFD	L2 BE_MMIO_Base Match Filter Disable 0 Enabled. The BIU does not send a bus-read or bus-write reflected command to the L2 if its address is inside the BE_MMIO_Base range. BE_MMIO_Base is architecturally defined as noncoherent. 1 Disabled. If this bit is ‘1’ and M = ‘1’, the BIU sends a bus-read or bus-write reflected command to the L2, regardless of its address.
5:13	Reserved	Bits are not implemented; all bits read back zero.
14:28	MemMskReg_EIBAddr	Memory Mask Register bits [0:14] AND mask for EIB address bits [22:36]
29:45	Reserved	Bits are not implemented; all bits read back zero.
46:60	MemRng_EIBAddr	Memory range bits [0:14]. Compare to the EIB address [22:36].
61:63	Reserved	Bits are not implemented; all bits read back zero.

2.4.2 BIU Mode Setup Register 2 (BIU_ModeSetup2)

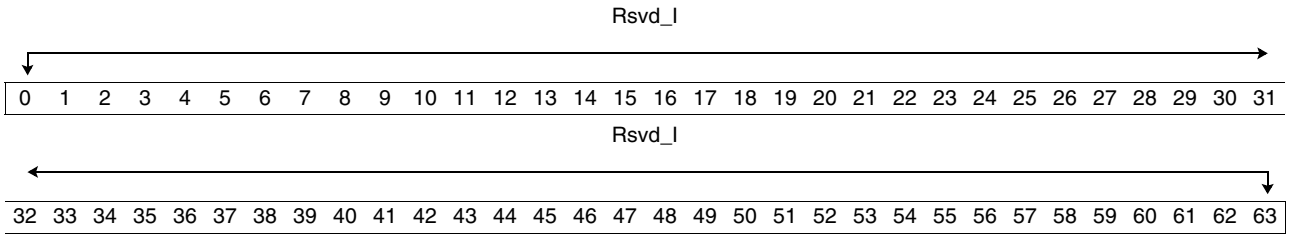
Register Short Name	BIU_ModeSetup2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500B50’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BIU



Bits	Field Name	Description
0:21	Reserved	Bits are not implemented; all bits read back zero.
22:31	IOIF1Msk	IOIF1 Mask
32:33	Reserved	Bits are not implemented; all bits read back zero.
34:63	MMIORng	MMIO range. Read only. **Note:** This is a copy of MMIO range0 scan configuration ring register (bz_bb_range0[22:51]).

2.4.3 BIU Reserved Registers 1-3 (BIU_Reserved_n)

Register Short Name	BIU_Reserved_1 BIU_Reserved_2 BIU_Reserved_3	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘500B60’ x‘500B68’ x‘500B70’	**Memory Map Area**	PPE Privilege
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BIU



Bits	Field Name	Description
0:63	Rsvd_I	Reserved. Latch bits are implemented; value read is the value written.

3. Synergistic Processor Element MMIO Registers

This section describes the SPE memory-mapped I/O (MMIO) registers. Registers and areas marked as implementation specific are not part of the architecture. The detailed description of each register includes the complete hexadecimal offset from BE_MMIO_Base. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

For a list of SPE memory-mapped control registers, see the following sections:

- *Section 3.1.1 SPE Privilege 1 Memory Map*
- *Section 3.1.2 SPE Privilege 2 Memory Map*
- *Section 3.1.3 SPE Problem State Memory Map*

The following notes apply to the register bit definitions:

- Registers that have no implementation-specific bit definitions (all bits are as defined in the architecture) and have initial power-on reset (POR) values scanned to all zeros show only a cross-reference to *Appendix A Registers Defined in the CBEA* on page 335.
- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

3.1 SPE Memory Map and Summary Tables

The detailed description of each register includes the complete hexadecimal offset from BE_MMIO_Base. The summary tables that follow list the registers by memory map area; therefore, only the last four digits of the offset are included.

3.1.1 SPE Privilege 1 Memory Map

Table 3-1. SPE Privilege 1 Memory Map (Page 1 of 3)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
MFC Registers *(CBEA Architected Registers)*				
x‘0000’	*MFC State Register 1 (MFC_SR1)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0008’	*MFC Logical Partition ID Register (MFC_LPID)*	64	R/W	*Section 3.2.1.1* on page 52
x‘0010’	*SPU Identification Register (SPU_ID)*	64	R	*Section 3.2.1.2* on page 53
x‘0018’	*MFC Version Register (MFC_VR)*	64	R	See *Appendix A Registers Defined in the CBEA*
x‘0020’	*SPU Version Register (SPU_VR)*	64	R	See *Appendix A Registers Defined in the CBEA*
x‘0028’ – x‘00FF’	Reserved			
Interrupt Registers (*CBEA Architected Registers)*				
x‘0100’	*Class 0 Interrupt Mask Register (INT_Mask_class0)*	64	R/W	*Section 3.2.2.1* on page 54
x‘0108’	*Class 1 Interrupt Mask Register (INT_Mask_class1)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0110’	*Class 2 Interrupt Mask Register (INT_Mask_class2)*	64	R/W	*Section 3.2.2.2* on page 55
x‘0118’ – x‘013F’	Reserved			
x‘0140’	*Class 0 Interrupt Status Register (INT_Stat_class0)*	64	R/W	*Section 3.2.2.3* on page 56
x‘0148’	*Class 1 Interrupt Status Register (INT_Stat_class1)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0150’	*Class 2 Interrupt Status Register (INT_Stat_class2)*	64	R/W	*Section 3.2.2.4* on page 57
x‘0158’ – x‘017F’	Reserved			
x‘0180’	*Interrupt Routing Register (INT_Route)*	64	R/W	*Section 3.2.2.5* on page 58
x‘0198’ – x‘01FF’	Reserved			
Atomic Unit Control Registers *(Implementation-Specific Registers)*				
x‘0200’	*MFC Atomic Flush Register (MFC_Atomic_Flush)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0280’	*Resource Allocation Group ID (RA_Group_ID)*	64	R/W	*Section 3.2.3.1* on page 59
x‘0288’	*Resource Allocation Enable Register (RA_Enable)*	64	R/W	*Section 3.2.3.2* on page 60

Table 3-1. SPE Privilege 1 Memory Map (Page 2 of 3)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
Fault Isolation Registers *(Implementation-Specific Registers)*				
x‘0290’ – x‘0378’	Reserved			
x‘0380’ x‘0388’ x‘0390’ x‘0398’ x‘03A0’ x‘03A8’ x‘03B0’	*MFC Fault Isolation Register (MFC_FIR)* *MFC Fault Isolation Register Set (MFC_FIR_Set)* *MFC Fault Isolation Register Reset (MFC_FIR_Reset)* *MFC Fault Isolation Error Mask (MFC_FIR_Err)* *MFC Fault Isolation Error Set Mask (MFC_FIR_Err_Set)* *MFC Fault Isolation Error Reset Mask (MFC_FIR_Err_Reset)* *MFC Checkstop Enable Register (MFC_FIR_ChkStpEnbl)*	64	R/W	*Section 3.2.4* on page 61 The starting point for SPE registers depends on the address of the SPE. There are up to eight SPEs on a chip.
Miscellaneous Registers *(Implementation-Specific Registers)*				
x ‘3B8’	*MFC SBI Data Error Address Register (MFC_SBI_Derr_Addr)*	64	R	*Section 3.2.5.1* on page 65
x ‘3C0’	*MFC Command Queue Error ID Register (MFC_CMDQ_Err_ID)*	64	R	*Section 3.2.5.2* on page 66
x‘03C8’ – x‘03F8’	Reserved			
MFC TLB Management Registers *(CBEA Architected Registers)*				
x‘0400’	*MFC Storage Description Register (MFC_SDR)*	64	R/W	*Section 3.2.8.1* on page 73
x‘0408’ – x‘04FF’	Reserved			
x‘0500’	*MFC TLB Index Hint Register (MFC_TLB_Index_Hint)*	64	R	*Section 3.2.8.2* on page 74
x‘0508’	*MFC TLB Index Register (MFC_TLB_Index)*	64	R/W	*Section 3.2.8.3* on page 75
x‘0510’	*MFC TLB Virtual Page Number Register (MFC_TLB_VPN)*	64	R/W	*Section 3.2.8.4* on page 76
x‘0518’	*MFC TLB Real Page Number Register (MFC_TLB_RPN)*	64	R/W	*Section 3.2.8.5* on page 77
x‘0520’ – x‘053F’	Reserved			
x‘0540’	*MFC TLB Invalidate Entry Register (MFC_TLB_Invalidate_Entry)*	64	W	*Section 3.2.8.6* on page 79
—	*MFC TLB Invalidate All Register (MFC_TLB_Invalidate_All)*	—	—	Not implemented
Memory Management Register *(Implementation-Specific Register)*				
x‘0580’	*SMM Hardware Implementation Dependent Register (SMM_HID)*	64	R/W	*Section 3.2.9.1* on page 81
MFC Status and Control Registers *(CBEA Architected Registers)*				
x‘0600’	*MFC Address Compare Control Register (MFC_ACCR)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0610’	*MFC Data-Storage Interrupt Status Register (MFC_DSISR)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0620’	*MFC Data Address Register (MFC_DAR)*	64	R/W	See *Appendix A Registers Defined in the CBEA*

Table 3-1. SPE Privilege 1 Memory Map (Page 3 of 3)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
x‘0628’ – x‘06FF’	Reserved			
Replacement Management Table (RMT) Registers *(Implementation-Specific Registers)*				
—	*MFC TLB Replacement Management Table Index Register (MFC_TLB_RMT_Index)*	—	—	Not implemented
x‘0710’	*MFC TLB Replacement Management Table Data Register (MFC_TLB_RMT_Data)*	64	R/W	*Section 3.2.10* on page 82
x‘0718’ – x‘07FF’	*SPU_RMT_ImplRegs*	—	—	Not implemented
MFC Command Data-Storage Interrupt Registers *(Implementation-Specific Registers)*				
x‘0800’	*MFC Data-Storage Interrupt Pointer Register (MFC_DSIPR)*	64	R	*Section 3.2.11.1* on page 83
x‘0808’	*MFC Local Store Address Compare Register (MFC_LSACR)*	64	R/W	*Section 3.2.11.2* on page 84
x‘0810’	*MFC Local Store Compare Results Register (MFC_LSCRR)*	64	R	*Section 3.2.11.3* on page 85
x‘0818’	Reserved			
x‘0820’	*MFC Transfer Class ID Register (MFC_TClassID)*	64	R/W	*Section 3.2.11.4* on page 86
x‘0828’ – x‘087F’	Reserved			
DMAC Unit Performance Monitor Control Register *(Implementation-Specific Register)*				
x‘0880’	*DMAC Unit Performance Monitor Control Register (DMAC_PMCR)*	64	R/W	*Section 3.2.12.1* on page 88
x‘0888’ – x‘08FF’	Reserved			
Real-Mode Support Registers *(CBEA Architected Register)*				
x‘0900’	*MFC Real Mode Address Boundary Register (MFC_RMAB)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘0908’ – x‘0BFF’	Reserved			
MFC Command Error Register				
x‘0C00’	*MFC Command Error Register (MFC_CER)*	64	R	*Section 3.2.13.1* on page 90
x‘0C08’ – x‘0FFF’	Reserved			
SPU ECC and Error Mask Registers *(Implementation-Specific Registers)*				
x‘1000’	*SPU ECC Control Register (SPU_ECC_Cntl)*	64	R/W	*Section 3.2.14.1* on page 91
x‘1008’	*SPU ECC Status Register (SPU_ECC_Stat)*	64	R/W	*Section 3.2.14.2* on page 92
x‘1010’	*SPU ECC Address Register (SPU_ECC_Addr)*	64	R	*Section 3.2.14.3* on page 94
x‘1018’	*SPU Error Mask Register (SPU_ERR_Mask)*	64	R/W	*Section 3.2.14.4* on page 95
x‘1028’ – x‘13FF’	Reserved			
Performance Monitor Register *(Implementation-Specific Register)*				
x‘1400’	*Performance Monitor/Trace Tag Status Wait Mask Register (PM_Trace_Tag_Wait_Mask)*	64	R/W	*Section 3.2.15.1* on page 96
x‘1408’ – x‘1FFF’	Reserved			

3.1.2 SPE Privilege 2 Memory Map

Multiple address offsets for a register indicate that there are multiple instances of this register.

Table 3-2. SPE Privilege 2 Memory Map (Page 1 of 2)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
MFC Registers				
x‘0000’ – x‘10FF’	Reserved			
SLB Management Registers *(CBEA Architected Registers)*				
x‘1100’	Reserved			
x‘1108’	*SLB Index Register (SLB_Index)*	64	R/W	*Section 3.3.1.1* on page 97
x‘1110’	*SLB Effective Segment ID Register (SLB_ESID)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘1118’	*SLB Virtual Segment ID Register (SLB_VSID)*	64	R/W	*Section 3.3.1.2* on page 98
x‘1120’	*SLB Invalidate Entry Register (SLB_Invalidate_Entry)*	64	W	*Section 3.3.1.3* on page 99
x‘1128’	*SLB Invalidate All Register (SLB_Invalidate_All)*	64	W	See *Appendix A Registers Defined in the CBEA*
x‘1130’ – x‘1FFF’	Reserved			
Context Save/Restore Register *(Implementation-Specific Register)*				
x‘2000’ – x‘22FF’	*MFC Command Queue Context Save/Restore Register (MFC_CQ_SR)*	64	R/W	*Section 3.3.2.1* on page 100
x‘2300’ – x‘2FF8’	Reserved			
MFC Control Register *(CBEA Architected Register)*				
x‘3000’	*MFC Control Register (MFC_CNTL)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘3008’ – x‘3FFF’	*MFC_Cntl1_ImplRegs*	—	—	Not implemented
Interrupt Mailbox Register *(Implementation-Specific Register)*				
x‘4000’	*SPU Outbound Interrupt Mailbox Register (SPU_OutIntrMbox)*	64	R	See *Appendix A Registers Defined in the CBEA*
SPU Control Registers *(CBEA Architected Registers)*				
x‘4040’	*SPU Privileged Control Register (SPU_PrivCntl)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘4058’	*SPU Local Store Limit Register (SPU_LSLR)*	64	R/W	*Section 3.3.2.2* on page 105
x‘4060’	*SPU Channel Index Register (SPU_ChnlIndex)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘4068’	*SPU Channel Count Register (SPU_ChnlCnt)*	64	R/W	*Section 3.3.2.3* on page 106
x‘4070’	*SPU Channel Data Register (SPU_ChnlData)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘4078’	*SPU Configuration Register (SPU_Cfg)*	64	R/W	See *Appendix A Registers Defined in the CBEA*
x‘4080’ – x‘4FFF	Reserved			
Context Save and Restore Registers *(Implementation-Specific Registers)*				
x‘5000’	Reserved			

Table 3-2. SPE Privilege 2 Memory Map (Page 2 of 2)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
x‘5008’	*Context Save and Restore for SPU MFC Commands Register (MFC_CSR_TSQ)*	64	R/W	*Section 3.3.3.1* on page 107
x‘5010’	*Context Save and Restore for SPU MFC Commands Register (MFC_CSR_CMD1)*	64	R/W	*Section 3.3.3.2* on page 108
x‘5018’	*Context Save and Restore for SPU MFC Commands Register (MFC_CSR_CMD2)*	64	R/W	*Section 3.3.3.3* on page 109
x‘5020’	*Context Save and Restore for SPU Atomic Immediate Command (MFC_CSR_ATO)*	64	R/W	*Section 3.3.3.4* on page 110
x‘5028’ – x‘1FFFF	Reserved			

3.1.3 SPE Problem State Memory Map

Multiple address offsets for a register indicate that there are multiple instances of this register.

Table 3-3. SPE Problem State Memory Map (Page 1 of 2)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
SPE Multisource Synchronization Register *(Implementation-Specific Register)*				
x‘0000’	*MFC Multisource Synchronization Register (MFC_MSSync)*	64	R/W	*Section 3.4.1.1* on page 111
x‘0008’ – x‘2FF8’	Reserved			
MFC Command Parameter Registers *(CBEA Architected Registers)*				
x‘3000’	Reserved			
x‘3004’	*MFC Local Store Address Register (MFC_LSA)*	32	W	*Section 3.4.2.1* on page 112
x‘3008’	*MFC Effective Address High Register (MFC_EAH)*	32	R/W	See *Appendix A Registers Defined in the CBEA*
x‘300C’	*MFC System Memory Address Register (MFC_EAL)*	32	R/W	See *Appendix A Registers Defined in the CBEA*
x‘3010’	*MFC Transfer Size Register (MFC_Size)*	16	R/W	See *Appendix A Registers Defined in the CBEA*
	MFC Command Tag Register (MFC_Tag)	16	R/W	See *Appendix A Registers Defined in the CBEA*
x‘3014’	*MFC Class ID and Command Opcode Register (MFC_ClassID_CMD)*	32	W	*Section 3.4.2.2* on page 113
x‘3014’	*MFC Command Status Register (MFC_CMDStatus)*	32	R	*Section 3.4.2.3* on page 114
Reserved Area				
x‘3020’ – x‘30FF’	Reserved			

Table 3-3. SPE Problem State Memory Map (Page 2 of 2)

Hexadecimal Offset	Register Name and (Short Name)	Width	Read/ Write	Additional Information
MFC Command Queue Control Registers *(CBEA Architected Registers)*				
x‘3104’	*MFC Queue Status Register (MFC_QStatus)*	32	R	*Section 3.4.3.1* on page 115
x‘3204’	*Proxy Tag-Group Query Type Register (Prxy_QueryType)*	32	R/W	See *Appendix A Registers Defined in the CBEA*
x‘321C’	*Proxy Tag-Group Query Mask Register (Prxy_QueryMask)*	32	R/W	See *Appendix A Registers Defined in the CBEA*
x‘322C’	*Proxy Tag-Group Status Register (Prxy_TagStatus)*	32	R	See *Appendix A Registers Defined in the CBEA*
Reserved Area				
x‘3330’ – x‘3FFF’	Reserved			
SPU Control Registers *(CBEA Architected Registers)*				
x‘4004’	*SPU Outbound Mailbox Register (SPU_Out_Mbox)*	32	R	See *Appendix A Registers Defined in the CBEA*
x‘400C’	*SPU Inbound Mailbox Register (SPU_In_Mbox)*	32	W	See *Appendix A Registers Defined in the CBEA*
x‘4014’	*SPU Mailbox Status Register (SPU_Mbox_Stat)*	32	R	*Section 3.4.3.2* on page 116
x‘401C’	*SPU Run Control Register (SPU_RunCntl)*	32	R/W	*Section 3.4.3.3* on page 117
x‘4024’	*SPU Status Register (SPU_Status)*	32	R	*Section 3.4.3.4* on page 118
x‘4034’	*SPU Next Program Counter Register (SPU_NPC)*	32	R/W	*Section 3.4.3.5* on page 120
Reserved Area				
x‘4038’ – x‘13FFF’	Reserved			
Signal Notification Registers *(CBEA Architected registers)*				
x‘1400C’	*SPU Signal Notification Register 1 (SPU_Sig_Notify_1)*	32	R/W	See *Appendix A Registers Defined in the CBEA*
x‘14010’ – x‘1BFFF’	Reserved			
x‘1C00C’	*SPU Signal Notification Register 2 (SPU_Sig_Notify_2)*	32	R/W	See *Appendix A Registers Defined in the CBEA*
x‘1C010’ – x‘1FFFF’	Reserved			

3.2 SPE Privilege 1 Memory Map Registers

This section lists the registers included in the SPE Privilege 1 memory map, with the exception of those that are defined in the CBEA. See *Appendix A Registers Defined in the CBEA* for information about those registers.

3.2.1 MFC Register

This area includes the Logical Partition ID Register and the SPU ID Register.

3.2.1.1 MFC Logical Partition ID Register (MFC_LPID)

The MFC_LPID Register contains a value that identifies the partition to which an SPE is assigned. Only five bits of the LPID are implemented.

Register Short Name	MFC_LPID	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400008’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC

Bits	Field Name	Description
0:55	Reserved	Bits are not implemented; all bits read back zero.
56:58	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
59:63	LPID	Logical partition ID.

3.2.1.2 SPU Identification Register (SPU_ID)

The SPU_ID Register is a read-only register that distinguishes a particular SPU from other SPUs in the system. This register is accessible from the PPE using a load doubleword (**ld**) instruction. Read access to the SPU_ID Register from the PPE is privileged, and access to this register from the SPU is not provided. There is one SPU_ID Register for each SPU in the Cell Broadband Engine.

For additional information, see the *Cell Broadband Engine Architecture* document.

Register Short Name	SPU_ID	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400010’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_000000NU’ (N = Node_ID, U = Unit_ID)	**Value During POR Set By**	Bits 56-59 set by configuration ring Bits 60-63 hardwired per SPE on chip
Specification Type	CBEA architected register	**Unit**	MFC

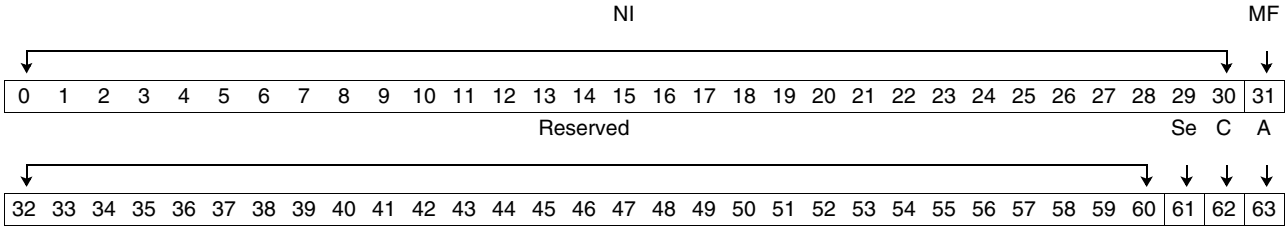
Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32:55	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
56:59	Node_ID	A 4-bit number that identifies the Node ID. This is set in the configuration ring, SPU/SBI[204:207] per SPE on chip.
60:63	Unit_ID	A 4-bit hardwired number that identifies the Unit ID in the bus.

3.2.2 Interrupt Registers

There are interrupt mask and interrupt status registers in each MFC: one for each class of interrupt (error, translation, application). The interrupt registers allow privileged software to select which MFC and SPU events are allowed to generate an external interrupt to the PPE. Each bit of the mask registers has a corresponding status bit. The mask register determines which bits are reported in the status register. See the *Cell Broadband Engine Architecture* document for more information about these registers.

3.2.2.1 Class 0 Interrupt Mask Register (INT_Mask_class0)

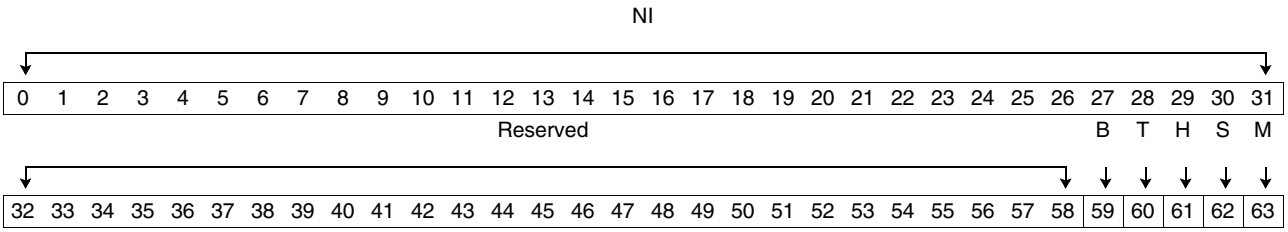
Register Short Name	INT_Mask_class0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400100’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:30	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
31	MF	Enable for MFC_FIR interrupt 0 Interrupt disabled 1 Interrupt enabled
32:60	Reserved	Bits are not implemented; all bits read back zero.
61	Se	Enable for SPU error interrupt 0 Interrupt disabled 1 Interrupt enabled
62	C	Enable for invalid direct memory access (DMA) command interrupt 0 Interrupt disabled 1 Interrupt enabled
63	A	Enable for MFC DMA alignment interrupt 0 Interrupt disabled 1 Interrupt enabled

3.2.2.2 Class 2 Interrupt Mask Register (INT_Mask_class2)

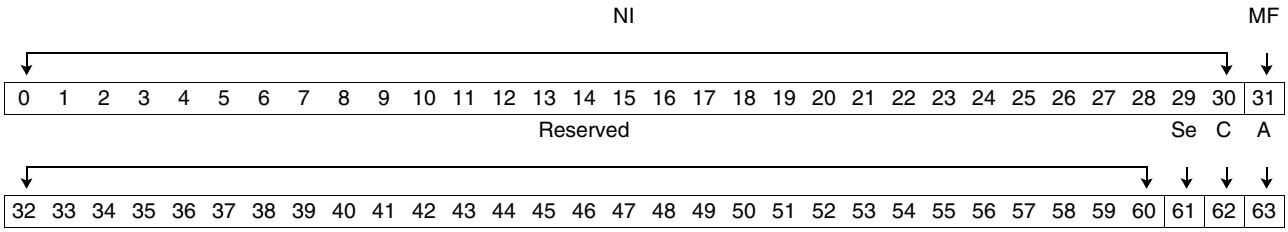
Register Short Name	INT_Mask_class2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400110’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:31	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
32:58	Reserved	Bits are not implemented; all bits read back zero.
59	B	Enable for SPU Mailbox threshold interrupt 0 Interrupt disabled 1 Interrupt enabled
60	T	Enable for Tag group completion 0 Interrupt disabled 1 Interrupt enabled
61	H	Enable for SPU Halt instruction trap 0 Interrupt disabled 1 Interrupt enabled
62	S	Enable for SPU Stop-and-signal instruction trap 0 Interrupt disabled 1 Interrupt enabled
63	M	Enable for Mailbox interrupt 0 Interrupt disabled 1 Interrupt enabled

3.2.2.3 Class 0 Interrupt Status Register (INT_Stat_class0)

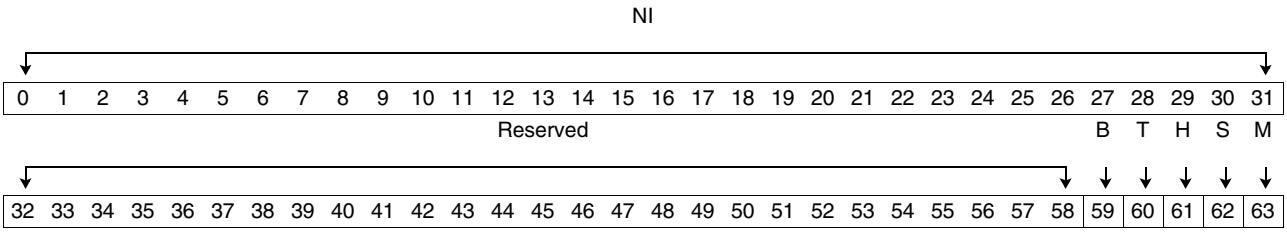
Register Short Name	INT_Stat_class0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400140’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:30	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
31	MF	Status for MFC_FIR interrupt. The invalid DMA command interrupt bit [62], DMA alignment interrupt bit [63], and five hang livelock indication conditions defined in the MFC_FIR do not generate an INT_Stat_class0[31] class 0 interrupt. 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type
32:60	Reserved	All bits read back zero.
61	Se	Status for SPU error interrupt. If there are invalid SPU instructions in SPU_Status[26] and if these interrupts are enabled in SPU_ERR_Mask[63], then this bit (INT_Stat_class0[61] goes to ‘1’. Also for uncorrectable ECC errors. Status is in SPU_ECC_Stat[62]; control is in SPU_ECC_Cntl[62]. 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type **Note:** This implementation differs from the *Cell Broadband Engine Architecture*.
62	C	Status for invalid DMA command interrupt 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type
63	A	Status for DMA alignment interrupt 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type

3.2.2.4 Class 2 Interrupt Status Register (INT_Stat_class2)

Register Short Name	INT_Stat_class2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400150’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC

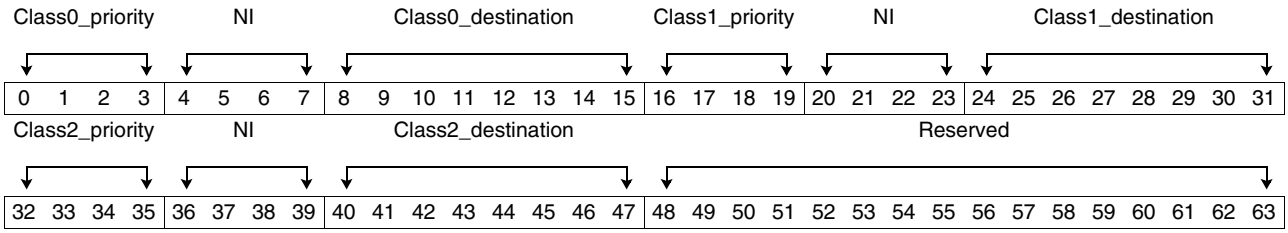


Bits	Field Name	Description
0:31	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
32:58	Reserved	All bits read back zero.
59	B	Status for SPU Mailbox threshold interrupt 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type
60	T	Status for DMA Tag group complete interrupt 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type
61	H	Status for SPU Halt instruction trap 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type **Note:** This implementation differs from the *Cell Broadband Engine Architecture*.
62	S	Status for SPU Stop-and-signal instruction trap 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type.
63	M	Status for Mailbox interrupt 0 Interrupt not pending for corresponding interrupt type 1 Interrupt pending for corresponding interrupt type

3.2.2.5 Interrupt Routing Register (INT_Route)

For each class of interrupt, only the most significant 4 bits of the priority field are implemented. See the *Cell Broadband Engine Architecture* document for more information about this register.

Register Short Name	INT_Route	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400180’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:3	Class0_priority	Priority for a class 0 interrupt. This priority corresponds to the upper 4 bits of class priority in the *Cell Broadband Engine Architecture* document.
4:7	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
8:15	Class0_destination	Destination ID for a class 0 interrupt (upper 4 bits = Node ID; lower 4 bits = Unit ID).
16:19	Class1_priority	Priority for a class 1 interrupt.
20:23	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
24:31	Class1_destination	Destination ID for a class 1 interrupt (upper 4 bits = Node ID, lower 4 bits = Unit ID).
32:35	Class2_priority	Priority for a class 2 interrupt.
36:39	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
40:47	Class2_destination	Destination ID for a class 2 interrupt (upper 4 bits = Node ID, lower 4 bits = Unit ID).
48:63	Reserved	Bits are not implemented; all bits read back zero.

3.2.3 Atomic Unit Control Registers

These registers control the atomic unit. See the *Cell Broadband Engine Architecture* document for information about these registers.

3.2.3.1 Resource Allocation Group ID (RA_Group_ID)

The RA_Group_ID is a 2-bit register.

Register Short Name	RA_Group_ID	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400280’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC

Bits	Field Name	Description
0:61	Reserved	Bits are not implemented; all bits read back zero.
62:63	RAID	Two-bit Resource Allocation ID.

3.2.3.2 Resource Allocation Enable Register (RA_Enable)

The RA_Enable Register allows you to override resource allocation. Resource allocation is enabled only when both bits [62] and [63] are '1'.

Register Short Name	RA_Enable	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x'400288' + (x'02000' x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero except Bit 62 set by configuration ring	**Value During POR Set By**	Scan initialization during POR Bit 62 set by configuration ring
Specification Type	Implementation-specific register	**Unit**	MFC

Bits	Field Name	Description
0:61	Reserved	Bits are not implemented; all bits read back zero.
62	C	Resource Allocation Override bit. Read only. Resource Allocation Override is set in the configuration ring, and this field enables a read of that configuration-ring setting. 0 No read of Resource Allocation Override setting in the configuration ring 1 Read of Resource Allocation Override setting in the configuration ring
63	M	Resource Allocation Enable bit. Resource allocation is enabled only when both the C bit and the M bit are set to '1'. 0 Resource allocation disabled. 1 Resource allocation is enabled if the C bit is also set to '1'.

3.2.4 MFC Fault Isolation, Error Mask, Checkstop Enable Registers

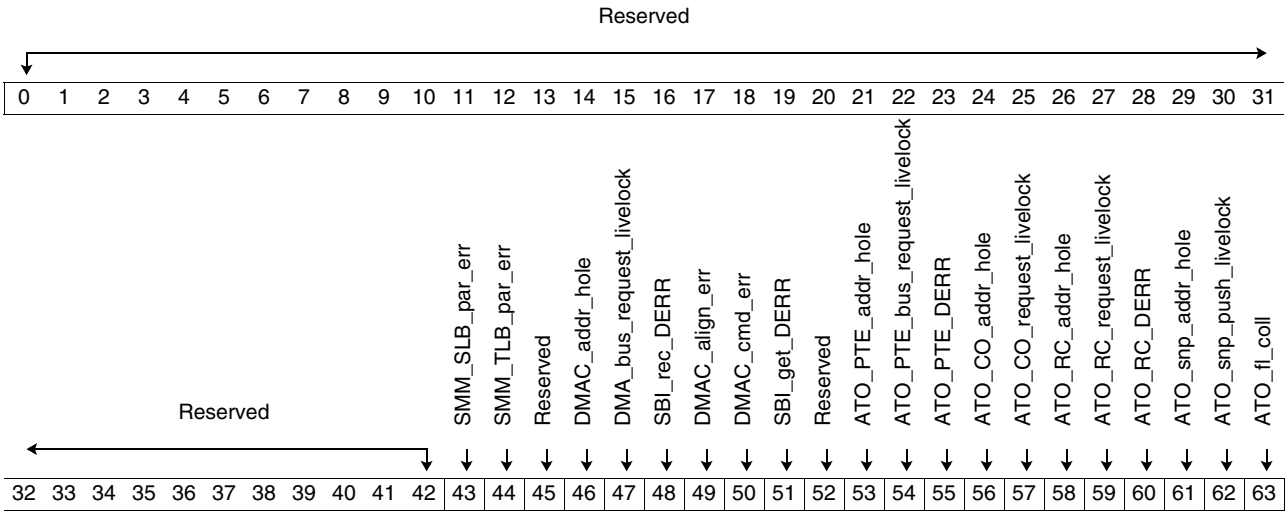
The MFC Fault Isolation Registers (MFC_FIR) are defined in this section. These registers are part of the SPE.

Note: A '1' written to any bit in the MFC_FIR_Reset Register clears the corresponding bit in the MFC_FIR Register. FIR registers in other units write a '0' to the FIR_Reset Register to clear the corresponding FIR bit.

Register Short Name	Register Name
MFC_FIR	MFC Fault Isolation Register
MFC_FIR_Set	MFC Fault Isolation Register Set
MFC_FIR_Reset	MFC Fault Isolation Register Reset
MFC_FIR_Err	MFC Fault Isolation Register Error Mask
MFC_FIR_Err_Set	MFC Fault Isolation Register Error Mask Set
MFC_FIR_Err_Reset	MFC Fault Isolation Register Error Mask Reset
MFC_FIR_ChkStpEnbl	MFC Fault Isolation Register Checkstop Enable

Register Short Name	See table above	**Privilege Type**	Privilege 1
Access Type	See table below	**Width**	64 bits
Hex Offset From BE_MMIO_Base	See table below	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SBI

Register Short Name	Hex Offset From BE_MMIO_Base	Access Type	FIR Function
MFC_FIR	SPE*n*: x'400380' + (x'02000' x *n*)	MMIO Read Only	FIR Read
MFC_FIR_Set	SPE*n*: x'400388' + (x'02000' x *n*)	MMIO Write Only	FIR Set
MFC_FIR_Reset	SPE*n*: x'400390' + (x'02000' x *n*)	MMIO Write Only	FIR Reset
MFC_FIR_Err	SPE*n*: x'400398' + (x'02000' x *n*)	MMIO Read Only	Error Mask Read
MFC_FIR_Err_Set	SPE*n*: x'4003A0' + (x'02000' x *n*)	MMIO Write Only	Error Mask Set
MFC_FIR_Err_Reset	SPE*n*: x'4003A8' + (x'02000' x *n*)	MMIO Write Only	Error Mask Reset
MFC_FIR_ChkStpEnbl	SPE*n*: x'4003B0' + (x'02000' x *n*)	MMIO Read/Write	Checkstop Enable



Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
0:42	Reserved	Bits are not implemented; all bits read back zero.	NA	NA
43	SMM_SLB_par_err	SLB parity error is detected. When an SLB parity error is detected, a class 0 interrupt (INT_Stat_class0) is generated. The DMA command that causes the parity error is suspended in the MFC command queues until either a software restart is issued to clear the command issue suspension state or a DMA purge is issued to reinitialize the DMA machine to the default condition.	0	0
44	SMM_TLB_par_err	Translation lookaside buffer (TLB) parity error is detected. When a TLB parity error is detected, a class 0 interrupt (INT_Stat_class0) is generated. The DMA command that causes the parity error is suspended in the MFC command queues until either a software restart is issued to clear the command issue suspension state or a DMA purge is issued to reinitialize the DMA machine to the default condition.	0	0
45	Reserved	Bits are not implemented; all bits read back zero.	0	0
46	DMAC_addr_hole	A null combined response was received, therefore the DMA bus request hit an address hole. The synergistic bus interface (SBI) records the MFC command queues entry index (the first 5 bits of the DMAC tag are recorded in the MFC_CMDQ_Err_ID Register). The SBI does not send a completion reply back to the DMAC. To clear this error, the software sends a DMA purge to reinitialize the DMA state machine. The SBI resets the state machine of the DMA bus request that caused the error. The SBI stops issuing any EIB commands until this FIR bit is reset.	0	0
47	DMA_bus_request_livelock	This condition may be caused by excessive retries to the same DMA bus request. This condition does not generate a class 0 interrupt. Instead, it causes the system to enter quiescent mode to resolve the livelock condition. Five bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
48	SBI_rec_DERR	When active, the SBI_rec_DERR bit indicates that the SBI received a DERR during a snoop write data phase. The SBI records the write address in the MFC_SBI_Derr_Addr Register. Even though this bit is set, the data is still stored to the destination in the SPE.	0	0

Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
49	DMAC_align_err	DMA alignment error conditions: • Transfer size: > 16 KB; size neither partial nor quadword; **sndsig** size not 4 bytes; partial check for effective address failed. • List transfer size: > 2 K list elements; bits [13:15] are nonzero. • Local store address: local store address (LSA) bits [28:31] are not equal to the effective address; bits [60:63] for **put**, **get**, or **sndsig** are not all zeros for quadword transfers. • List address: Lower 3 bits are nonzero. The class 0 interrupt is reported, and the MFC command queue index is logged in the MFC_CER Register. A '1' value is written to the error-bit field of the error entry in the MFC command queues. The DMA suspend sequence is started by the DMA control state machine. DMAC_align_err is blocked for DMA commands that produce DMAC_err_cond.	1	0
50	DMAC_cmd_err	DMA command error conditions: • An atomic command is received while one is still pending in the queue. • MFC proxy command queue side: list or atomic commands present. • MFC SPU command queue side: start modifier present. • Upper 8 bits of opcode are not all zero. • Invalid opcode. • Transfer tag bits [0:10] are nonzero. • An interrupt is reported. The class 0 interrupt is reported and the MFC command queue index is logged in the MFC_CER Register. A '1' value is written to the error-bit field of the error entry in the MFC command queue. The DMA-suspend sequence is started by the DMA control state machine.	1	0
51	SBI_get_DERR	The SBI records the MFC command queue entry index in the MFC_CMDQ_Err_ID Register. The SBI does not send the completion reply back to the DMAC. (To clear this error, the software sends a DMA purge to reinitialize the DMA state machine.) The SBI resets the state machine of the DMA get bus request that caused the error. The SBI stops issuing any EIB commands until this FIR bit is reset.	0	0
52	Reserved	Bit is not implemented; bit reads back zero.	0	0
53	ATO_PTE_addr_hole	The atomic (ATO) Page Table Entry Group (PTEG) bus request hit an address hole. The ATO read and claim (RC) machine freezes until software resets this FIR bit. To clear this error, the software sends a DMA purge to reinitialize the DMA state machine. The FIR bit reset reinitializes the synergistic memory management (SMM) page table entry (PTE) tablewalk state machine.	0	1
54	ATO_PTE_bus_request_livelock	The RC machine hangs while servicing a PTE request. This condition does not cause a checkstop; instead, it causes the system to enter quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
55	ATO_PTE_DERR	The ATO PTEG bus request detected a DERR. The ATO RC machine freezes until the software resets this FIR bit. Software should only set this FIR bit to be checkstop enabled.	0	1
56	ATO_CO_addr_hole	The ATO castout machine (CO) hit an address hole. After the CO machine detects an address hole, it may pulse this signal to SBI again later when ATO global combined response (GRESP) is activated. The ATO CO machine freezes until software resets this FIR bit. Software should only set this FIR bit to be checkstop enabled.	0	1

Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
57	ATO_CO_request_livelock	The ATO castout machine detects a hang at either the command phase or the data phase. The CO retry count latch can be used to distinguish between a bus hang or a retry hang. This condition does not cause a checkstop; instead, it causes the system to enter the quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
58	ATO_RC_addr_hole	An ATO RC machine bus request hit an address hole. The ATO RC machine freezes until software resets this FIR bit. The ATO sends the hang signal to suspend the DMA operation. Software should only set this FIR bit to be checkstop enabled.	0	1
59	ATO_RC_request_livelock	The ATO RC machine detects a hang either at the command phase or at the data phase. The RC retry count latch can be used to distinguish between a bus hang or a retry hang. This condition does not cause checkstop; instead, it causes the system to enter the quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
60	ATO_RC_DERR	An ATO RC machine bus request received a DERR. (The data can be sourced from the intervened cache, local store (LS), or the system memory.) The cache state is updated at the time of this error. The ATO RC machine freezes until the software resets this FIR bit. The ATO sends a hang signal to suspend DMA operation. The state of the frozen RC machine can be scanned to tell the DERR source: cache intervention, LS, or the system memory. Software should only set this FIR bit to be checkstop enabled.	0	1
61	ATO_snp_addr_hole	An ATO snoop hit an address hole. The ATO snoop machine freezes until the software resets this FIR bit. Software should only set this FIR bit to be checkstop enabled.	0	1
62	ATO_snp_push_livelock	An ATO snoop push hang was detected. The ATO snoop machine detected a hang on the bus because either the combined response was not received or the data phase cannot be completed. This condition does not cause a checkstop; instead, it causes the system to enter the quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
63	ATO_fl_coll	ATO flush collision. The ATO detects a DMA or SMM request while the ATO flush is active. Software is guaranteed to prevent this error. Software should only set this FIR bit to be checkstop-enabled.	0	1

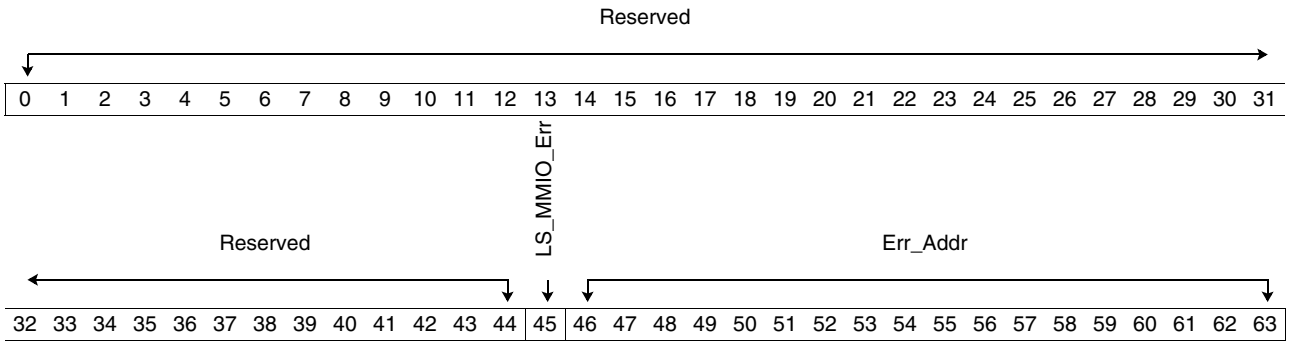
3.2.5 Miscellaneous Registers

The following registers are used with the FIR registers.

3.2.5.1 MFC SBI Data Error Address Register (MFC_SBI_Derr_Addr)

This register captures the destination address (as a slave device) when the SPE receives data that has an error (DERR = '1' on the EIB) and the error is recorded in MFC_FIR[48].

Register Short Name	MFC_SBI_Derr_Addr	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x'4003B8' + (x'02000' x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SPU

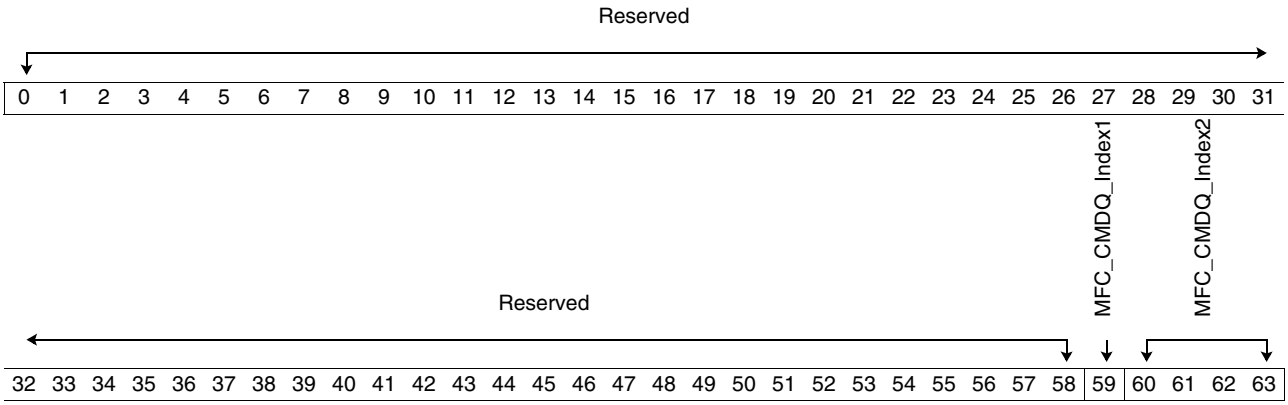


Bits	Field Name	Description
0:44	Reserved	Bits are not implemented; all bits read back zero.
45	LS_MMIO_Err	Local store or MMIO error 0 An LS write access caused the error. 1 An MMIO write access caused the error.
46:63	Err_Addr	Error Address If bit [45] equals '0', bits [46:63] capture the LS write address. If bit [45] equals '1' and bit [46] equals '1', the address is from a Privilege 1 MMIO write operation. If bit [45] equals '1' and bit [47] equals '1', the address is from a Privilege 2 MMIO write operation. If bit [45] equals '1' and bit [48] equals '1', the address is from a Problem State MMIO write operation. If bit [45] equals '1', then bits [49:63] capture the real address [25:39] of the MMIO write operation.

3.2.5.2 MFC Command Queue Error ID Register (MFC_CMDQ_Err_ID)

This register saves the MFC command queue entry index number that caused a DMA bus transaction request error. If multiple errors occur (as recorded in MFC_FIR bits[46] [47] [51]), only the first error is recorded. After the FIR bit corresponding to the error is reset, a new value can be written to this register.

Register Short Name	MFC_CMDQ_Err_ID	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘4003C0’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SPU



Bits	Field Name	Description
0:58	Reserved	Bits are not implemented; all bits read back zero.
59	MFC_CMDQ_Index1	MFC command queue index 0 MFC proxy command queue index 1 MFC SPU command queue index
60:63	MFC_CMDQ_Index2	Points to the command queue entry that caused an SBI bus transaction request error. When the MFC_CMDQ_Index1[59] field is set to ‘0’, this MFC_CMDQ_Index2[60:63] field is also set to ‘0’.

3.2.6 MFC Fault Isolation, Error Mask, Checkstop Enable Registers

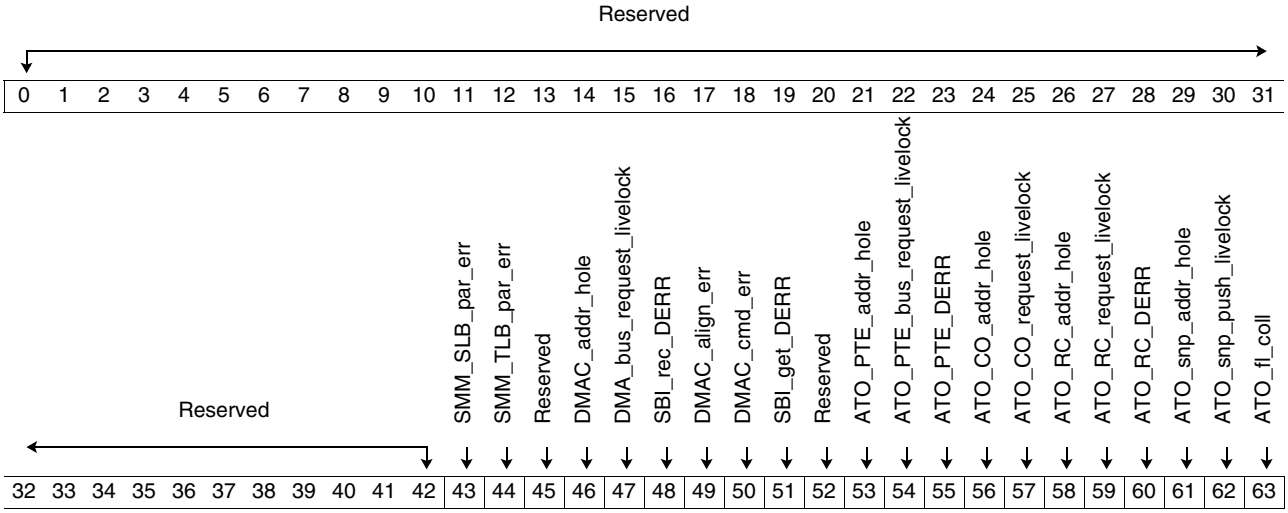
The MFC Fault Isolation Registers (MFC_FIR) are defined in this section. This register is part of the SPE.

Note: A '1' written to any bit in the MFC_FIR_Reset Register clears the corresponding bit in the MFC_FIR Register. FIR registers in other units write a '0' to the FIR_Reset Register to clear the corresponding FIR bit.

Register Short Name	Register Name
MFC_FIR	MFC Fault Isolation Register
MFC_FIR_Set	MFC Fault Isolation Register Set
MFC_FIR_Reset	MFC Fault Isolation Register Reset
MFC_FIR_Err	MFC Fault Isolation Register Error Mask
MFC_FIR_Err_Set	MFC Fault Isolation Register Error Mask Set
MFC_FIR_Err_Reset	MFC Fault Isolation Register Error Mask Reset
MFC_FIR_ChkStpEnbl	MFC Fault Isolation Register Checkstop Enable

Register Short Name	See table above	**Privilege Type**	Privilege 1
Access Type	See table below	**Width**	64 bits
Hex Offset From BE_MMIO_Base	See table below	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SBI

Register Short Name	Hex Offset From BE_MMIO_Base	Access Type	FIR Function
MFC_FIR	SPE*n*: x'400380' + (x'02000' x *n*)	MMIO Read Only	FIR Read
MFC_FIR_Set	SPE*n*: x'400388' + (x'02000' x *n*)	MMIO Write Only	FIR Set
MFC_FIR_Reset	SPE*n*: x'400390' + (x'02000' x *n*)	MMIO Write Only	FIR Reset
MFC_FIR_Err	SPE*n*: x'400398' + (x'02000' x *n*)	MMIO Read Only	Error Mask Read
MFC_FIR_Err_Set	SPE*n*: x'4003A0' + (x'02000' x *n*)	MMIO Write Only	Error Mask Set
MFC_FIR_Err_Reset	SPE*n*: x'4003A8' + (x'02000' x *n*)	MMIO Write Only	Error Mask Reset
MFC_FIR_ChkStpEnbl	SPE*n*: x'4003B0' + (x'02000' x *n*)	MMIO Read/Write	Checkstop Enable



Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
0:42	Reserved	Bits are not implemented; all bits read back zero.	NA	NA
43	SMM_SLB_par_err	SLB parity error is detected. When an SLB parity error is detected, a class 0 interrupt (INT_Stat_class0) is generated. The DMA command that causes the parity error is suspended in the MFC command queues until either a software restart is issued to clear the command issue suspension state or a DMA purge is issued to reinitialize the DMA machine to the default condition.	0	0
44	SMM_TLB_par_err	Translation lookaside buffer (TLB) parity error is detected. When a TLB parity error is detected, a class 0 interrupt (INT_Stat_class0) is generated. The DMA command that causes the parity error is suspended in the MFC command queues until either a software restart is issued to clear the command issue suspension state or a DMA purge is issued to reinitialize the DMA machine to the default condition.	0	0
45	Reserved	Bits are not implemented; all bits read back zero.	0	0
46	DMAC_addr_hole	A null combined response was received, therefore the DMA bus request hit an address hole. The synergistic bus interface (SBI) records the MFC command queues entry index (the first 5 bits of the DMAC tag are recorded in the MFC_CMDQ_Err_ID Register). The SBI does not send a completion reply back to the DMAC. To clear this error, the software sends a DMA purge to reinitialize the DMA state machine. The SBI resets the state machine of the DMA bus request that caused the error. The SBI stops issuing any EIB commands until this FIR bit is reset.	0	0
47	DMA_bus_request_livelock	This condition may be caused by excessive retries to the same DMA bus request. This condition does not generate a class 0 interrupt. Instead, it causes the system to enter quiescent mode to resolve the livelock condition. Five bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
48	SBI_rec_DERR	When active, the SBI_rec_DERR bit indicates that the SBI received a DERR during a snoop write data phase. The SBI records the write address in the MFC_SBI_Derr_Addr Register. Even though this bit is set, the data is still stored to the destination in the SPE.	0	0

Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
49	DMAC_align_err	DMA alignment error conditions: • Transfer size: > 16 KB; size neither partial nor quadword; **sndsig** size not 4 bytes; partial check for effective address failed. • List transfer size: > 2 K list elements; bits [13:15] are nonzero. • Local store address: local store address (LSA) bits [28:31] are not equal to the effective address; bits [60:63] for **put**, **get**, or **sndsig** are not all zeros for quadword transfers. • List address: Lower 3 bits are nonzero. The class 0 interrupt is reported, and the MFC command queue index is logged in the MFC_CER Register. A '1' value is written to the error-bit field of the error entry in the MFC command queues. The DMA suspend sequence is started by the DMA control state machine. DMAC_align_err is blocked for DMA commands that produce DMAC_err_cond.	1	0
50	DMAC_cmd_err	DMA command error conditions: • An atomic command is received while one is still pending in the queue. • MFC proxy command queue side: list or atomic commands present. • MFC SPU command queue side: start modifier present. • Upper 8 bits of opcode are not all zero. • Invalid opcode. • Transfer tag bits [0:10] are nonzero. • An interrupt is reported. The class 0 interrupt is reported and the MFC command queue index is logged in the MFC_CER Register. A '1' value is written to the error-bit field of the error entry in the MFC command queue. The DMA-suspend sequence is started by the DMA control state machine.	1	0
51	SBI_get_DERR	The SBI records the MFC command queue entry index in the MFC_CMDQ_Err_ID Register. The SBI does not send the completion reply back to the DMAC. (To clear this error, the software sends a DMA purge to reinitialize the DMA state machine.) The SBI resets the state machine of the DMA get bus request that caused the error. The SBI stops issuing any EIB commands until this FIR bit is reset.	0	0
52	Reserved	Bit is not implemented; bit reads back zero.	0	0
53	ATO_PTE_addr_hole	The atomic (ATO) Page Table Entry Group (PTEG) bus request hit an address hole. The ATO read and claim (RC) machine freezes until software resets this FIR bit. To clear this error, the software sends a DMA purge to reinitialize the DMA state machine. The FIR bit reset reinitializes the synergistic memory management (SMM) page table entry (PTE) tablewalk state machine.	0	1
54	ATO_PTE_bus_request_livelock	The RC machine hangs while servicing a PTE request. This condition does not cause a checkstop; instead, it causes the system to enter quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
55	ATO_PTE_DERR	The ATO PTEG bus request detected a DERR. The ATO RC machine freezes until the software resets this FIR bit. Software should only set this FIR bit to be checkstop enabled.	0	1
56	ATO_CO_addr_hole	The ATO castout machine (CO) hit an address hole. After the CO machine detects an address hole, it may pulse this signal to SBI again later when ATO global combined response (GRESP) is activated. The ATO CO machine freezes until software resets this FIR bit. Software should only set this FIR bit to be checkstop enabled.	0	1

Bits	Field Name	Description	Recommended Settings	
			Error Mask	Checkstop Enable
57	ATO_CO_request_livelock	The ATO castout machine detects a hang at either the command phase or the data phase. The CO retry count latch can be used to distinguish between a bus hang or a retry hang. This condition does not cause a checkstop; instead, it causes the system to enter the quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
58	ATO_RC_addr_hole	An ATO RC machine bus request hit an address hole. The ATO RC machine freezes until software resets this FIR bit. The ATO sends the hang signal to suspend the DMA operation. Software should only set this FIR bit to be checkstop enabled.	0	1
59	ATO_RC_request_livelock	The ATO RC machine detects a hang either at the command phase or at the data phase. The RC retry count latch can be used to distinguish between a bus hang or a retry hang. This condition does not cause checkstop; instead, it causes the system to enter the quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
60	ATO_RC_DERR	An ATO RC machine bus request received a DERR. (The data can be sourced from the intervened cache, local store (LS), or the system memory.) The cache state is updated at the time of this error. The ATO RC machine freezes until the software resets this FIR bit. The ATO sends a hang signal to suspend DMA operation. The state of the frozen RC machine can be scanned to tell the DERR source: cache intervention, LS, or the system memory. Software should only set this FIR bit to be checkstop enabled.	0	1
61	ATO_snp_addr_hole	An ATO snoop hit an address hole. The ATO snoop machine freezes until the software resets this FIR bit. Software should only set this FIR bit to be checkstop enabled.	0	1
62	ATO_snp_push_livelock	An ATO snoop push hang was detected. The ATO snoop machine detected a hang on the bus because either the combined response was not received or the data phase cannot be completed. This condition does not cause a checkstop; instead, it causes the system to enter the quiescent mode to resolve the livelock condition. The 5 bits of the MFC_FIR_Err (bits 47, 54, 57, 59, and 62) are hardwired to return '1' for a read operation. There is no effect on these 5 bits for write operations. Because they are hardwired to return '1', the corresponding checkstop-enable bit cannot be used to enable checkstop.	0	0
63	ATO_fl_coll	ATO flush collision. The ATO detects a DMA or SMM request while the ATO flush is active. Software is guaranteed to prevent this error. Software should only set this FIR bit to be checkstop-enabled.	0	1

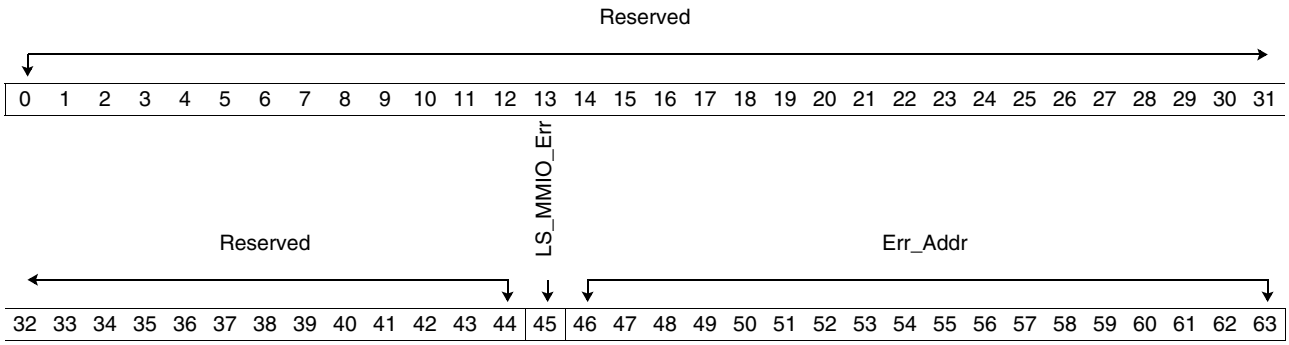
3.2.7 Miscellaneous Registers

The following registers are used with the FIR registers.

3.2.7.1 MFC SBI Data Error Address Register (MFC_SBI_Derr_Addr)

This register captures the destination address (as a slave device) when the SPE receives data that has an error (DERR = '1' on the EIB) and the error is recorded in MFC_FIR[48].

Register Short Name	MFC_SBI_Derr_Addr	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x'4003B8' + (x'02000' x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SPU

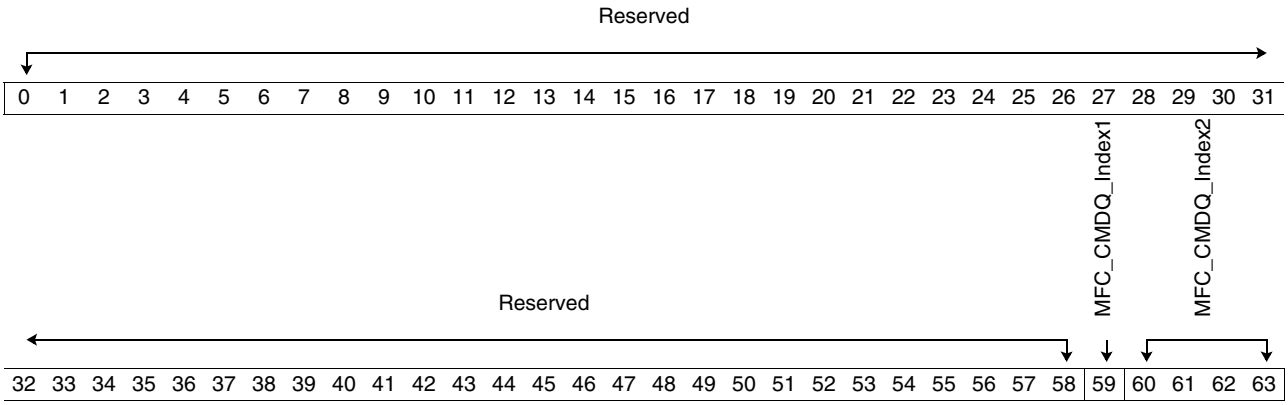


Bits	Field Name	Description
0:44	Reserved	Bits are not implemented; all bits read back zero.
45	LS_MMIO_Err	Local store or MMIO error 0 An LS write access caused the error. 1 An MMIO write access caused the error.
46:63	Err_Addr	Error Address If bit [45] equals '0', bits [46:63] capture the LS write address. If bit [45] equals '1' and bit [46] equals '1', the address is from a Privilege 1 MMIO write operation. If bit [45] equals '1' and bit [47] equals '1', the address is from a Privilege 2 MMIO write operation. If bit [45] equals '1' and bit [48] equals '1', the address is from a Problem State MMIO write operation. If bit [45] equals '1', then bits [49:63] capture the real address [25:39] of the MMIO write operation.

3.2.7.2 MFC Command Queue Error ID Register (MFC_CMDQ_Err_ID)

This register saves the MFC command queue entry index number that caused a DMA bus transaction request error. If multiple errors occur (as recorded in MFC_FIR bits[46] [47] [51]), only the first error is recorded. After the FIR bit corresponding to the error is reset, a new value can be written to this register.

Register Short Name	MFC_CMDQ_Err_ID	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘4003C0’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SPU



Bits	Field Name	Description
0:58	Reserved	Bits are not implemented; all bits read back zero.
59	MFC_CMDQ_Index1	MFC command queue index 0 MFC proxy command queue index 1 MFC SPU command queue index
60:63	MFC_CMDQ_Index2	Points to the command queue entry that caused an SBI bus transaction request error. When the MFC_CMDQ_Index1[59] field is set to ‘0’, this MFC_CMDQ_Index2[60:63] field is also set to ‘0’.

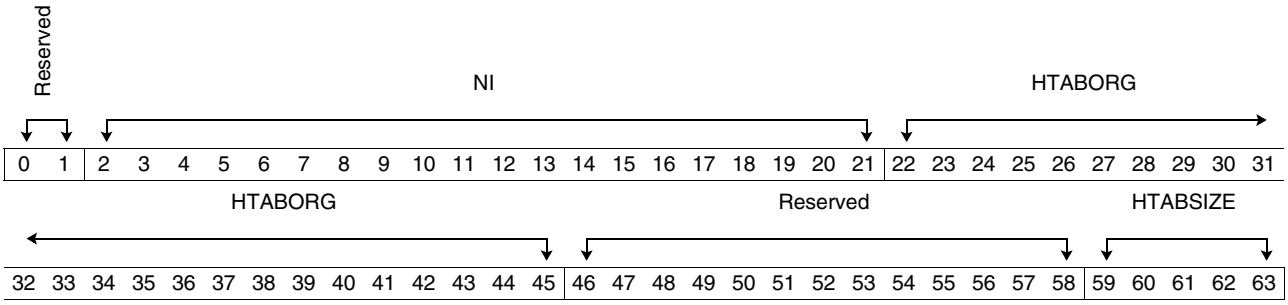
3.2.8 MFC TLB Management Registers

These registers support software TLB management.

3.2.8.1 MFC Storage Description Register (MFC_SDR)

The MFC_SDR Register contains the hash table origin and size. The functionality is identical to the PPE SDR1 Register (see the *PowerPC Architecture, Book III* for more information). The implemented bits are shown below.

Register Short Name	MFC_SDR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400400’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



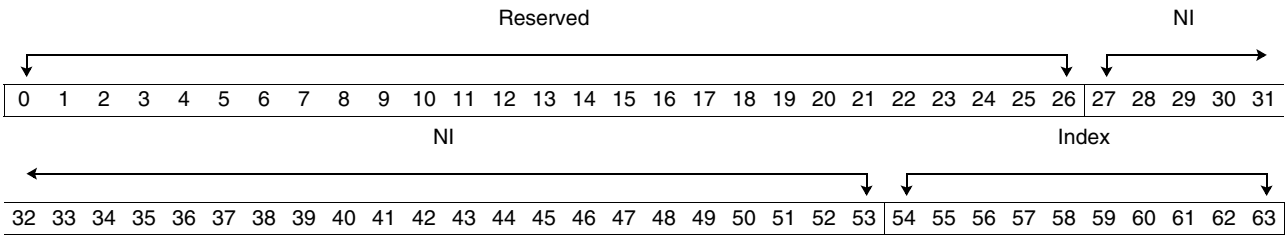
Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:21	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
22:45	HTABORG	Page-table origin (real address of the page table). The HTABORG field contains the high-order 44 bits of the 62-bit real address of the page table. The page table is thus constrained to lie on a minimum 2^{18}-byte (256 KB) boundary. The number of low-order zero bits in HTABORG must be greater than or equal to the value in HTABSIZE.
46:58	Reserved	Bits are not implemented; all bits read back zero.
59:63	HTABSIZE	Encoded size of page table. The HTABSIZE field contains an integer giving the number of bits (in addition to the minimum of 11 bits) from the hash that are used in the page-table index. This number must not exceed 28.

3.2.8.2 MFC TLB Index Hint Register (MFC_TLB_Index_Hint)

This register contains the MFC translation lookaside buffer (TLB) index and congruence class for the most likely candidate for replacement when the SMM has a translation miss in the TLB. The index is written for both hardware and software tablewalks. Software may use this index as a suggestion or make an index of its own.

The CBE uses the lower 12 bits of this register to index the TLB cache. For the PPE_TLB_Index_Hint Register, bits [52:59] contain the TLB congruence class, and bits [60:63] contain the set within the congruence class. For the SPE (MFC_TLB_Index_Hint Register), bits [54:59] contain the TLB congruence class, and bits [60:63] contain the set within the congruence class.

Register Short Name	MFC_TLB_Index_Hint	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400500’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM



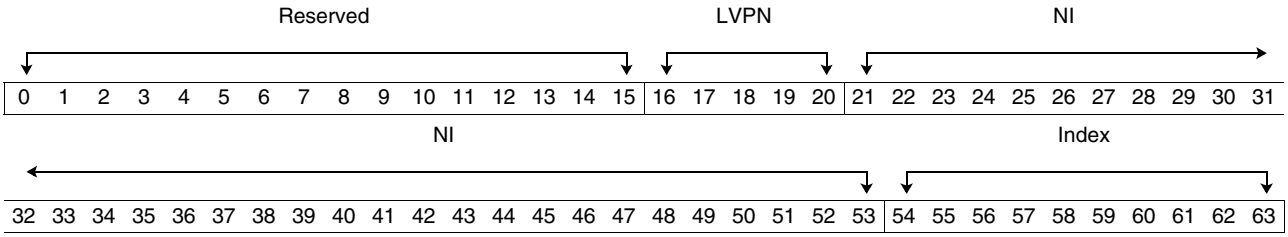
Bits	Field Name	Description
0:26	Reserved	Bits are not implemented; all bits read back zero.
27:53	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
54:63	Index	[54:59] Selects one of the 64 congruence classes [60:63] Selects one of the four entries in the congruence class. These 4 bits decide which entry of the congruence class is updated after a tablewalk. After a tablewalk completes and new data from a PTE is ready to be written into the TLB, these bits indicate the array selected. The SMM hardware writes this register any time there is a TLB miss. Because hardware writes this register directly, only valid combinations are possible. 0000 Does not update a TLB array; causes another tablewalk the next time that address is translated 0001 Selects entry 3 (TLB array 3, the fourth of four entries in the congruence class) 0010 Selects entry 2 (TLB array 2) 0100 Selects entry 1 (TLB array 1) 1000 Selects entry 0 (TLB array 0) All other combinations are not valid. Asserting more than 1 bit writes more than 1 array and corrupts the TLB data, causing a multiple hit on the next address translation.

3.2.8.3 MFC TLB Index Register (MFC_TLB_Index)

The MFC_TLB_Index Register points to a TLB entry (virtual page number [VPN] and real page number [RPN]) to be read or written by the MMIO. This register must be written before any access to the TLB array.

The CBE uses the lower 12 bits of the TLB Index Register to index the TLB cache. For the PPE_TLB_Index Register, bits [52:59] contain the TLB congruence class, and bits [60:63] contain the set within the congruence class. For the SPE (MFC_TLB_Index), bits [54:59] contain the TLB congruence class, and bits [60:63] contain the set within the congruence class.

Register Short Name	MFC_TLB_Index	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400508’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM



Bits	Field Name	Description
0:15	Reserved	Bits are not implemented; all bits read back zero.
16:20	LVPN	These lower virtual page number (LVPN) bits [16:20] correspond to the virtual address bits [57:61]. This field is updated during a VPN read. When writing a TLB entry by using the MMIO, the LVPN data is concatenated with the abbreviated virtual page number (AVPN) data from the MFC_TLB_VPN Register. The LVPN data is the least significant portion of the abbreviated page index in the implemented AVPN field.
21:26	NI	This portion of the LVPN field is not implemented in the CBE, but these bits are defined in the *Cell Broadband Engine Architecture*. All bits read back zero.
27:53	NI	This portion of the Index field is not implemented in the CBE, but these bits are defined in the *Cell Broadband Engine Architecture*. All bits read back zero.
54:63	Index	[54:59] Selects one of the 64 congruence classes. [60:63] Selects one of the four entries in the congruence class. (One-hot decoding. See MFC_TLB_Index_Hint[54:63] for more information.)

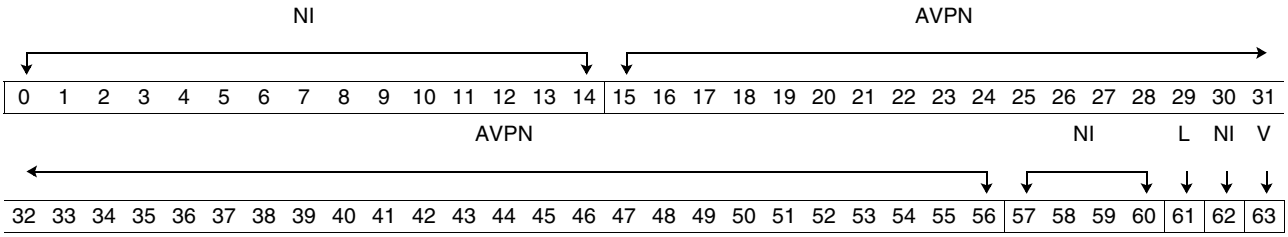
3.2.8.4 MFC TLB Virtual Page Number Register (MFC_TLB_VPN)

The MFC_TLB_VPN Register contains the virtual page number used in translating the effective address to a real address.

To write a new entry in the TLB by the MMIO, the MFC_TLB_Index Register should be written first. The index points to the target congruence class and entry. The RPN data should be written after the index, followed by a write to the VPN register. RPN and VPN write data is collected, and then the entire TLB entry is written at once. Any write to the VPN register sets off a write to the TLB array, whether the index and RPN were previously written or not.

To read the MFC_TLB_VPN data, first write the MFC_TLB_Index to specify the entry. The VPN read command retrieves data from the TLB array entry specified by the MFC_TLB_Index value.

Register Short Name	MFC_TLB_VPN	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400510’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM



Bits	Field Name	Description
0:14	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
15:56	AVPN	Abbreviated virtual page number.
57:60	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
61	L	Large page. 0 4 KB page 1 Large page
62	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
63	V	Valid bit. 0 Invalid 1 Valid

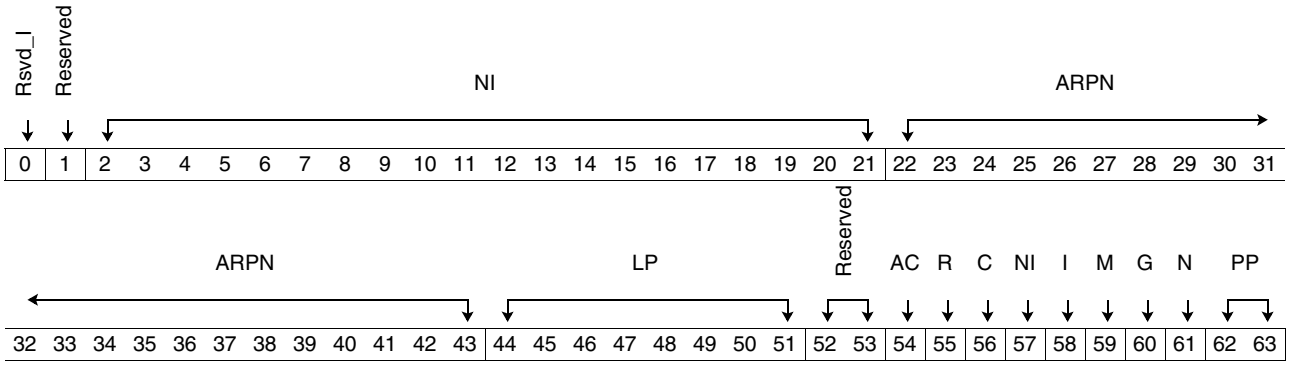
3.2.8.5 MFC TLB Real Page Number Register (MFC_TLB_RPN)

The MFC_TLB_RPN Register contains the real page number used in translating the effective address to a real address.

To write a new entry in the TLB by the MMIO, the MFC_TLB_Index Register should be written first. The index points to the target congruence class and entry. The RPN data should be written after the index, followed by a write to the VPN register. RPN and VPN write data is collected, and then the entire TLB entry is written at once. Any write to the VPN register sets off a write to the TLB array, whether the index and RPN were previously written or not.

To read the MFC_TLB_RPN data, first write the MFC_TLB_Index to specify the entry. The RPN read command retrieves data from the TLB array entry specified by the MFC_TLB_Index value.

Register Short Name	MFC_TLB_RPN	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400518’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_00000110’	**Value During POR Set By**	Scan initialization during POR Bits 55, 59 hardwired
Specification Type	CBEA architected register	**Unit**	SMM



Bits	Field Name	Description
0	Rsvd_I	Reserved. Latch bit is implemented; value read is the value written.
1	Reserved	Bits are not implemented; all bits read back zero.
2:21	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
22:43	ARPN	Abbreviated real page number.
44:51	LP	Size selector for a large virtual page. This field supports up to eight concurrent large page sizes. The MFC supports three large page sizes in total, but only two concurrently. The two concurrent sizes are encoded in the SMM_HID register. This LP field selects one of those two page sizes for the intended TLB entry. The least significant bit of this field is used as part of the comparison to the virtual address on a TLB lookup. Depending on the page size, some bits of this field may be concatenated with the ARPN field to form the RPN on a successful lookup. aaaaaaaa MFC_TLB_VPN[L] = ‘0’ aaaaaaa0 Large page size one if MFC_TLB_VPN[L] = ‘1’ aaaaaa01 Large page size two if MFC_TLB_VPN[L] = ‘1’

Bits	Field Name	Description
52:53	Reserved	Bits are not implemented; all bits read back zero.
54	AC	Address compare bit.
55	R	Reference bit. This bit is not written into the TLB array because it is assumed to be ‘1’. It is returned as ‘1’ during an MMIO RPN read.
56	C	Change bit.
57	NI	This bit is defined in the *Cell Broadband Engine Architecture* but is not implemented in the CBE.
58	I	Cache-inhibited page bit.
59	M	Memory-coherency storage-control bit. This bit is not written into the TLB array because it is assumed to be ‘1’. It is returned as ‘1’ during an MMIO RPN read. The SBI sends local bus command requests with M = ‘1’. However, the EIB can detect whether it is a local (on-chip) address by comparison to the EIB_LBAR0 and EIB_LBAR1 registers and then forcing M = ‘0’. See the EIB Registers section of this document for more information.
60	G	Guarded page bit.
61	N	No-execute page bit.
62:63	PP	Page protection bits.

3.2.8.6 MFC TLB Invalidate Entry Register (MFC_TLB_Invalidate_Entry)

The MFC_TLB_Invalidate_Entry Register is used to invalidate TLB entries in the MFC. The function of this register is similar to the PowerPC **tlbie** instruction. Access to this register is privileged.

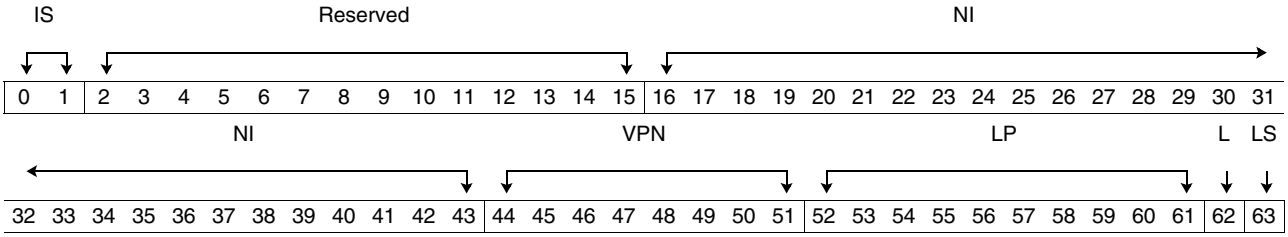
The MFC_TLB_Invalidate_Entry Register contains a virtual page number (VPN) field and an invalidation selector (IS) field. The VPN is used to identify a particular entry to invalidate, and the IS field is used to control how selective the invalidate should be.

This register is not available for the PPE. To invalidate entries in a PPE, the local form of the PowerPC **tlbie** instruction should be used. See the *PowerPC Architecture* document for details of this instruction.

Notes:

- If the VPN is being invalidated to change the protection attributes of a page or to steal the page, a TLB invalidate entry command must be issued to invalidate any cache of the effective-to-real address translation that may be associated with the TLB entry being invalidated. The IS field in the MFC TLB Invalidate Entry Register can only be used to invalidate the cache and does not affect TLB entries.
- Care must be taken in using this function in TLB-managed environments, because hardware can invalidate all TLB entries in the associated congruence class. This can adversely affect TLB set management and deterministic response. To avoid this side effect, privileged software can use the TLB index and TLB direct-modification functions to locate the specific entry to be invalidated in the congruence class and only invalidate the entry that matches.

Register Short Name	MFC_TLB_Invalidate_Entry	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400540’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	N/A	**Value During POR Set By**	N/A
Specification Type	CBEA architected register	**Unit**	SMM



Bits	Field Name	Description
0:1	IS	Invalidation selector 01 The TLB entry is not invalidated. Any lower-level caches of the translation are invalidated. 00, 10, 11 The TLB does a congruence-class invalidate, regardless of logical partition identification (LPID) match.
2:15	Reserved	Bits are not implemented; all bits read back zero.
16:43	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
44:51	VPN	Virtual page number (index).

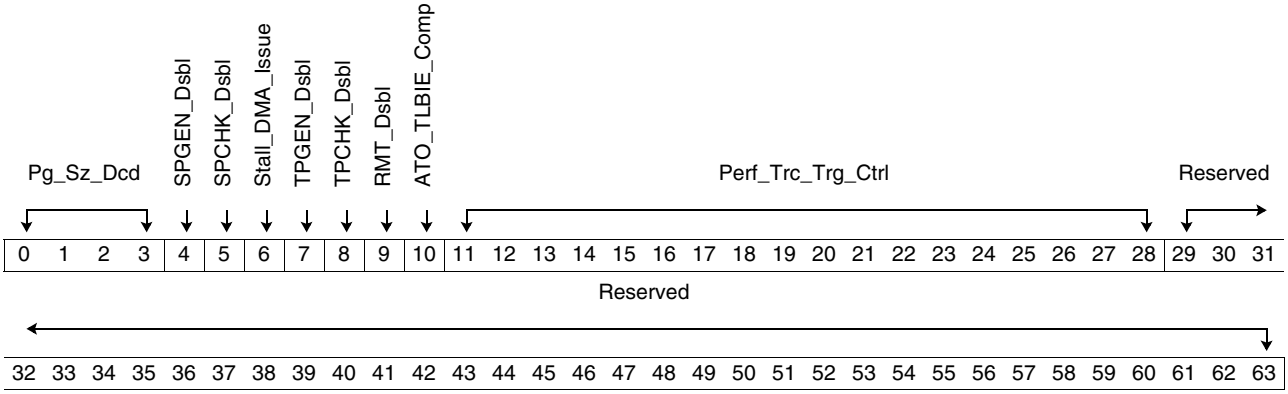
Bits	Field Name	Description
52:61	LP	Size selector for a large virtual page. This field supports up to eight concurrent large page sizes. The MFC supports three large page sizes in total, but only two concurrently. The two concurrent sizes are encoded in the SMM_HID register. This LP field selects one of those two pages. Depending on the page size selected, some bits in this field can be concatenated with the VPN field to determine which entries are invalidated. aaaaaaaa TLB_Invalidate_Entry[L] = '0' aaaaaaa0 Large page size one if TLB_Invalidate_Entry[L] = '1' aaaaaa01 Large page size two if TLB_Invalidate_Entry[L] = '1' **Note:** Software should set the least significant bit of the LP field to the same value as the LS bit.
62	L	Large Page bit The L and LS bits of MFC_TLB_Invalidate_Entry are used in conjunction with the Page Size Decode bits in the SMM_HID register to determine the large page size. 0 Small page (4 KB) 1 Large pages (64 KB, 1 MB, or 16 MB)
63	LS	Large Page Selection The L and LS bits of MFC_TLB_Invalidate_Entry are used in conjunction with the Page Size Decode bits in the SMM_HID register to determine the large page size. 0 First large page (implementation-dependent size) 1 Second large page (implementation-dependent size) **Note:** Software should set this bit to the same value as the least significant bit of the LP field for compatibility with implementations that only support two large page sizes.

3.2.9 Memory Management Register

3.2.9.1 SMM Hardware Implementation Dependent Register (SMM_HID)

The hardware implementation dependent (HID) register bit fields contain configuration bits that control many essential functions of the SMM. This register should be set before the SMM performs any translations and before the arrays are loaded with data.

Register Short Name	SMM_HID	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400580’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SMM



Bits	Field Name	Description
0:3	Pg_Sz_Dcd	Page size decode (HID bits)
4	SPGEN_Dsbl	SLB parity generation 0 Enable 1 Disable
5	SPCHK_Dsbl	SLB parity checking 0 Enable 1 Disable
6	Stall_DMA_Issue	This bit controls DMA requests in response to pending TLBIE conditions. 0 Pending TLBIE condition stalls the DMAC from issuing new requests to the ATO and SBI units 1 Pending TLBIE condition does not stall new DMA requests
7	TPGEN_Dsbl	TLB parity generation 0 Enable 1 Disable
8	TPCHK_Dsbl	TLB parity checking 0 Enable 1 Disable

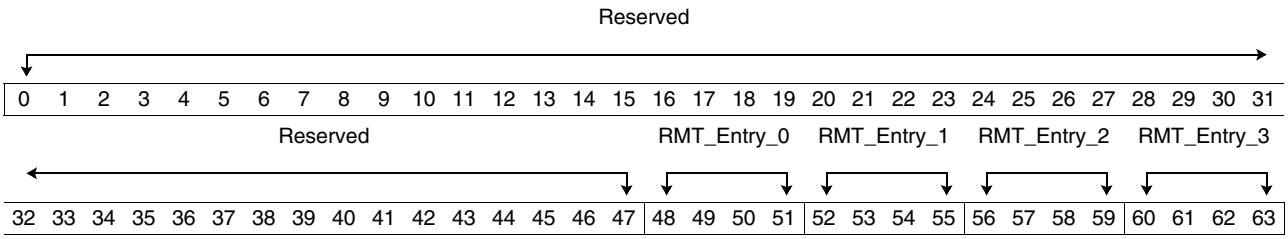
Bits	Field Name	Description
9	RMT_Dsbl	Replacement Management Table (RMT) function 0 Enable 1 Disable
10	ATO_TLBIE_Comp	This bit controls the ATO response to a pending TLBIE under livelock conditions. 0 The ATO does not force completion of a pending TLBIE under livelock conditions (livelock set by RC, castout, or PTE fetch). 1 The ATO forces completion of a pending TLBIE under livelock conditions.
11:28	Perf_Trc_Trg_Ctrl	Performance, trace, and trigger control bits
29:63	Reserved	Bits are not implemented; all bits read back zero.

3.2.10 MFC TLB Replacement Management Table Data Register (MFC_TLB_RMT_Data)

The MFC supports four Replacement Management Table (RMT) entries. The 2 lower bits of the RClassID from a DMA command opcode select one of the four RMT table entries. The RMT entries use 4 bits to determine which TLB entry to replace within a congruence class during a tablewalk. See the cache replacement management facility section of the *Cell Broadband Engine Architecture* document for more information.

The SMM implements Replacement Management Table (RMT) functionality in the same manner as the PPE.

Register Short Name	MFC_TLB_RMT_Data	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400710’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM



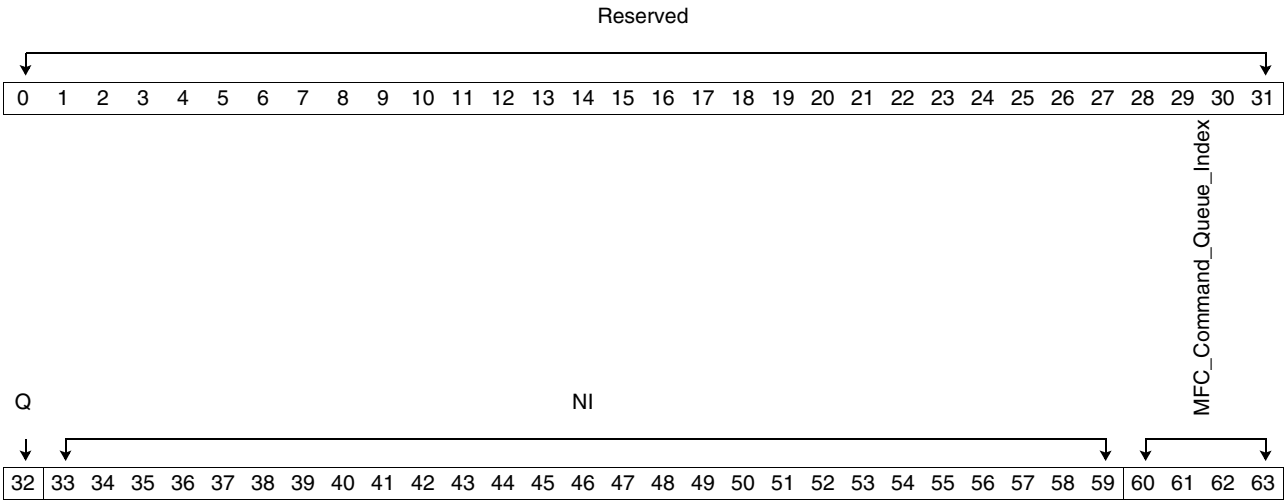
Bits	Field Name	Description
0:47	Reserved	Bits are not implemented; all bits read back zero.
48:51	RMT_Entry_0	RMT Entry 0 set enable bits
52:55	RMT_Entry_1	RMT Entry 1 set enable bits
56:59	RMT_Entry_2	RMT Entry 2 set enable bits
60:63	RMT_Entry_3	RMT Entry 3 set enable bits

3.2.11 MFC Command Data-Storage Interrupt (DSI) Registers

3.2.11.1 MFC Data-Storage Interrupt Pointer Register (MFC_DSIPR)

The MFC_DSIPR Register contains the index (pointer) to the command in the selected command queue that has an MFC data-storage interrupt (DSI) or an MFC data-segment interrupt. The cause of an MFC data-storage interrupt is identified in the *MFC Data-Storage Interrupt Pointer Register (MFC_DSIPR)*.

Register Short Name	MFC_DSIPR	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400800’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32	Q	MFC Command Queue Index 0 MFC proxy command queue index 1 MFC SPU command queue index
33:59	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
60:63	MFC_Command_Queue_Index	Points to the queue entry that caused the data-storage interrupt. The number of bits implemented in this field is implementation dependent: 8 bits for the MFC proxy command queue, 16 bits for the MFC SPU command queue. Bit [60] is ‘0’ for the MFC SPU command queue ID.

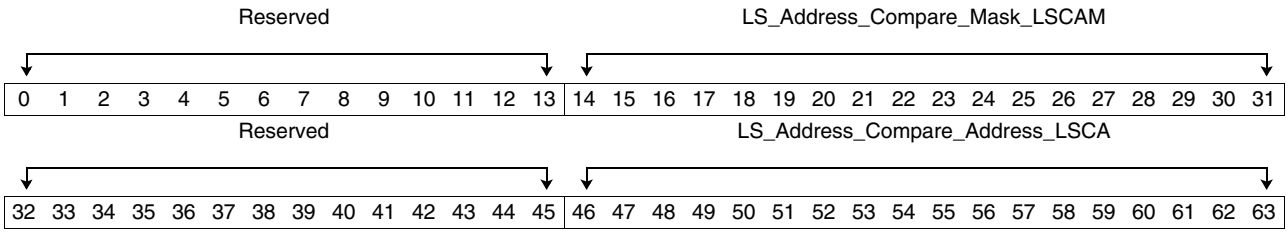
Implementation Note: Only one translation fault can be outstanding. The implementation can either stop all command queue processing on the first translation error or continue processing. If processing is continued, all ordering rules must be followed (a command must not be processed if it is dependent on a command that is waiting for a translation fault to be resolved). The state of the MFC must appear as if the command (or partial command) were never issued. This is also the case if a second translation fault occurs.

3.2.11.2 MFC Local Store Address Compare Register (MFC_LSACR)

The MFC Local Store Address Compare Register (MFC_LSACR) contains the local store address and local store address mask to be used in the MFC local store address compare operation selected by the MFC_ACCR Register. Access to the MFC_LSACR is privileged.

A local store address compare occurs when the local store address accessed is within the range of addresses specified by the bit-wise AND of the local store compare address mask (LSCAM) and the local storage compare address (LSCA) fields.

Register Short Name	MFC_LSACR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400808’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SBI



Bits	Field Name	Description
0:13	Reserved	Bits are not implemented; all bits read back zero.
14:31	LS_Address_Compare_Mask_LSCAM	The local store address compare mask used in the LS address compare operation
32:45	Reserved	Bits are not implemented; all bits read back zero.
46:63	LS_Address_Compare_Address_LSCA	The local store compare address used in the LS address compare operation

3.2.11.3 MFC Local Store Compare Results Register (MFC_LSCRR)

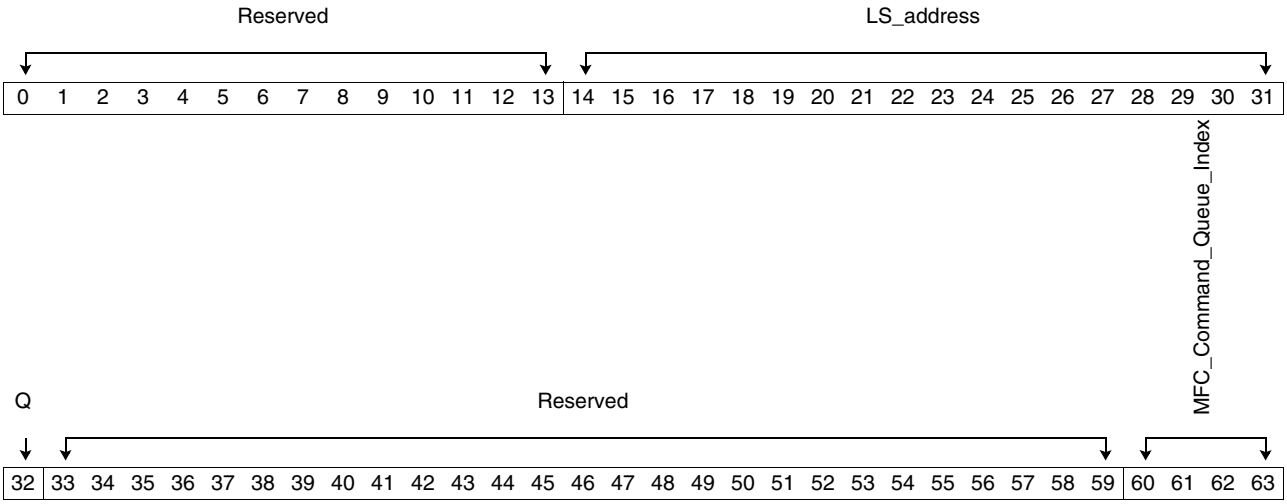
The MFC_LSCRR contains the local store address that triggered the compare, as well as the MFC command queue index of the DMA command that triggered the compare stop.

Contents of this register are only valid when a class 1 interrupt occurs with the INT_Mask_class1[LP] bit or INT_Mask_class1[LG] bit, or both interrupt status bits set. The MFC_LSCRR[Q] bit indicates whether the command was issued from the MFC proxy command queue or the MFC SPU command queue.

Access to this register should be privileged. The contents of this register become indeterminate once MFC operation is resumed.

A new value is captured and locked into the register by the first qualified compare match. The register is unlocked and ready to capture the next value after the register is read or a DMA purge is issued to the MFC_CNTL[PC] Register.

Register Short Name	MFC_LSCRR	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400810’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SBI



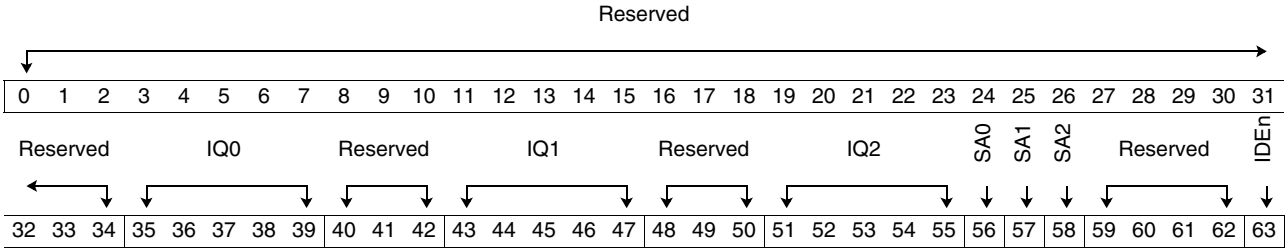
Bits	Field Name	Description
0:13	Reserved	Bits are not implemented; all bits read back zero.
14:31	LS_address	The local store address that triggered the compare match
32	Q	MFC command type 0 Compare match is triggered by an MFC proxy command. 1 Compare match is triggered by an MFC SPU command.
33:59	Reserved	Bits are not implemented; all bits read back zero.
60:63	MFC_Command_Queue_Index	Points to the MFC command queue entry that triggered the compare match. Bit [60] is always zero for the MFC proxy command queue index ID.

3.2.11.4 MFC Transfer Class ID Register (MFC_TClassID)

When slot alternation is enabled, the **put** command is placed into slot 0 and the **get** command is placed into slot 1. When slot alternation is disabled, the corresponding Transfer Class ID group is always placed in and issued from slot 0.

The outgoing queue size for bus transactions is 16, therefore the total of the values in IQ0, IQ1, and IQ2 must be clamped to 16.

Register Short Name	MFC_TClassID	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400820’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_10000000’	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32:34	Reserved	Bits are not implemented; all bits read back zero.
35:39	IQ0	Issue quota for TClassID0. Initial value after purge is ‘10000’. 00000 quota value of 1 00001 quota value of 1 00010 - 10000 quota value of 2 - 16 10001 - 11111 quota value of 16
40:42	Reserved	Bits are not implemented; all bits read back zero.
43:47	IQ1	Issue quota for TClassID1 00000 quota value of 1 00001 quota value of 1 00010 - 10000 quota value of 2 - 16 10001 - 11111 quota value of 16
48:50	Reserved	Bits are not implemented; all bits read back zero.
51:55	IQ2	Issue quota for TClassID2 00000 quota value of 1 00001 quota value of 1 00010 - 10000 quota value of 2 - 16 10001 - 11111 quota value of 16
56	SA0	Slot alternation 0 Enabled for TClassID0 1 Disabled for TClassID0

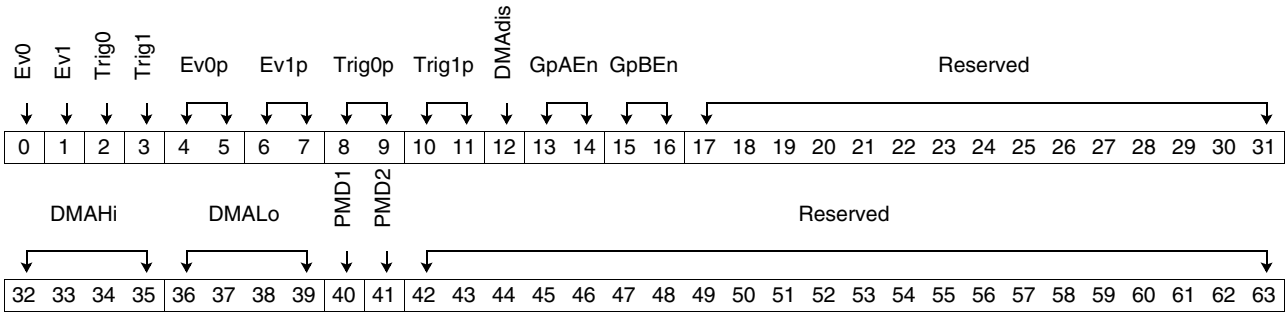
Bits	Field Name	Description
57	SA1	Slot alternation 0 Enabled for TClassID1 1 Disabled for TClassID1
58	SA2	Slot alternation 0 Enabled for TClassID2 1 Disabled for TClassID2
59:62	Reserved	Bits are not implemented; all bits read back zero.
63	IDEn	TClassID enable bit. Also called the streaming hint enable bit 0 Transfer class ID ignored; all transfers default to TClassID0. 1 Bus transactions issued round-robin for TClassID0, TClassID1, and TClassID2.

3.2.12 DMAC Unit Performance Monitor Control Register

3.2.12.1 DMAC Unit Performance Monitor Control Register (DMAC_PMCR)

This register provides controls for selecting and enabling signals for the trigger and event buses and groups of signals for the trace bus and performance monitor.

Register Short Name	DMAC_PMCR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400880’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



Bits	Field Name	Description
0	Ev0	Event 0 enable. Set to ‘1’ to enable an MFC proxy command queue full event
1	Ev1	Event 1 enable. Set to ‘1’ to enable an MFC SPU command queue full event
2	Trig0	Trigger 0 enable. Set to ‘1’ to enable an MFC command or alignment error
3	Trig1	Trigger 1 enable. Set to ‘1’ to enable atomic access to a cache-inhibited page (ATO CI)
4:5	Ev0p	Event 0 position select. Places event 0 onto d_event_out[0:3] 00 Place event 0 onto bit 0 01 Place event 0 onto bit 1 10 Place event 0 onto bit 2 11 Place event 0 onto bit 3
6:7	Ev1p	Event 1 position select. Places event 1 onto d_event_out[0:3] 00 Place event 1 onto bit 0 01 Place event 1 onto bit 1 10 Place event 1 onto bit 2 11 Place event 1 onto bit 3
8:9	Trig0p	Trigger 0 position select. Places trigger 0 onto d_trigger_out[0:3] 00 Place trigger 0 onto bit 0 01 Place trigger 0 onto bit 1 10 Place trigger 0 onto bit 2 11 Place trigger 0 onto bit 3

Bits	Field Name	Description
10:11	Trig1p	Trigger 1 position select. Places trigger 1 onto d_trigger_out[0:3] 00 Place trigger 1 onto bit 0 01 Place trigger 1 onto bit 1 10 Place trigger 1 onto bit 2 11 Place trigger 1 onto bit 3
12	DMAdis	Activate disable for DMA trace output latches. Set to '1' to disable tracing SPU or DMA trace or performance data. This can be set when the Performance Monitor and Trace Bus are not in use by the DMAC or SPU (SPU trace data passes through the DMA) in order to save power by disabling the latches used for tracing.
13:14	GpAEn	Group A Enable: 00 Select neither group 10 Select GroupA0: REQIF0 Trace Data 01 Select GroupA1: REQIF1 Trace Data
15:16	GpBEn	Group B Enable: 00 Select neither group 10 Select GroupB0: MMIO Trace Data 01 Select GroupB1: TAG Trace Data
17:31	Reserved	Bits are not implemented; all bits read back zero.
32:35	DMAHi	DMA Group High data select for d_trc_data_out[64:95] 1000 Select Group A[0:31] 0100 Select Group B[0:31] 0010 Select Group C[0:31] 0001 Select Group D[0:31]
36:39	DMALo	DMA Group Low data select for d_trc_data_out[96:127] 1000 Select Group A[32:63] 0100 Select Group B[32:63] 0010 Select Group C[32:63] 0001 Select Group D[32:63]
40	PMD1	Enable DMA Performance Monitor Data 1 to d_trc_data_out[0:31]
41	PMD2	Enable DMA Performance Monitor Data 2 to d_trc_data_out[64:95]
42:63	Reserved	Bits are not implemented; all bits read back zero.

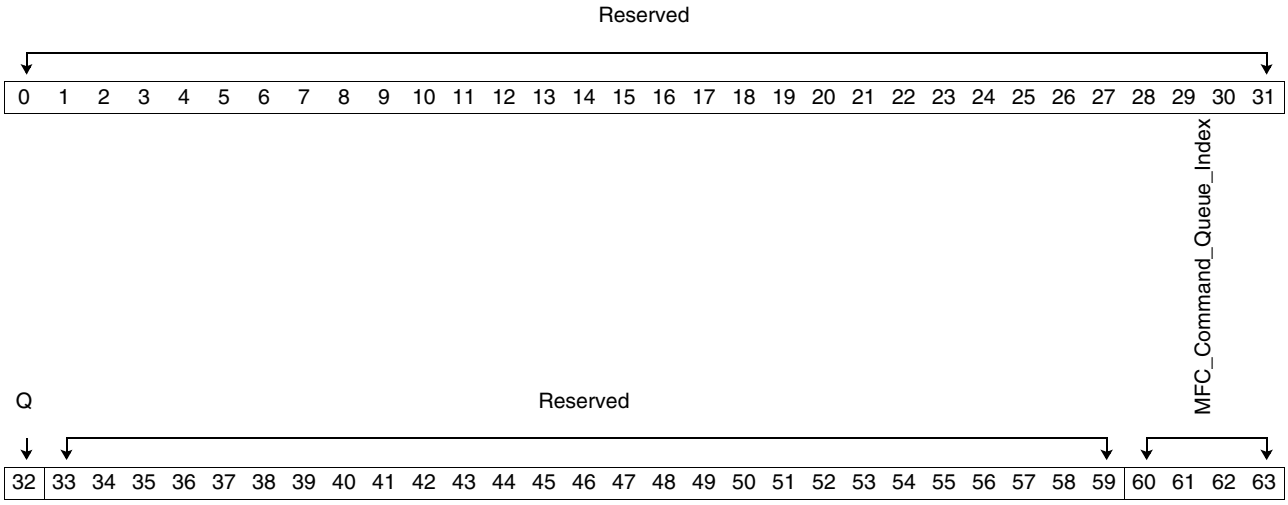
3.2.13 MFC Command Error Area

3.2.13.1 MFC Command Error Register (MFC_CER)

The MFC_CER Register contains the MFC command queue entry index of the command that generated the invalid DMA command interrupt or the DMA alignment interrupt.

The MFC must stop execution on the first error. The MFC_CER Register must point to the command that caused the first error.

Register Short Name	MFC_CER	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400C00’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SBI



Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32	Q	MFC command queue index 0 MFC proxy command queue index 1 MFC SPU command queue index
33:59	Reserved	Bits are not implemented; all bits read back zero.
60:63	MFC_Command_Queue_Index	Points to the MFC command queue entry that caused the command error. Bit [60] is always ‘0’ for the MFC proxy command queue index.

3.2.14 SPU ECC and Error Mask Registers

The SPU uses an error checking and correcting (ECC) mechanism to protect the SPU local-store memory from soft errors. A soft error is a bit that changes in memory without being written.

3.2.14.1 SPU ECC Control Register (SPU_ECC_Cntl)

This is an implementation-specific register that controls the operation of the SPU ECC memory-checking mechanism. See *SPU ECC Status Register (SPU_ECC_Stat)* on page 92 for more information.

Register Short Name	SPU_ECC_Cntl	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘401000’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_00000003’	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC

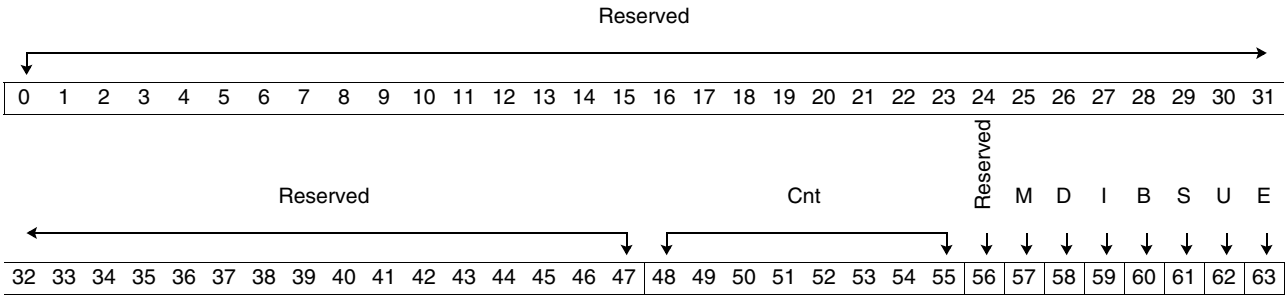
Bits	Field Name	Description
0:57	Reserved	Bits are not implemented; all bits read back zero.
58	D	ECC detection disable. 0 Enable ECC detection and correction 1 Disable ECC detection and correction **Note:** Only enable this bit when the SPU is not performing DMA transfers and is not running. For example, enable this bit when no ECC detection is occurring and there is no ECC error status. ECC generation continues, so that if the ECC is reenabled, the local store contains the correct ECC values.
59:60	I	ECC error injection (used for chip validation). 00 The SPU stores DMA write data correctly to the local store. 11 The SPU stores all DMA full quadword stores to the local store with an uncorrectable two-bit ECC error. 10, 01 The SPU stores all DMA full quadword stores to the local store with a correctable single-bit ECC error. SPU_ECC_Cntl[59:60] are XORed with DMA write-data bits [0:1] as the data is loaded into the local store.
61	B	Start background ECC local-store scrubbing. Used to continuously run ECC local-store scrubbing. 0 The SPU only scrubs when uncorrectable ECC errors are detected. 1 The SPU immediately starts scrubbing the SPU local store unless disabled by the ECC local-store scrubbing enable bit [63]. Contact your IBM technical support representative for more information about scrubbing and data ECC error detection and recovery.

Bits	Field Name	Description
62	S	ECC error stop enable. 0 The SPU does not detect when an uncorrectable ECC error occurs and does not generate a DERR on the EIB. Also, class 0 error interrupts are not generated. 1 The SPU stops when an uncorrectable ECC error occurs, or when it receives a DERR from the EIB and generates a DERR on the EIB any time the local store is transferring data onto the EIB. The address is not saved when an ECC error occurs.
63	E	ECC local-store scrubbing enable. If disabled, ECC errors are not corrected in the local store except for instruction fetch ECC errors. 0 ECC local-store scrubbing disabled 1 ECC local-store scrubbing enabled

3.2.14.2 SPU ECC Status Register (SPU_ECC_Stat)

This implementation-specific register contains the status of ECC errors and repair operation. These are rare but can occur at any time. ‘0’ must be written to this register to clear the status bits. Writing ‘1’ sets the status bit, but this does not affect any other SPU operations.

Register Short Name	SPU_ECC_Stat	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘401008’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



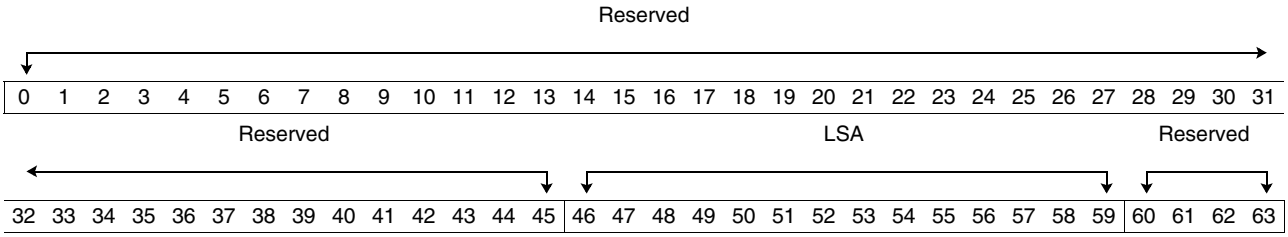
Bits	Field Name	Description
0:47	Reserved	Bits are not implemented; all bits read back zero.
48:55	Cnt	Count of local-store quadwords repaired by ECC scrubbing or by an instruction ECC error (clamped at x‘FF’).
56	Reserved	Bit is not implemented; bit reads back zero.
57	M	DMA ECC error. 0 The SPU has not detected a load DMA ECC error. 1 The SPU has detected one or more DMA ECC errors. One or more ECC errors were detected on DMA transfers. The error is repaired inline if it is a correctable ECC error. DMA correctable ECC errors repaired inline are not counted in ECC repair count. If an uncorrectable error is detected, this bit and the ECC error stop are set. This bit indicates that a DMA write DERR was received from the MFC only if uncorrectable ECC is enabled.

Bits	Field Name	Description
58	D	Data ECC error. 0 The SPU has not detected a load instruction ECC error. 1 The SPU has detected one or more load instruction ECC errors. One or more ECC errors were detected during an SPU local store load instruction. The error is repaired inline if it is a correctable ECC error. DMA data-correctable ECC errors repaired inline are not counted in ECC repair count. If an uncorrectable error is detected, this bit and the ECC error stop [62] are set.
59	I	Instruction ECC error. 0 The SPU has not detected an instruction ECC error. 1 The SPU has detected one or more Instruction ECC errors. One or more ECC errors were detected on an instruction fetch. The error is repaired by ECC scrubbing logic if it is a correctable ECC error. Instruction-correctable ECC errors repaired are counted in ECC repair count. If an uncorrectable error is detected, this bit and the ECC error stop [62] are set only if uncorrectable ECC detection is enabled. See the *SPU_ECC_Cntl* Register for more information.
60	B	ECC scrub in process. Read only. 0 The SPU is not currently running an ECC scrub or repair. 1 The SPU is currently running a local-store ECC scrub or repair.
61	S	ECC scrub completed. 0 The SPU has not completed a scrub of the local store. 1 The SPU has completed one or more scrubs of the full local store.
62	U	ECC error stop. 0 The SPU has not detected an uncorrectable ECC error. 1 The SPU has detected, stopped, and generated a class 0 interrupt for an uncorrectable ECC error. Uncorrectable ECC errors found during scrubbing are not logged and are not repaired.
63	E	ECC error status. 0 The SPU has not detected an ECC error. 1 The SPU has detected and repaired an ECC error in the SPU local store that was found during scrubbing or for an instruction ECC error.

3.2.14.3 SPU ECC Address Register (SPU_ECC_Addr)

This implementation-specific register contains the local-store address of the ECC error most recently repaired by the SPU ECC scrubbing logic. Uncorrectable ECC errors are not logged. ECC errors are rare but can occur at any time.

Register Short Name	SPU_ECC_Addr	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘401010’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



Bits	Field Name	Description
0:45	Reserved	Bits are not implemented; all bits read back zero.
46:59	LSA	Local store quadword address
60:63	Reserved	Bits are not implemented; all bits read back zero.

3.2.14.4 SPU Error Mask Register (SPU_ERR_Mask)

Note: Reads and writes can be to the full 64 bits or to only the lower-word bits [32:63].

Register Short Name	SPU_ERR_Mask	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘401018’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_00000001’	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC

Bits	Field Name	Description
0:62	Reserved	Bits are not implemented; all bits read back zero.
63	I	Invalid instruction interrupt enable 0 Invalid instruction interrupt generation disabled. 1 Invalid instruction interrupt generation enabled. Generates a class 0 SPU error interrupt in INT_Stat_class0[61]. **Note:** Illegal operation detection and SPU halting are always enabled.

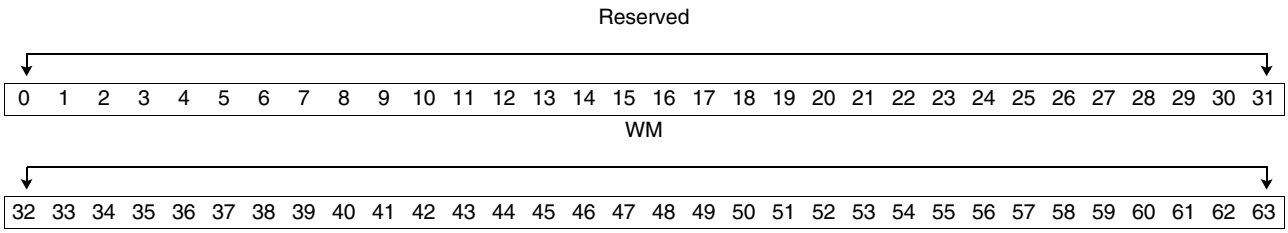
3.2.15 MFC Performance Monitor Register

3.2.15.1 Performance Monitor/Trace Tag Status Wait Mask Register (PM_Trace_Tag_Wait_Mask)

The wait mask is applied to the tag status information in the MFC_RdTagStat channel. If all tag status bits are set to 0 (the tag group has outstanding operations or has been disabled by the query mask), then the mask waits for bits to be set to '1' (the tag group has no outstanding operations or was not disabled by the query mask).

In this wait mask register, bit [32] corresponds to tag group 31, and bit [63] corresponds to tag group 0.

Register Short Name	PM_Trace_Tag_Wait_Mask	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x'401400' + (x'02000' x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32:63	WM	Wait mask (for each bit in field) 0 Waiting on all masked tags. 1 At least one masked tag is complete.

3.3 SPE Privilege 2 Memory Map Registers

This section lists the registers included in the SPE Privilege 2 memory map.

3.3.1 SLB Management Registers

These registers maintain the SLB array contents.

3.3.1.1 SLB Index Register (SLB_Index)

To properly load the array with data, the SLB requires a write sequence similar to the one for the TLB. First, the index must be written to specify the entry for loading. The VSID and ESID fields are written independently, unlike the TLB writes. The SLB_VSID write should follow the index. The SLB_ESID data is written last because it contains the Valid bit, and the entry should not be valid until all data is loaded.

The CBE implements the 3 least significant bits for the MMU. These registers are not available for the PPE.

See the *Cell Broadband Engine Architecture* document for more information about this register.

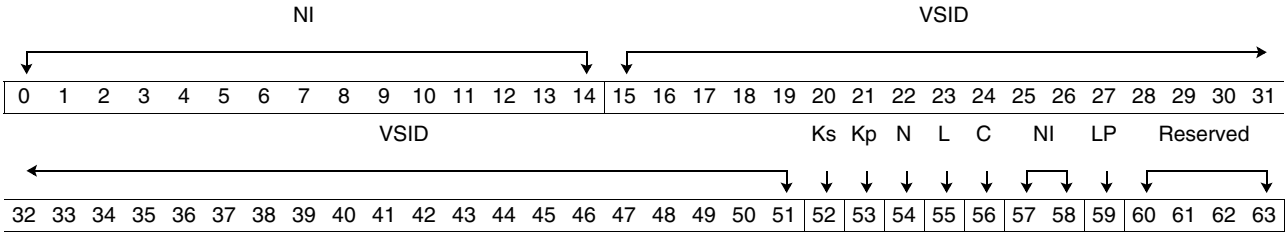
Register Short Name	SLB_Index	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘061108’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM

Bits	Field Name	Description
0:51	Reserved	Bits are not implemented; all bits read back zero.
52:60	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
61:63	SLB_Index	Bits are encoded to select one of eight SLB entries. The SLB_Index-implemented bits are defined in the range [61:63] (3 bits total). The eight possible index values are the eight binary combinations of those 3 bits: ‘000’, ‘001’, ‘010’, ‘011’, and so forth. The SLB array entries are most commonly referred to by the decimal equivalent of those binary numbers: 0, 1, 2, ..., 7.

3.3.1.2 SLB Virtual Segment ID Register (SLB_VSID)

To properly load the array with data, the SLB requires a write sequence similar to the one for the TLB. First, the index must be written to specify the entry for loading. The VSID and ESID fields are written independently, unlike the TLB writes. The SLB_VSID write should follow the index. The SLB_ESID data is written last because it contains the Valid bit, and the entry should not be valid until all data is loaded.

Register Short Name	SLB_VSID	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘061118’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM

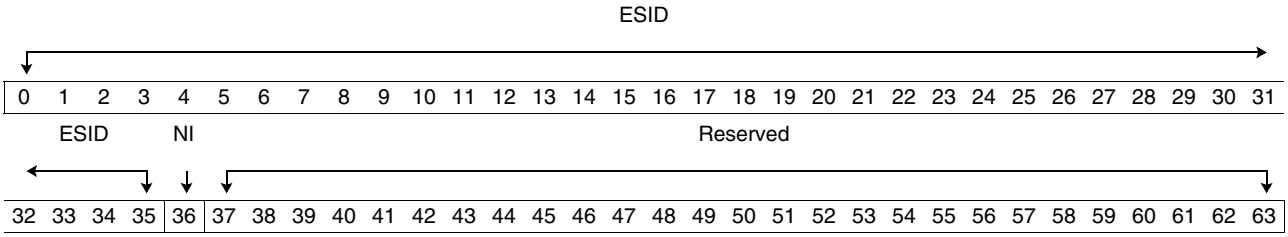


Bits	Field Name	Description
0:14	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
15:51	VSID	Virtual Segment ID.
52	Ks	Storage Key supervisor (privileged) state.
53	Kp	Storage Key problem state.
54	N	No Execute Segment.
55	L	Large page bit. 1 Large page (64 KB, 1 MB, 16 MB) 0 Small page (4 KB)
56	C	Class bit.
57:58	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
59	LP	Large page-size bit.
60:63	Reserved	Bits are not implemented; all bits read back zero.

3.3.1.3 SLB Invalidate Entry Register (SLB_Invalidate_Entry)

Software uses this register to maintain the SLB and update entries.

Register Short Name	SLB_Invalidate_Entry	**Privilege Type**	Privilege 2
Access Type	MMIO Write Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘061120’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	SMM



Bits	Field Name	Description
0:35	ESID	Effective Segment ID.
36	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
37:63	Reserved	Bits are not implemented; all bits read back zero.

3.3.2 Context Save and Restore Register

The hardware supports the capability to suspend a task on the SPE, fully save its context, fully restore that context at a later time, and resume the task.

3.3.2.1 MFC Command Queue Context Save/Restore Register (MFC_CQ_SR)

Table 3-4 through *Table 3-10* define the MFC Command Queue Context Save/Restore Register.

Register Short Name	MFC_CQ_SR	**Privilege Type**	Privileged
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset Address	x‘02000’ – ‘022FF’ See *Table 3-4* through *Table 3-10*	**From Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	SMF

Context for the MFC command queue and issue logic is accessed starting at offset x‘2000’, as shown in *Table 3-4* on page 101. The MFC command queues contain 16 entries for the SPU and eight for the PPU, and each MFC command queue entry is accessed with three MMIO doublewords. The contents of doubleword 0 (DW0) are listed in *Table 3-5* on page 102. The contents of doubleword 1 (DW1) are listed in *Table 3-6* on page 103. The contents of doubleword 2 (DW2) are listed in *Table 3-7* on page 103.

The rightmost two columns of *Table 3-4* describe the fourth doubleword for each entry in the MFC command queue. For the first 16 entries, the doubleword is divided into the SPU Issue Word (SIW) and the Tag Completion Word. For the last eight entries, the doubleword is divided into the PPU Issue Word (PIW) and a Reserved Word. The format of the SPU Issue Word is described in *Table 3-8* on page 103. The format of the PPU Issue Word is described in *Table 3-9* on page 103. *Table 3-10* on page 104 describes the data in the Tag Completion Word for the first 16 entries. The Tag Completion Word contains the counts for two tag groups for the SPU and PPU, the stall, list, and finish bits for each SPU entry, and the start and finish bits for each PPU entry (only in the first eight Tag Completion Words).

See the *Cell Broadband Engine Architecture* document for more general information about Context Save and Restore.

Table 3-4. Definitions for the MFC Context Save/Restore Registers (Read/Write) (Page 1 of 2)

Address Offset	+ x'000(CMDQ_DW0)[1]	+ x'008(CMDQ_DW1)[2]	+ x'010u(CMDQ_DW2)[3]	+ x'018 (Issue_Word) (Issue Machine States) SIW[4]/PIW[5]	+ x'01C (Tag_word)
x'02000'	MFC SPUQ Entry 0, DW0	MFC SPUQ Entry 0, DW1	MFC SPUQ Entry 0, DW2	MFC SPUQ Entry 0, SIW	See *Section 3-10* on page 104
x'02020'	MFC SPUQ Entry 1, DW0	MFC SPUQ Entry 1, DW1	MFC SPUQ Entry 1, DW2	MFC SPUQ Entry 1, SIW	See *Section 3-10* on page 104
x'02040'	MFC SPUQ Entry 2, DW0	MFC SPUQ Entry 2, DW1	MFC SPUQ Entry 2, DW2	MFC SPUQ Entry 2, SIW	See *Section 3-10* on page 104
x'02060'	MFC SPUQ Entry 3, DW0	MFC SPUQ Entry 3, DW1	MFC SPUQ Entry 3, DW2	MFC SPUQ Entry 3, SIW	See *Section 3-10* on page 104
x'02080'	MFC SPUQ Entry 4, DW0	MFC SPUQ Entry 4, DW1	MFC SPUQ Entry 4, DW2	MFC SPUQ Entry 4, SIW	See *Section 3-10* on page 104
x'020A0'	MFC SPUQ Entry 5, DW0	MFC SPUQ Entry 5, DW1	MFC SPUQ Entry 5, DW2	MFC SPUQ Entry 5, SIW	See *Section 3-10* on page 104
x'020C0'	MFC SPUQ Entry 6, DW0	MFC SPUQ Entry 6, DW1	MFC SPUQ Entry 6, DW2	MFC SPUQ Entry 6, SIW	See *Section 3-10* on page 104
x'020E0'	MFC SPUQ Entry 7, DW0	MFC SPUQ Entry 7, DW1	MFC SPUQ Entry 7, DW2	MFC SPUQ Entry 7, SIW	See *Section 3-10* on page 104
x'02100'	MFC SPUQ Entry 8, DW0	MFC SPUQ Entry 8, DW1	MFC SPUQ Entry 8, DW2	MFC SPUQ Entry 8, SIW	See *Section 3-10* on page 104
x'02120'	MFC SPUQ Entry 9, DW0	MFC SPUQ Entry 9, DW1	MFC SPUQ Entry 9, DW2	MFC SPUQ Entry 9, SIW	See *Section 3-10* on page 104
x'02140'	MFC SPUQ Entry a, DW0	MFC SPUQ Entry a, DW1	MFC SPUQ Entry a, DW2	MFC SPUQ Entry a, SIW	See *Section 3-10* on page 104
x'02160'	MFC SPUQ Entry b, DW0	MFC SPUQ Entry b, DW1	MFC SPUQ Entry b, DW2	MFC SPUQ Entry b, SIW	See *Section 3-10* on page 104
x'02180'	MFC SPUQ Entry c, DW0	MFC SPUQ Entry c, DW1	MFC SPUQ Entry c, DW2	MFC SPUQ Entry c, SIW	See *Section 3-10* on page 104
x'021A0'	MFC SPUQ Entry d, DW0	MFC SPUQ Entry d, DW1	MFC SPUQ Entry d, DW2	MFC SPUQ Entry d, SIW	See *Section 3-10* on page 104
x'021C0'	MFC SPUQ Entry e, DW0	MFC SPUQ Entry e, DW1	MFC SPUQ Entry e, DW2	MFC SPUQ Entry e, SIW	See *Section 3-10* on page 104

1. For a definition of this format, see *Table 3-5 Context Save/Restore MFC Command Queue Doubleword 0 (CSR_CMDQ_DW0)* on page 102.
2. For a definition of this format, see *Table 3-6 Context Save/Restore MFC Command Queue Doubleword 1 (CSR_CMDQ_DW1)* on page 103.
3. For a definition of this format, see *Table 3-7 Context Save/Restore MFC Command Queue Doubleword 2 (CSR_CMDQ_DW02)* on page 103.
4. For a definition of this format, see *Table 3-8 Context Save/Restore for MFC SPU Command Queue Issue State Machine* on page 103.
5. For a definition of this format, see *Table 3-9 Context Save/Restore for MFC Proxy Command Queue Issue Machine State* on page 103.

Table 3-4. Definitions for the MFC Context Save/Restore Registers (Read/Write) (Page 2 of 2)

Address Offset	+ x'000(CMDQ_DW0)[1]	+ x'008(CMDQ_DW1)[2]	+ x'010u(CMDQ_DW2)[3]	+ x'018 (Issue_Word) (Issue Machine States) SIW[4]/PIW[5]	+ x'01C (Tag_word)
x'021E0'	MFC SPUQ Entry f, DW0	MFC SPUQ Entry f, DW1	MFC SPUQ Entry f, DW2	MFC SPUQ Entry f, SIW	See *Section 3-10* on page 104
x'02200'	MFC PrxyQ Entry 0, DW0	MFC PrxyQ Entry 0, DW1	MFC PrxyQ Entry 0, DW2	MFC PrxyQ Entry 0, PIW	Reserved
x'02220'	MFC PrxyQ Entry 1, DW0	MFC PrxyQ Entry 1, DW1	MFC PrxyQ Entry 1, DW2	MFC PrxyQ Entry 1, PIW	Reserved
x'02240'	MFC PrxyQ Entry 2, DW0	MFC PrxyQ Entry 2, DW1	MFC PrxyQ Entry 2, DW2	MFC PrxyQ Entry 2, PIW	Reserved
x'02260'	MFC PrxyQ Entry 3, DW0	MFC PrxyQ Entry 3, DW1	MFC PrxyQ Entry 3, DW2	MFC PrxyQ Entry 3, PIW	Reserved
x'02280'	MFC PrxyQ Entry 4, DW0	MFC PrxyQ Entry 4, DW1	MFC PrxyQ Entry 4, DW2	MFC PrxyQ Entry 4, PIW	Reserved
x'022A0'	MFC PrxyQ Entry 5, DW0	MFC PrxyQ Entry 5, DW1	MFC PrxyQ Entry 5, DW2	MFC PrxyQ Entry 5, PIW	Reserved
x'022C0'	MFC PrxyQ Entry 6, DW0	MFC PrxyQ Entry 6, DW1	MFC PrxyQ Entry 6, DW2	MFC PrxyQ Entry 6, PIW	Reserved
x'022E0'	MFC PrxyQ Entry 7, DW0	MFC PrxyQ Entry 7, DW1	MFC PrxyQ Entry 7, DW2	MFC PrxyQ Entry 7, PIW	Reserved

1. For a definition of this format, see *Table 3-5 Context Save/Restore MFC Command Queue Doubleword 0 (CSR_CMDQ_DW0)* on page 102.
2. For a definition of this format, see *Table 3-6 Context Save/Restore MFC Command Queue Doubleword 1 (CSR_CMDQ_DW1)* on page 103.
3. For a definition of this format, see *Table 3-7 Context Save/Restore MFC Command Queue Doubleword 2 (CSR_CMDQ_DW02)* on page 103.
4. For a definition of this format, see *Table 3-8 Context Save/Restore for MFC SPU Command Queue Issue State Machine* on page 103.
5. For a definition of this format, see *Table 3-9 Context Save/Restore for MFC Proxy Command Queue Issue Machine State* on page 103.

Table 3-5. Context Save/Restore MFC Command Queue Doubleword 0 (CSR_CMDQ_DW0)

CSR_CMDQ_DW0	Description
0:14	List Address[0:14]
15:26	List Size[0:11]
27:34	MFC Command Opcode[0:7]
35:39	MFC Command Tag[0:4]
40	List Valid Bit
41:43	RclassID[0:2]
44:46	TclassID[0:2]
47:63	Reserved

Table 3-6. Context Save/Restore MFC Command Queue Doubleword 1 (CSR_CMDQ_DW1)

CS_CMDQ_DW1	Description
0:51	Effective Address (EA)[0:51]
52:63	Reserved

Table 3-7. Context Save/Restore MFC Command Queue Doubleword 2 (CSR_CMDQ_DW02)

CS_CMDQ_DW2	Description
0:13	Local-Store Address[0:13]
14:24	Transfer Size[0:10]
25:36	Effective Address[52:63]
37	No-op Valid Bit
38	Quadword (QW) or Multiple QW Valid Bit
39	EA Valid Bit
40	Command Error Bit
41:63	Reserved

Table 3-8. Context Save/Restore for MFC SPU Command Queue Issue State Machine

CS_SPU_Issue_Word	Description
0:15	Entry Dependency State
16	Entry Valid
17:18	TclassID
19:23	MFC Command Tag[0:4]
24	0 DMA **get** command 1 DMA **put** command
25	Last Bit
26:27	Dependency Type 00 Normal 01 MFC command with barrier modifier 10 Barrier, **mfcsync**, or **mfceieio** DMA command 11 Reserved
28	Stall Bit
29	Issue Dependency Bit
30:31	Reserved

Table 3-9. Context Save/Restore for MFC Proxy Command Queue Issue Machine State

CS_Prxy_Issue_Word	Description
0:7	Entry Dependency State
8:15	Reserved
16	Entry Valid

Table 3-9. Context Save/Restore for MFC Proxy Command Queue Issue Machine State

CS_Prxy_Issue_Word	Description
17:18	TclassID
19:23	MFC Command Tag[0:4]
24	0 DMA **get** command 1 DMA **put** command
25	Last Bit
26:27	Dependency Type 00 Normal 01 MFC command with barrier modifier 10 Barrier, **mfcsync**, or **mfceieio** MFC command 11 Reserved
28	Reserved
29	Issue Dependency Bit
30:31	Reserved

Table 3-10. Context Save/Restore for Tag Completion Machine State

Word Address	SPU Tag Group Count		PPU Tag Group Count		MFC SPUQ Stall Bit	MFC SPUQ List Bit	MFC SPUQ Finish Bit	MFC ProxyQ Start Bit	MFC ProxyQ Finish Bit	Not Used
Bit Position in Word	[0:4]	[5:9]	[10:13]	[14:17]	[18]	[19]	[20]	[21]	[22]	[23:31]
x‘0201C’	tag 0	tag 1	tag 0	tag 1	Entry 0	Entry 0	Entry 0	Entry 0	Entry 0	Reserved
x‘0203C’	tag 2	tag 3	tag 2	tag 3	Entry 1	Entry 1	Entry 1	Entry 1	Entry 1	Reserved
x‘0205C’	tag 4	tag 5	tag 4	tag 5	Entry 2	Entry 2	Entry 2	Entry 2	Entry 2	Reserved
x‘0207C’	tag 6	tag 7	tag 6	tag 7	Entry 3	Entry 3	Entry 3	Entry 3	Entry 3	Reserved
x‘0209C’	tag 8	tag 9	tag 8	tag 9	Entry 4	Entry 4	Entry 4	Entry 4	Entry 4	Reserved
x‘020BC’	tag 10	tag 11	tag 10	tag 11	Entry 5	Entry 5	Entry 5	Entry 5	Entry 5	Reserved
x‘020DC’	tag 12	tag 13	tag 12	tag 13	Entry 6	Entry 6	Entry 6	Entry 6	Entry 6	Reserved
x‘020FC’	tag 14	tag 15	tag 14	tag 15	Entry 7	Entry 7	Entry 7	Entry 7	Entry 7	Reserved
x‘0211C’	tag 16	tag 17	tag 16	tag 17	Entry 8	Entry 8	Entry 8	Reserved	Reserved	Reserved
x‘0213C’	tag 18	tag 19	tag 18	tag 19	Entry 9	Entry 9	Entry 9	Reserved	Reserved	Reserved
x‘0215C’	tag 20	tag 21	tag 20	tag 21	Entry 10	Entry 10	Entry 10	Reserved	Reserved	Reserved
x‘0217C’	tag 22	tag 23	tag 22	tag 23	Entry 11	Entry 11	Entry 11	Reserved	Reserved	Reserved
x‘0219C’	tag 24	tag 25	tag 24	tag 25	Entry 12	Entry 12	Entry 12	Reserved	Reserved	Reserved
x‘021BC’	tag 26	tag 27	tag 26	tag 27	Entry 13	Entry 13	Entry 13	Reserved	Reserved	Reserved
x‘021DC’	tag 28	tag 29	tag 28	tag 29	Entry 14	Entry 14	Entry 14	Reserved	Reserved	Reserved
x‘021FC’	tag 30	tag 31	tag 30	tag 31	Entry 15	Entry 15	Entry 15	Reserved	Reserved	Reserved

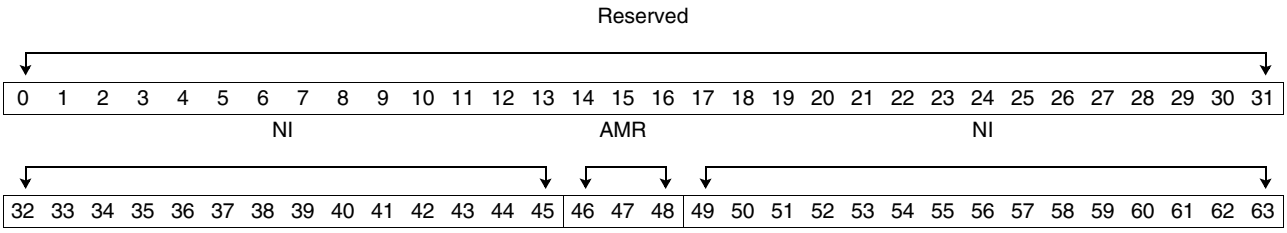
See the *Cell Broadband Engine Architecture* document for more information about this register.

3.3.2.2 SPU Local Store Limit Register (SPU_LSLR)

Access to this register is privileged. The SPU_LSLR Register provides privileged software with a means to artificially limit the size of local store available to an application. This provides for backwards compatibility with applications sensitive to the size of local store. The value written to this register limits the size of the local store. If an application performs a quadword load or store from the SPU beyond the range of the SPU_LSLR, the operation occurs at the resulting wrapped address. The default value initialized at POR is a local-store address limit of 256 KB. This register can only be updated while the SPU is stopped, as indicated in SPU_Status[31].

Note: Reads and writes can be to the full 64 bits or to only the lower-word bits [32:63].

Register Short Name	SPU_LSLR	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘064058’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	x‘00000000_0003FFFF’	**Value During POR Set By**	Scan initialization during POR Bits 49-63 hardwired
Specification Type	CBEA architected register	**Unit**	MFC

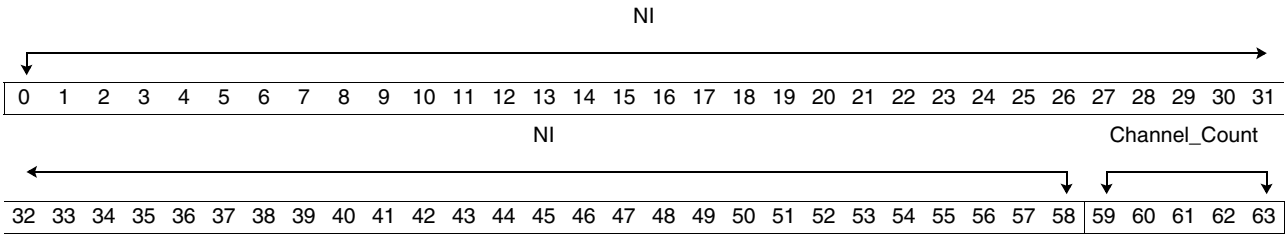


Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32:45	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
46:48	AMR	Address Mask Register (AMR) Local Store Limit Range. 111 256 KB local store access limit 011 128 KB local store access limit 001 64 KB local store access limit 000 32 KB local store access limit Other combinations are invalid.
49:63	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. Reads always return ones.

3.3.2.3 SPU Channel Count Register (SPU_ChnlCnt)

The SPU_ChnlCnt Register is used to read or initialize the count associated with the channel selected by the *SPU Channel Index Register (SPU_ChnlIndex)*.

Register Short Name	SPU_ChnlCnt	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘064068’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	N/A	**Value During POR Set By**	N/A
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:31	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
32:58	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
59:63	Channel_Count	The 5-bit channel count used by software. Typically set to 0, 1, or 16. See the **Programming Note**.

Programming Note: It is recommended that channel counts be initialized by privileged software before a new context is started in a SPU, as shown in the following table:

Channel Names	Recommended Initialization for the Channel Count Setting
SPU_RdEventStat channel SPU_RdSigNotify1 channel SPU_RdSigNotify2 channel MFC_RdTagStat channel MFC_RdListStallStat channel MFC_RdAtomicStat channel SPU_RdInMbox channel	Initialize to 0
MFC_WrMSSyncReq channel MFC_WrTagUpdate channel MFC_WrOutMbox channel MFC_WrOutIntrMbox channel	Initialize to 1
MFC_Cmd channel	Initialize to 16

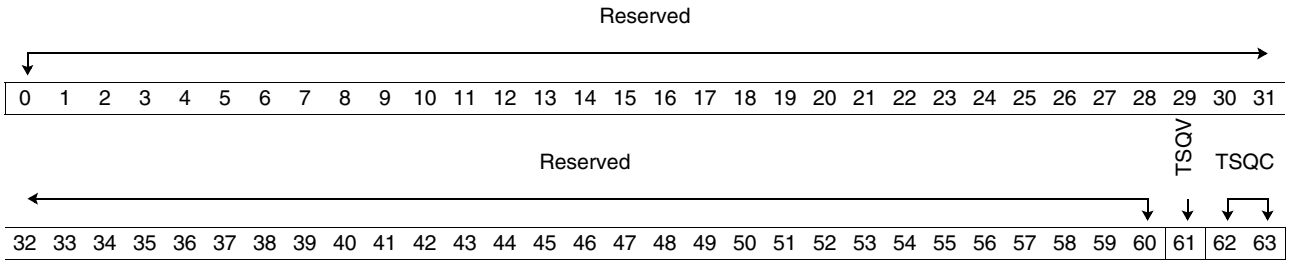
3.3.3 Context Save and Restore Registers (Implementation-Specific)

There are four implementation-specific context save and restore registers.

3.3.3.1 Context Save and Restore for SPU MFC Commands Register (MFC_CSR_TSQ)

This register stores context data for the state of the Tag Status Query logic in the DMAC unit of the MFC.

Register Short Name	MFC_CSR_TSQ	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘065008’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC

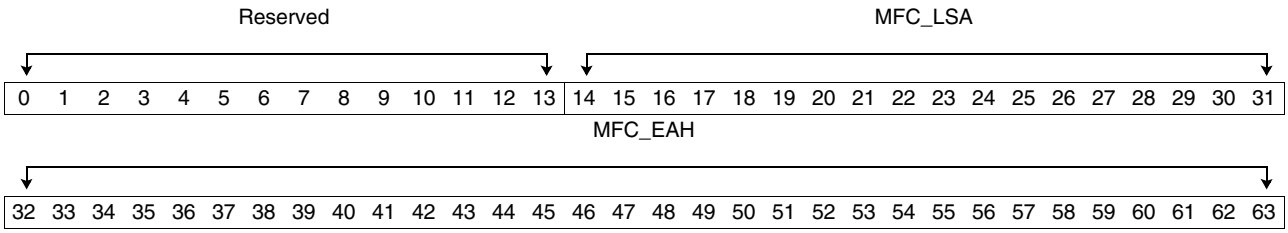


Bits	Field Name	Description
0:60	Reserved	Bits are not implemented; all bits read back zero.
61	TSQV	Tag Status Query Valid TSQV = 1 and TSQC = x‘01’ Waiting for ANY condition to be met TSQV = 1 and TSQC = x‘10’ Waiting for ALL conditions to be met TSQV = 0 No pending query **Note:** A valid query with immediate condition ‘00’ should never be saved as context because the query is always completed before a context save.
62:63	TSQC	Tag Status Query Condition

3.3.3.2 Context Save and Restore for SPU MFC Commands Register (MFC_CSR_CMD1)

MFC_CSR_CMD1 and MFC_CSR_CMD2 are used for context save and restore operations. MFC_CSR_CMD1 stores the data for the MFC_LSA channel and the MFC_EAH channel.

Register Short Name	MFC_CSR_CMD1	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘065010’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC

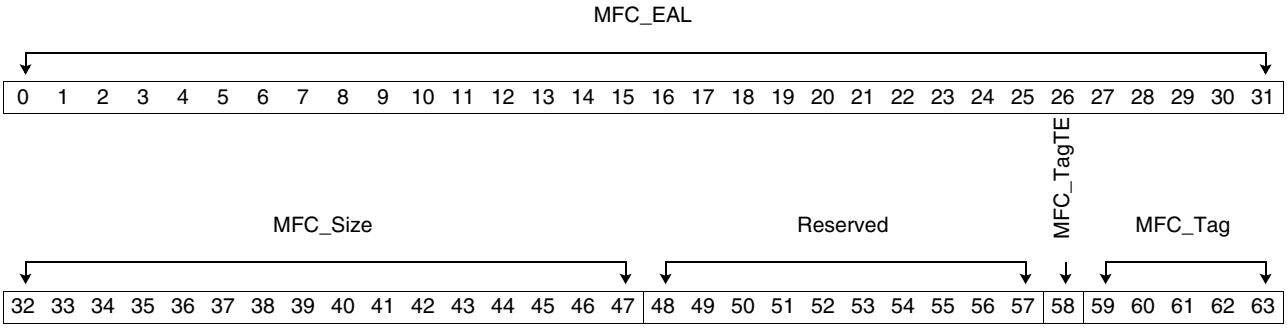


Bits	Field Name	Description
0:13	Reserved	Bits are not implemented; all bits read back zero.
14:31	MFC_LSA	MFC local store address.
32:63	MFC_EAH	MFC effective address high.

3.3.3.3 Context Save and Restore for SPU MFC Commands Register (MFC_CSR_CMD2)

MFC_CSR_CMD1 and MFC_CSR_CMD2 are used for context save and restore operations. MFC_CSR_CMD2 stores the data for the MFC_EAL channel, the MFC_Size channel, and the MFC_TagID channel.

Register Short Name	MFC_CSR_CMD2	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘065018’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



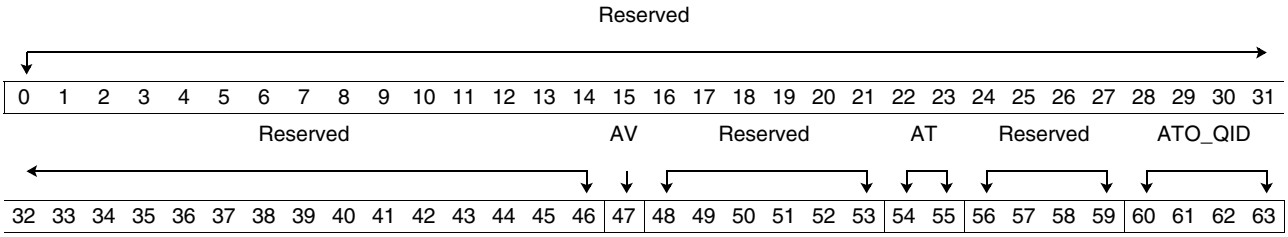
Bits	Field Name	Description
0:31	MFC_EAL	MFC effective address low.
32:47	MFC_Size	MFC transfer size.
48:57	Reserved	Bits are not implemented; all bits read back zero.
58	MFC_TagTE	MFC Tag data [0:26] is nonzero. A tag error causes an MFC command error interrupt.
59:63	MFC_Tag	MFC Tag ID data [27:31].

3.3.3.4 Context Save and Restore for SPU Atomic Immediate Command (MFC_CSR_ATO)

This register stores the state of the Atomic Immediate command in the MFC SPUQ.

The Atomic Immediate command is only valid for context saves when the command has not yet gone to the atomic unit for processing. Any atomic command that has gone to the atomic unit is completed prior to context save and is invalidated.

Register Short Name	MFC_CSR_ATO	**Privilege Type**	Privilege 2
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘065020’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Privilege 2
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC



Bits	Field Name	Description
0:46	Reserved	Bits are not implemented; all bits read back zero.
47	AV	Atomic valid.
48:53	Reserved	Bits are not implemented; all bits read back zero.
54:55	AT	Atomic type. 00 No atomic immediate command 01 **getllar** 10 **putllc** 11 **putlluc**
56:59	Reserved	Bits are not implemented; all bits read back zero.
60:63	ATO_QID	Entry ID of the MFC SPU command queue holding the atomic command.

3.4 SPE Problem State Memory Map Registers

This section lists the registers included in the problem state memory map.

3.4.1 MFC Multisource Synchronization Register

3.4.1.1 MFC Multisource Synchronization Register (MFC_MSSync)

This register is the interface to the MFC multisource synchronization facility. See the *Cell Broadband Engine Architecture* document for more information about this facility. Writing any value to this register requests a synchronization. At the time of the write, the MFC starts to track all outstanding transfers targeting the corresponding SPE. When read, this register returns the current status of the last request. A value of zero is returned when all transfers targeting the SPE and received before the last write of the MFC_MSSync Register are complete.

Register Short Name	MFC_MSSync	**Privilege Type**	Problem State
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘040000’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MFC

Bits	Field Name	Description
0:62	Reserved	Bits are not implemented; all bits read back zero.
63	P	Pending 0 All transfers before writing the MFC_MSSync Register are complete. 1 All transfers before writing the MFC_MSSync Register are not complete.

3.4.2 MFC Command Parameter Registers

These registers describe the MFC Command Parameter channels.

3.4.2.1 MFC Local Store Address Register (MFC_LSA)

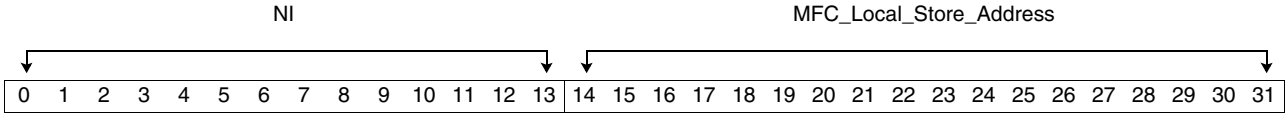
The MFC local store address parameter stored in the MFC_LSA Register is used to supply the SPU local store address associated with a DMA command to be queued. This address is used as the source or destination of the DMA transfer as it is defined in the DMA command.

The contents of the MFC local store address parameter are not persistent and must be written for each DMA command-enqueued sequence.

The validity of this parameter is checked asynchronous to the instruction stream. If the address is unaligned, MFC command queue processing is suspended, and an MFC DMA alignment exception is generated.

Note: Providing a local store address above the implemented range of local store causes the local store address to wrap around to a valid address, but an exception is not posted if this occurs.

Register Short Name	MFC_LSA	**Privilege Type**	Problem State
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘043004’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



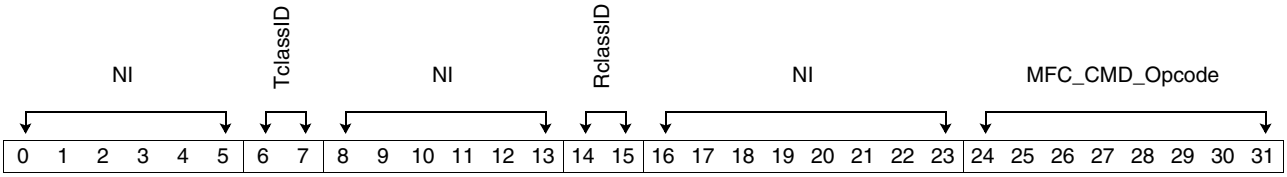
Bits	Field Name	Description
0:13	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
14:31	MFC_Local_Store_Address	MFC_LSA[25:31] must always be aligned to a transfer-size boundary and the 4 least significant bits of the local-store address must match the 4 least significant bits of the effective address.

3.4.2.2 MFC Class ID and Command Opcode Register (MFC_ClassID_CMD)

This register is documented in the *Cell Broadband Engine Architecture* as two registers, MFC_ClassID and MFC_CMD. The MFC class ID parameter is used to specify the replacement class ID and the transfer class ID for each MFC command. The transfer class ID (TclassID) bit field is used to identify access to storage with differing characteristics.

The write-only MFC_ClassID_CMD Register is related to the MFC_CMDStatus Register, which is read only. See *MFC Command Status Register (MFC_CMDStatus)* on page 114 for more information.

Register Short Name	MFC_ClassID_CMD	**Privilege Type**	Problem State
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘043014’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:5	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
6:7	TclassID	Transfer class identifier
8:13	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
14:15	RclassID	Replacement class identifier
16:23	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
24:31	MFC_CMD_Opcode	MFC command opcode

Programming Note: The total number of queue slots is implementation-specific and varies between implementations. For portability of an application, the enqueue sequence for MFC commands and the method to determine the number of queue slots available should be provided as a macro.

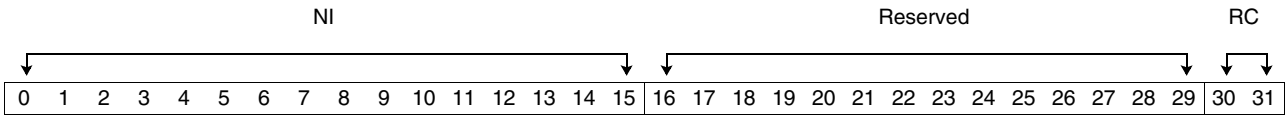
3.4.2.3 MFC Command Status Register (MFC_CMDStatus)

The MFC_CMDStatus Register contains the return code from the last attempt to enqueue an MFC command. The return code is read from the same location as the command, tag, and size.

Note: The MFC_CMDStatus Register is a read-only, 32-bit register; the most significant 16 bits are implementation specific. The MFC command return code in the least significant bits returns the command status when read.

The MFC_CMDStatus Register, which is read only, is related to the MFC_ClassID_CMD Register, which is write only. See *MFC Class ID and Command Opcode Register (MFC_ClassID_CMD)* on page 113 for more information.

Register Short Name	MFC_CMDStatus	**Privilege Type**	Problem State
Access Type	MMIO Read Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘043014’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:5	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
6:7	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. Bits return the last value written to this address (the TClass ID field, bits [6:7] of the MFC_ClassID_CMD Register).
8:13	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
14:15	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. Bits return the last value written to this address (the RClass ID field, bits [14:15] of the MFC_ClassID_CMD Register).
16:29	Reserved	Bits are not implemented; all bits read back zero.
30:31	RC	MFC command return code 00 Command enqueue successful. 01 Command enqueue failed due to sequencing error. 10 Command enqueue failed due to insufficient space in the command queue (the free space in the command queue is zero). 11 Command enqueue failed due to sequencing error, and free space in the command queue is zero.

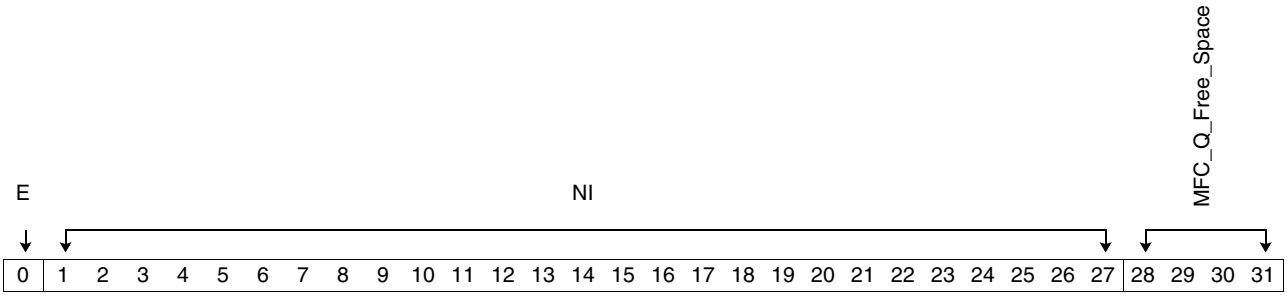
3.4.3 MFC Proxy Command Queue Control Registers

The registers in this section are used to control the MFC proxy command queue.

3.4.3.1 MFC Queue Status Register (MFC_QStatus)

The MFC_QStatus Register contains the current status of the MFC command queue. MFC_QStatus[0] indicates whether the MFC proxy command queue is empty or contains valid commands that are not yet complete. The least significant 4 bits of this register return the number of entries available in the MFC proxy command queue. A value of zero in this field indicates that the queue is full.

Register Short Name	MFC_QStatus	**Privilege Type**	Problem State
Access Type	MMIO Read Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘043104’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC

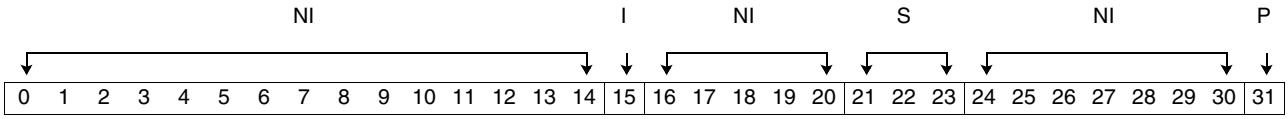


Bits	Field Name	Description
0	E	MFC proxy command queue empty. All MFC operations are complete. 0 MFC proxy command queue contains commands 1 MFC proxy command queue does not contain commands
1:27	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
28:31	MFC_Q_Free_Space	MFC queue free space This field contains the number of queue entries available. Software can use this field to set a loop count for the number of MFC commands to enqueue. Software must not assume a command is enqueued based on the free space. Other conditions may cause the command issue sequence to fail. See the *Cell Broadband Engine Architecture* document for more information.

3.4.3.2 SPU Mailbox Status Register (SPU_Mbox_Stat)

The SPU_Mbox_Stat Register contains the current state of the mailbox queues between the SPU and other processors and devices.

Register Short Name	SPU_Mbox_Stat	**Privilege Type**	Problem State
Access Type	MMIO Read Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘044014’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	x‘00000400’	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC

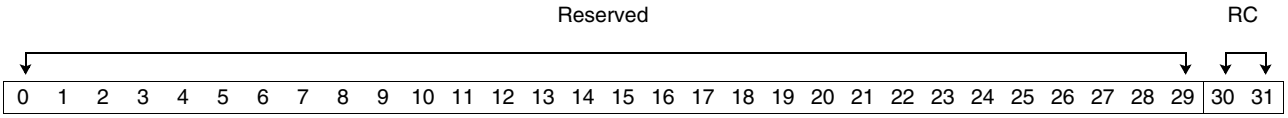


Bits	Field Name	Description
0:14	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
15	I	SPU Outbound Interrupt Mailbox status is equal to 1 minus the SPU_WrOutIntrMbox channel count. This status bit is set when the SPU Outbound Interrupt Mailbox is written by the SPU. It is reset when the PPU reads the SPU Outbound Interrupt Mailbox. 0 SPU Outbound Interrupt Mailbox is empty. 1 SPU Outbound Interrupt Mailbox contains a new value.
16:20	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
21:23	S	SPU Inbound Mailbox status is equal to 4 minus the SPU_RdInMbox channel count. This status bit decrements when the SPU Inbound Mailbox is written by the PPU. It increments when the SPU reads the SPU Inbound Mailbox. 000 SPU Inbound Mailbox is full. 001 SPU Inbound Mailbox has one location available to load. 010 SPU Inbound Mailbox has two locations available to load. 011 SPU Inbound Mailbox has three locations available to load. 100 SPU Inbound Mailbox has four locations available to load. ‘101’ through ‘111’ are unused.
24:30	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
31	P	SPU Outbound Mailbox status is equal to 1 minus the SPU_WrOutMbox channel count. This status bit is set when the SPU Outbound Mailbox is written by the SPU. It is reset when the PPU reads the SPU Outbound Mailbox. 0 SPU Outbound Mailbox is empty. 1 SPU Outbound Mailbox contains a new value.

3.4.3.3 SPU Run Control Register (SPU_RunCntl)

The SPU_RunCntl Register is used to start and stop the execution of instructions in the SPU. The SPU can dynamically change the state of the run bit. The current status of the SPU run state is available in the *SPU Status Register (SPU_Status)*. When this register is read, it returns the last data written for the last valid write.

Register Short Name	SPU_RunCntl	**Privilege Type**	Problem State
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘04401C’ + (x‘80000’ x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:29	Reserved	Bits are not implemented; all bits read back zero.
30:31	RC	SPU run control 00 SPU stop request. No instructions are issued. 01 SPU run request. Instruction is issued if not stalled on condition. 10, 11 Reserved. The current status of the SPU run state is available in the SPU_Status Register.

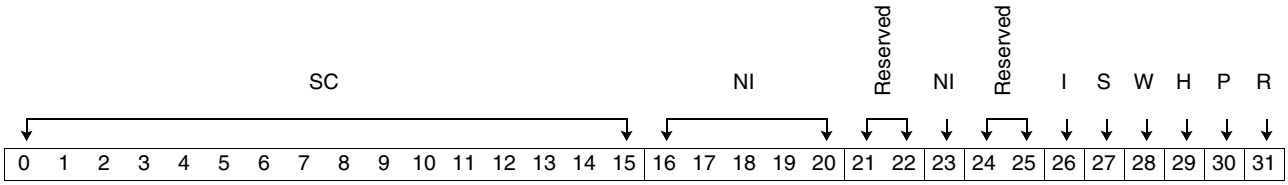
Programming Note: After SPU_RunCntl[30:31], the run control bit, is set to stop the SPU, the SPU is not stopped until SPU_Status[31], the run status bit, reads ‘0’. See the *SPU Status Register (SPU_Status)* for more information. A write of ‘01’ while the SPU is idle causes the SPU to restart from the Program Counter at which it stopped. The SPU ignores writes of ‘01’ unless the SPU is idle, as indicated when SPU_Status[31], the R bit, reads ‘0’.

3.4.3.4 SPU Status Register (SPU_Status)

Reading this register provides a snapshot of the current SPU status. The status read can be dynamically changing if the SPU is running. If the SPU is stopped or halted, the status remains static until it is changed by software. If the SPU was stopped under PPU control at the same time that the SPU was waiting on a blocked channel, the SPU Wait Status bit is set in conjunction with the SPU Run Status bit. Multiple state bits ([26:27] and [29:30]) may be set based on the program design.

When SPU_Status[31] changes to '1', SPU_Status[26:27] and SPU_Status[29:30] are reset to '0'.

Register Short Name	SPU_Status	**Privilege Type**	Problem State
Access Type	MMIO Read Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x'044024' + (x'80000' x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:15	SC	If SPU_Status[30], the P bit, which is the stop and signal indication, is set to '1', this field provides a copy of bits [18-31] of the SPU stop and signal instruction that caused the SPU stop. Bits [0:1] of this field are always set to '0'. If SPU_Status[30], the P bit, is not set, data in this field is not valid. A stop and signal with dependencies (**stopd**) instruction, used for debugging, always sets each of bits [2:15] to '1'.
16:20	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
21:22	Reserved	Reserved. Software should ignore value read.
23	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
24	Reserved	Reserved. Software should ignore value read.
25	Reserved	Bit is not implemented; bit reads back zero.
26	I	Invalid Instruction Detected. The SPU does not stop precisely, and the *SPU Next Program Counter Register (SPU_NPC)* might not indicate the instruction after the illegal instruction. 0 No illegal opcodes have been issued. 1 An illegal opcode has been issued, and the SPU has been halted. An SPU error interrupt is also generated in INT_Class0[61] if enabled in SPU_ERR_Mask[63].
27	S	SPU Single-Step Status. The *SPU Next Program Counter Register (SPU_NPC)* points to the next instruction after the instructions committed for the single-step operation. 0 SPU not stopped due to single-step mode. 1 SPU stopped after one completed instruction in single-issue mode or a pair of instructions in dual-issue mode.

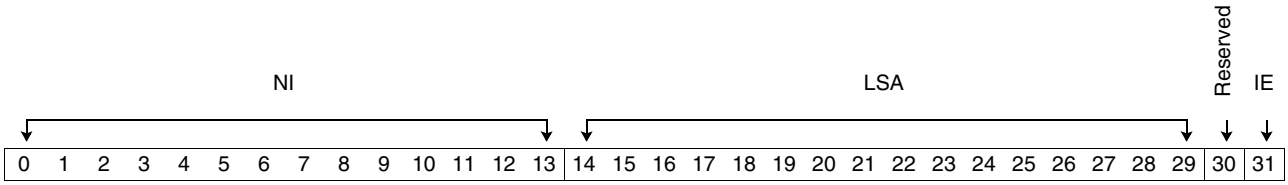
Bits	Field Name	Description
28	W	SPU Wait Status. 0 SPU is not waiting on a blocked channel. 1 SPU is waiting on a blocked channel.
29	H	SPU Halt Status. The SPU does not stop precisely, and the *SPU Next Program Counter Register (SPU_NPC)* might not indicate the instruction after the halt instruction. SPU_Status[29] is not set if SPU stops due to single step. 0 SPU is not halted due to a **halt** instruction. 1 SPU is halted due to a **halt** instruction.
30	P	SPU Program Stop and Signal Status. The *SPU Next Program Counter Register (SPU_NPC)* points to the instruction after the committed stop instruction. 0 SPU is not stopped due to a **Stop and Signal** instruction. 1 SPU is stopped due to a **Stop and Signal** instruction.
31	R	SPU Run Status. 0 SPU stopped (idle). 1 SPU running.

3.4.3.5 SPU Next Program Counter Register (SPU_NPC)

The Local Store Address (LSA) value read from this register is limited by the value of SPU_LSLR[46:48], the AMR field.

A read from this register is only valid when the SPU is stopped (SPU_Status[31] is set to '0'). Otherwise, it returns meaningless data (all zeros). Values written to this register while the SPU is running have no effect on the operation of the SPU and are ignored. SPU Interrupts can only be enabled in this register prior to starting the SPU and cannot be enabled while the SPU is running. That is, the internal enable bit gets its value from this register when the SPU starts running. When the SPU is stopped, the internal enable bit is loaded to this register and may have been changed during program execution by an indirect branch.

Register Short Name	SPU_NPC	**Privilege Type**	Problem State
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x'044034' + (x'80000' x *n*)	**Memory Map Area**	SPE Problem State
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	CBEA architected register	**Unit**	MFC



Bits	Field Name	Description
0:13	NI	Bits are defined in the *Cell Broadband Engine Architecture* but are not implemented in the CBE. All bits read back zero.
14:29	LSA	Word-aligned local-store address (LSA).
30	Reserved	Bit is not implemented; bit reads back zero.
31	IE	Interrupt enable state. 0 SPU interrupts disabled at start. 1 SPU interrupts enabled at start.

Programming Note: SPU_NPC is not valid for an illegal instruction.

4. BEI I/O Command (IOC) MMIO Registers

This section describes the Cell Broadband Engine Interface (BEI) IOC memory-mapped I/O (MMIO) registers. *Table 4-1* shows the BEI IOC MMIO memory map and lists the BEI IOC registers. The BEI IOC register space starts at x‘511C00’ and ends at x‘511FFF’. Offsets are from the start of the BEI IOC register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

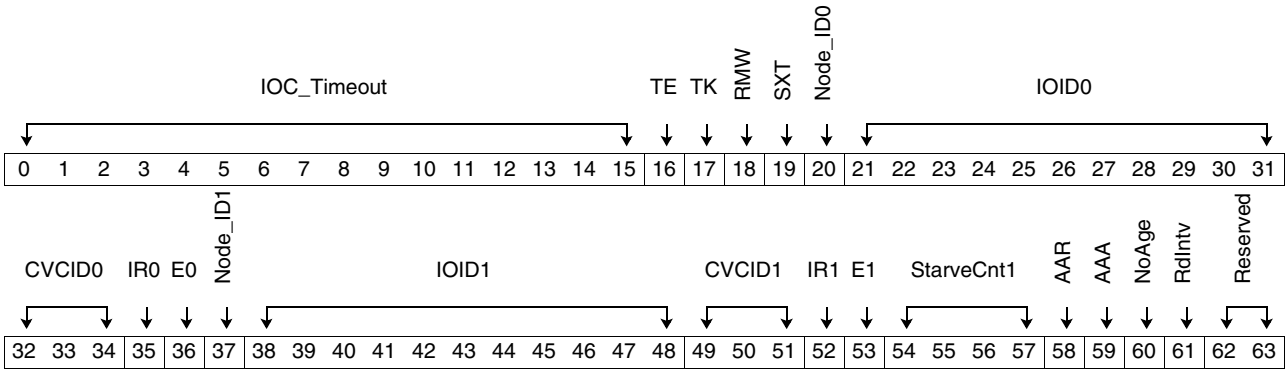
Table 4-1. BEI IOC MMIO Memory Map

Hexadecimal Offset (x‘511*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x‘C00’	*IOC IOCmd Configuration Register (IOC_IOCmd_Cfg)*	64	R/W	*Section 4.1* on page 122
x‘C08’	*IOC Memory Base Address Register (IOC_MemBaseAddr)*	64	R/W	*Section 4.2* on page 124
x‘C10’	*IOC Base Address Register 0 (IOC_BaseAddr0)*	64	R/W	*Section 4.3* on page 125
x‘C18’	*IOC Base Address Mask Register 0 (IOC_BaseAddrMask0)*	64	R/W	*Section 4.4* on page 126
x‘C20’	*IOC Base Address Register 1 (IOC_BaseAddr1)*	64	R/W	*Section 4.5* on page 127
x‘C28’	*IOC Base Address Mask Register 1 (IOC_BaseAddrMask1)*	64	R/W	*Section 4.6* on page 128
x‘C30’ – x‘C50’	Reserved			
x‘C58’	*IOC SRAM Parity Error Capture Register (IOC_SRAM_ParityErrCap)*	64	R	*Section 4.7* on page 129
x‘C60’	*IOC IOIF0 Queue Threshold Register (IOC_IOIF0_QueThshld)*	64	R/W	*Section 4.8* on page 130
x‘C68’	*IOC IOIF1 Queue Threshold Register (IOC_IOIF1_QueThshld)*	64	R/W	*Section 4.9* on page 131
x‘C70’ – x‘FFF’	Reserved			

4.1 IOC IOCmd Configuration Register (IOC_IOCmd_Cfg)

This register configures basic settings for the IOC.

Register Short Name	IOC_IOCmd_Cfg	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C00’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BEI



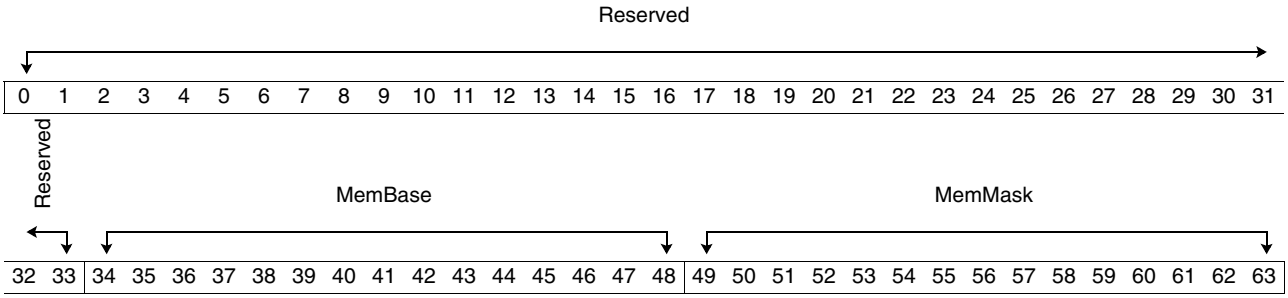
Bits	Field Name	Description
0:15	IOC_Timeout	Timeout time for commands in InCmd or OutCmd. Only the leftmost bit is used. x‘0000’ Disable timeout x‘0001’ 2^{16} / (NClk/2) x‘0002’ 2^{17} / (NClk/2) x‘0004’ 2^{18} / (NClk/2) x‘0008’ 2^{19} / (NClk/2) x‘0010’ 2^{20} / (NClk/2) ... x‘8000’ 2^{31} / (NClk/2)
16	TE	Enable I/O address translation.
17	TK	Enable tokens.
18	RMW	Read-modify-write. 0 Get one token for all stores. 1 Get two tokens for stores less than 128 bytes in length.
19	SXT	16 bytes. 0 Get one token for all stores. 1 Get two tokens for stores less than 16 bytes in length.
20	Node_ID0	Node ID bit for CBE to use in IOTtags for commands being sent on I/O interface 0 (IOIF0). 0 Use node ID 0. 1 Use node ID 1.
21:31	IOID0	I/O ID for CBE to use in commands being sent on IOIF0.
32:34	CVCID0	CVC ID for CBE to use in commands being sent on IOIF0.

Bits	Field Name	Description
35	IR0	Enable broadcast interrupt reissue commands seen on the Element Interconnect Bus (EIB) onto IOIF0.
36	E0	Enable broadcast of enforce in-order execution of I/O transaction (**eieio**) commands seen on the EIB onto IOIF0.
37	Node_ID1	Node ID bit for CBE to use in IOTtags for commands being sent on I/O Interface 1 (IOIF1). 0 Use node ID 0. 1 Use node ID 1.
38:48	IOID1	I/O ID for CBE to use in commands being sent on IOIF1.
49:51	CVCID1	CVC ID for CBE to use in commands being sent on IOIF1.
52	IR1	Enable broadcast interrupt reissue commands seen on the EIB onto IOIF1.
53	E1	Enable broadcast of **eieio** commands seen on the EIB onto IOIF1.
54:57	StarveCnt1	Starvation count for IOIF1. IOIF0 has priority over IOIF1 in using the translation logic unless IOIF1 has a command waiting for this field's value of cycles or more.
58	AAR	Token setup: allow all requests. 0 Requests for IOIFO tokens are not made by a command until it has obtained all its nonIOIF0 tokens. 1 Allow requests for IOIF0 tokens to be made as soon as possible.
59	AAA	Token setup: allow all assignments. 0 IOIF0 tokens are only assigned to commands that have obtained all their nonIOIF0 tokens. 1 Allow assignments of IOIF0 tokens to be made as soon as possible.
60	NoAge	Token setup: disable aging. 0 Older commands are favored over newer commands within a resource allocation group (RAG). 1 Disable aging.
61	RdIntv	Read intervention. 0 Enable putting a '0' in the N bit on a read of 128 bytes. 1 Enable putting a '1' in the N bit on a read of 128 bytes.
62:63	Reserved	Bits are not implemented; all bits read back zero.

4.2 IOC Memory Base Address Register (IOC_MemBaseAddr)

This register configures the memory base address and mask used by the IOC to determine the real addresses that are mapped to memory. For incoming IOIF reads and writes, the IOC determines whether the associated real address accesses memory. If memory is accessed, the IOC requests the appropriate memory bank token.

Register Short Name	IOC_MemBaseAddr	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C08’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BEI

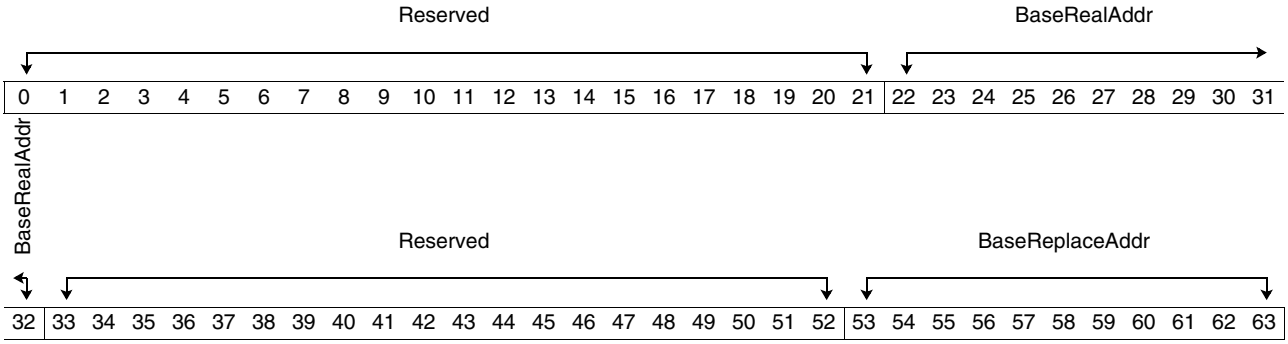


Bits	Field Name	Description
0:33	Reserved	Bits are not implemented; all bits read back zero.
34:48	MemBase	Memory Base Real Address. Assumed to be aligned on a 128 MB boundary. This field represents bits [0:14] of the real memory address.
49:63	MemMask	Memory Mask. Only memory sizes that are powers of two are allowed.

4.3 IOC Base Address Register 0 (IOC_BaseAddr0)

This register configures the address and replacement address for I/O adapter 0.

Register Short Name	IOC_BaseAddr0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C10’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	BEI

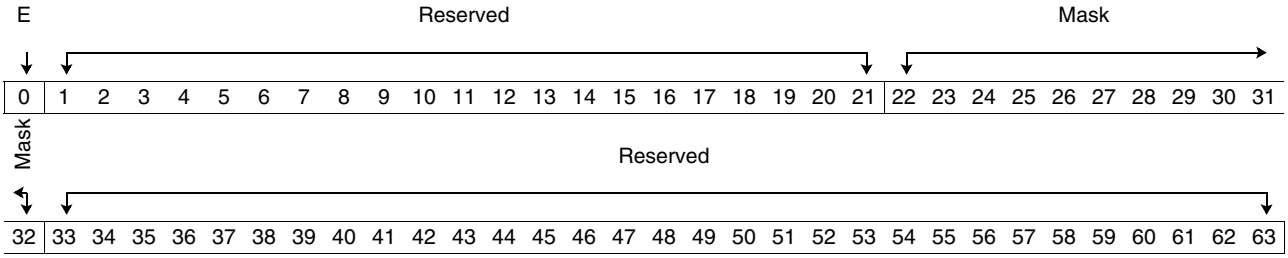


Bits	Field Name	Description
0:21	Reserved	Bits are not implemented; all bits read back zero.
22:32	BaseRealAddr	Base real address. Specifies a base real address for accesses that are routed to IOIF0.
33:52	Reserved	Bits are not implemented; all bits read back zero.
53:63	BaseReplaceAddr	Base replacement address. Specifies the value to replace part or all of the base real address that is passed to IOIF0.

4.4 IOC Base Address Mask Register 0 (IOC_BaseAddrMask0)

This register configures the address mask for I/O adapter 0.

Register Short Name	IOC_BaseAddrMask0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C18’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	BEI

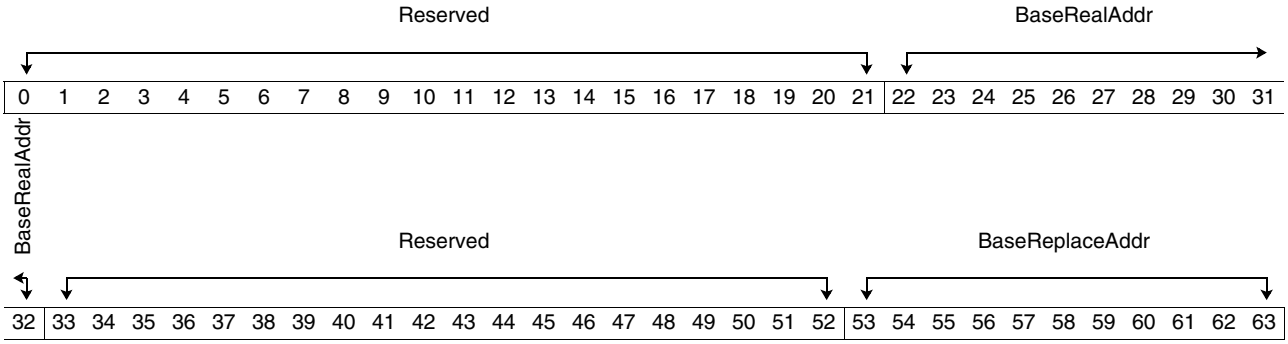


Bits	Field Name	Description
0	E	When set, enables passing accesses to IOIF0.
1:21	Reserved	Bits are not implemented; all bits read back zero.
22:32	Mask	Mask. Specifies the size of the address range that is routed to IOIF0.
33:63	Reserved	Bits are not implemented; all bits read back zero.

4.5 IOC Base Address Register 1 (IOC_BaseAddr1)

This register configures the address and replacement address for I/O adapter 1.

Register Short Name	IOC_BaseAddr1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C20’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	BEI

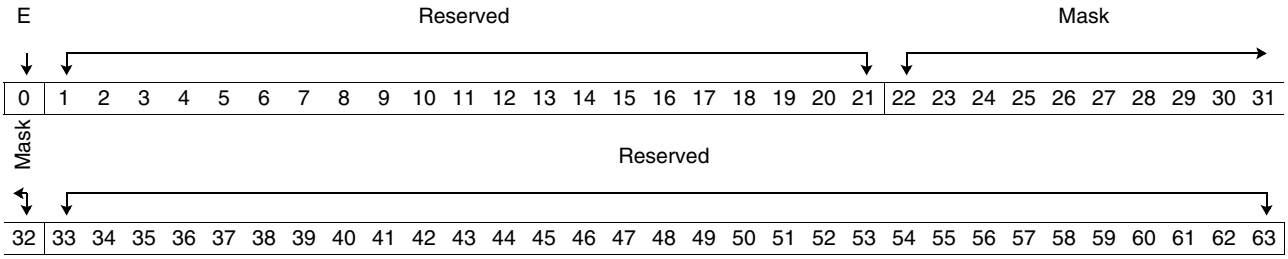


Bits	Field Name	Description
0:21	Reserved	Bits are not implemented; all bits read back zero.
22:32	BaseRealAddr	Base real address. Specifies a base real address for accesses that are routed to IOIF1
33:52	Reserved	Bits are not implemented; all bits read back zero.
53:63	BaseReplaceAddr	Base replacement address. Specifies the value to replace part or all of the base real address that is passed to IOIF1.

4.6 IOC Base Address Mask Register 1 (IOC_BaseAddrMask1)

This register configures the address mask for I/O adapter 1.

Register Short Name	IOC_BaseAddrMask1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C28’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	BEI

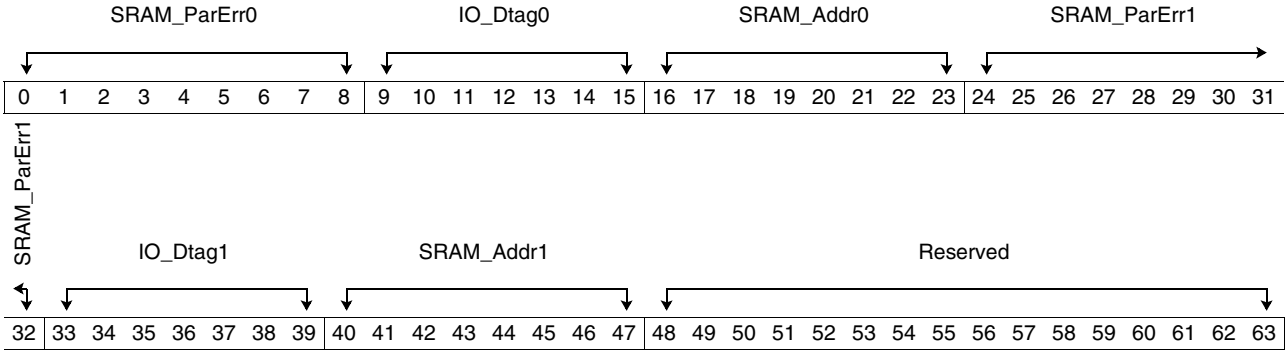


Bits	Field Name	Description
0	E	When set, enables passing accesses to IOIF1.
1:21	Reserved	Bits are not implemented; all bits read back zero.
22:32	Mask	Specifies the size of the address range that is routed to IOIF1.
33:63	Reserved	Bits are not implemented; all bits read back zero.

4.7 IOC SRAM Parity Error Capture Register (IOC_SRAM_ParityErrCap)

This is a read-only register for software that is used to determine the cause of outbound command errors.

Register Short Name	IOC_SRAM_ParityErrCap	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511C58'	**Memory Map Area**	IOC I/O Command
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BEI

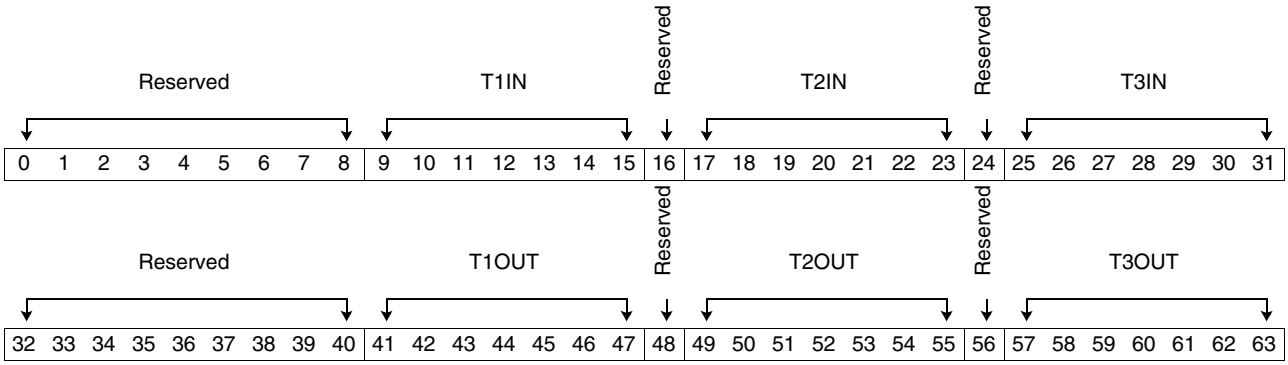


Bits	Field Name	Description
0:8	SRAM_ParErr0	IOIF0 static random access memory (SRAM) parity error. bits [0:7] HWn (where n = 0 through 7) parity error bit [8] Control field parity error
9:15	IO_Dtag0	IOIF0 I/O tag of command with parity error.
16:23	SRAM_Addr0	IOIF0 SRAM address that had parity error.
24:32	SRAM_ParErr1	IOIF1 SRAM parity error. bits [24:31] HWn (where n = 0 through 7) parity error bit [32] Control field parity error
33:39	IO_Dtag1	IOIF1 I/O tag of command with parity error.
40:47	SRAM_Addr1	IOIF1 SRAM address that had parity error.
48:63	Reserved	Bits are not implemented; all bits read back zero.

4.8 IOC IOIF0 Queue Threshold Register (IOC_IOIF0_QueThshld)

This register configures the resource allocation back-pressure threshold values for IOIF0 command queue entries.

Register Short Name	IOC_IOIF0_QueThshld	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C60’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BEI

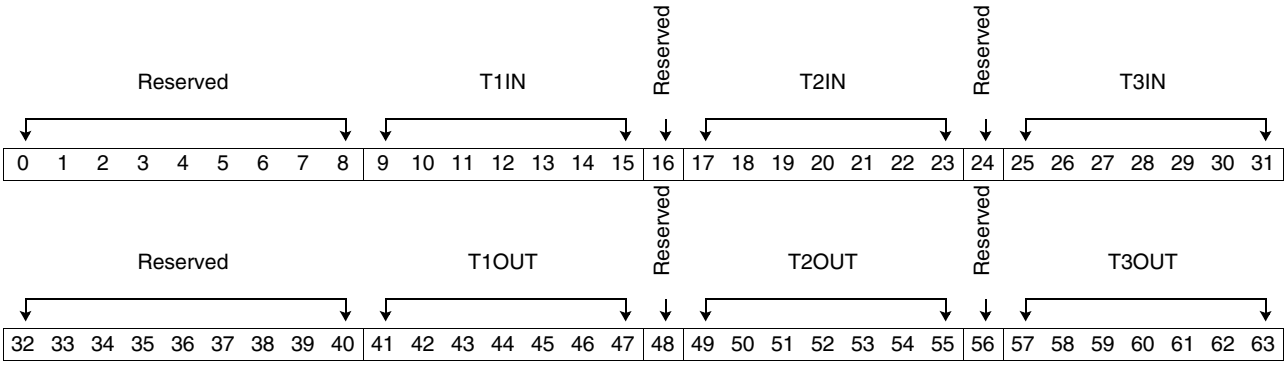


Bits	Field Name	Description
0:8	Reserved	Bits are not implemented; all bits read back zero.
9:15	T1IN	Threshold 1 for IOIF0 command queue entries that require the IOIF0 In bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘7A’ should be loaded.
16	Reserved	Bits are not implemented; all bits read back zero.
17:23	T2IN	Threshold 2 for IOIF0 command queue entries that require the IOIF0 In bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘7A’ should be loaded.
24	Reserved	Bit is not implemented; bit reads back zero.
25:31	T3IN	Threshold 3 for IOIF0 command queue entries that require the IOIF0 In bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘7A’ should be loaded.
32:40	Reserved	Bits are not implemented; all bits read back zero.
41:47	T1OUT	Threshold 1 for IOIF0 command queue entries that require the IOIF0 Out bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘7A’ should be loaded.
48	Reserved	Bits are not implemented; all bits read back zero.
49:55	T2OUT	Threshold 2 for IOIF0 command queue entries that require the IOIF0 Out bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘7A’ should be loaded.
56	Reserved	Bits are not implemented; all bits read back zero.
57:63	T3OUT	Threshold 3 for IOIF0 command queue entries that require the IOIF0 Out bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘7A’ should be loaded.

4.9 IOC IOIF1 Queue Threshold Register (IOC_IOIF1_QueThshld)

This register configures the resource allocation back-pressure threshold values for IOIF1 command queue entries. This register configures the maximum credits allowed for both inbound and outbound commands on IOIF0 and IOIF1.

Register Short Name	IOC_IOIF1_QueThshld	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘511C68’	**Memory Map Area**	IOC I/O Command
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BEI



Bits	Field Name	Description
0:8	Reserved	Bits are not implemented; all bits read back zero.
9:15	T1IN	Threshold 1 for IOIF1 command queue entries that require the IOIF1 In bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘3A’ should be loaded.
16	Reserved	Bits are not implemented; all bits read back zero.
17:23	T2IN	Threshold 2 for IOIF1 command queue entries that require the IOIF1 In bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘3A’ should be loaded.
24	Reserved	Bits are not implemented; all bits read back zero.
25:31	T3IN	Threshold 3 for IOIF1 command queue entries that require the IOIF1 In bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘3A’ should be loaded.
32:40	Reserved	Bits are not implemented; all bits read back zero.
41:47	T1OUT	Threshold 1 for IOIF1 command queue entries that require the IOIF1 Out bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘3A’ should be loaded.
48	Reserved	Bits are not implemented; all bits read back zero.
49:55	T2OUT	Threshold 2 for IOIF1 command queue entries that require the IOIF1 Out bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘3A’ should be loaded.
56	Reserved	Bits are not implemented; all bits read back zero.
57:63	T3OUT	Threshold 3 for IOIF1 command queue entries that require the IOIF1 Out bus for a data transfer. This value is inverted, so that for a threshold of x‘05’, a value of x‘3A’ should be loaded.

5. IOC Address Translation MMIO Registers

This section describes the I/O Command (IOC) address translation memory-mapped I/O (MMIO) registers. *Table 5-1* shows the IOC address translation MMIO memory map and lists the IOC registers. The IOC address translation space starts at x‘510000’ and ends at x‘510FFF’. Offsets are from the start of the IOC address translation register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

Table 5-1. IOC Address Translation MMIO Memory Map

Hexadecimal Offset (x‘510*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
IOC Address Translation				
x‘000’–x‘1F8’	*IOC IOPT Cache Directory Register (IOC_IOPT_CacheDir)* (First way)	64	R/W	*Section 5.1* on page 134
x‘200’–x‘3F8’	*IOC IOPT Cache Directory Register (IOC_IOPT_CacheDir)* (Second way)	64	R/W	*Section 5.1* on page 134
x‘400’–x‘5F8’	*IOC IOPT Cache Directory Register (IOC_IOPT_CacheDir)* (Third way)	64	R/W	*Section 5.1* on page 134
x‘600’–x‘7F8’	*IOC IOPT Cache Directory Register (IOC_IOPT_CacheDir)* (Fourth way)	64	R/W	*Section 5.1* on page 134
x‘800’–x‘8F8’	*IOC IOST Cache Register (IOC_IOST_Cache)*	64	R/W	*Section 5.2* on page 135
x‘900’	*IOC IOST Cache Invalidate Register (IOC_IOST_CacheInvd)*	64	R/W	*Section 5.3* on page 136
x‘908’	*IOC IOPT Cache Invalidate Register (IOC_IOPT_CacheInvd)*	64	R/W	*Section 5.4* on page 137
x‘910’	*IOC IOPT Cache Register (IOC_IOPT_Cache)*	64	R/W	*Section 5.5* on page 138
x‘918’	*IOC IOST Origin Register (IOC_IOST_Origin)*	64	R/W	*Section 5.6* on page 139
x‘920’	*IOC I/O Exception Status Register (IOC_IO_ExcpStat)*	64	R/W	*Section 5.7* on page 140
x‘928’	*IOC I/O Exception Mask Register (IOC_IO_ExcpMask)*	64	R/W	*Section 5.8* on page 141
x‘930’	*IOC Translation Configuration Register (IOC_XlateCfg)*	64	R/W	*Section 5.9* on page 142
x‘931’–x‘FFF’	Reserved			

5.1 IOC IOPT Cache Directory Register (IOC_IOPT_CacheDir)

The I/O page table (IOPT) cache and its directory have 64 sets with 4-way associativity, for a total of 256 entries. When an IOPT cache directory entry is read, the IOPT cache register is automatically loaded from the corresponding IOPT cache entry. When an IOPT cache directory entry is written, the corresponding IOPT cache entry is automatically written from the IOPT cache register.

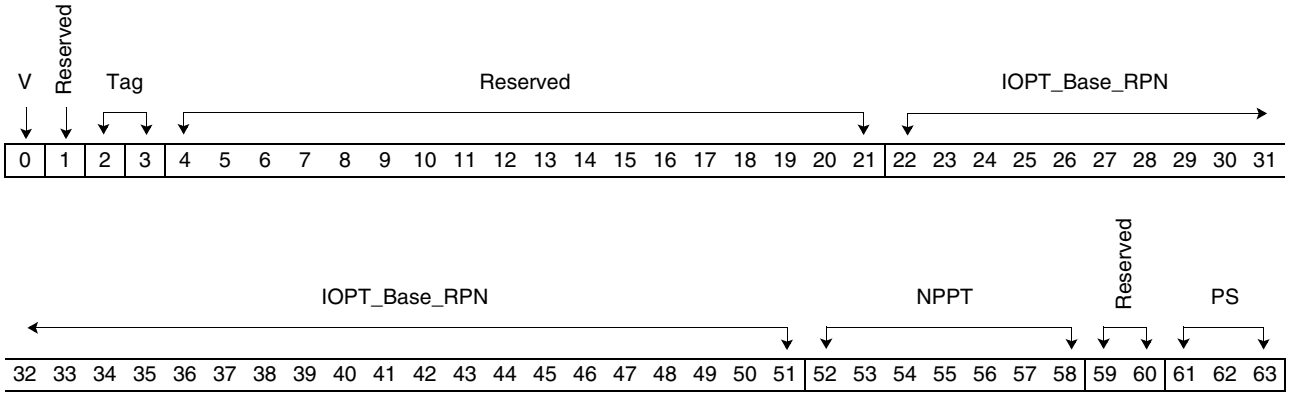
Register Short Name	IOC_IOPT_CacheDir	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘510000’ – x‘5101F8’: First way x‘510200’ – x‘5103F8’: Second way x‘510400’ – x‘5105F8’: Third way x‘510600’ – x‘5107F8’: Fourth way	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC

Bits	Field Name	Description
0:29	Reserved	Bits are not implemented; all bits read back zero.
30	V	Valid. 0 The IOPT cache entry is not valid. 1 The IOPT cache entry is valid.
31:63	Tag	IOPT directory tag.

5.2 IOC IOST Cache Register (IOC_IOST_Cache)

The I/O segment table (IOST) Cache Register is a direct-mapped cache with 32 entries. From an MMIO standpoint, the IOST cache directory is considered to be part of the IOST cache. The fields in the IOST cache register have the same meaning as the corresponding fields in the IOST cache. However, there is no hint bit in the register. Only 30 bits of the IOPT base real page number (RPN) are implemented. Also, the register V and Tag fields are fields in the IOST cache directory, so there are no corresponding fields in the IOST cache.

Register Short Name	IOC_IOST_Cache	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘510800’ – x‘5108F8’	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC

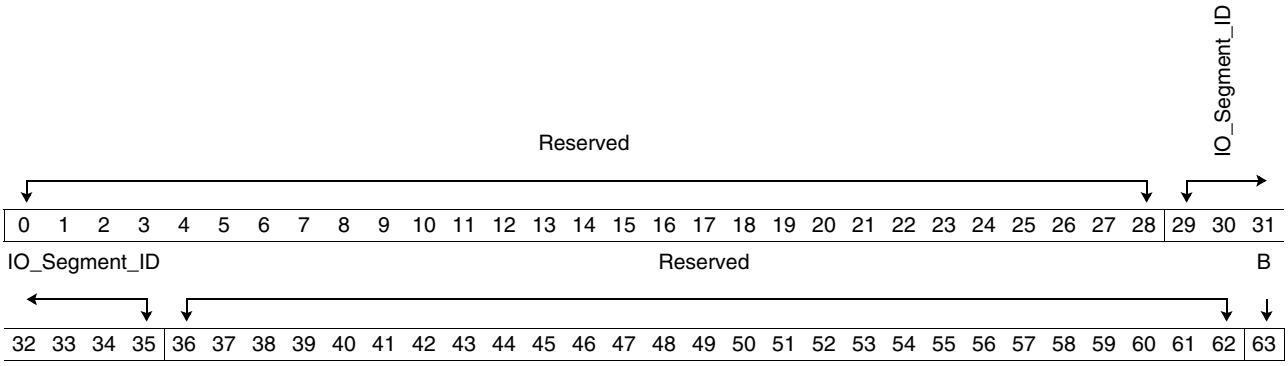


Bits	Field Name	Description
0	V	Valid. 0 IOST cache entry is not valid. 1 IOST cache entry is valid.
1	Reserved	Bits are not implemented; all bits read back zero.
2:3	Tag	IOST directory tag.
4:21	Reserved	Bits are not implemented; all bits read back zero.
22:51	IOPT_Base_RPN	I/O page table base real page number. This is the 4 KB RPN of the first entry in the I/O page table for this I/O segment.
52:58	NPPT	Number of 4 KB pages in the IOPT for this I/O segment minus one. For IOPT format 1, the number of IOPT entries for the segment is equal to 512 times the sum of the NPPT plus 1. The IOPT entries are for the lowest-numbered pages in the segment. For IOPT format 2, the number of IOPT entries for the segment is equal to 256 times the sum of the NPPT plus 1.
59:60	Reserved	Bits are not implemented; all bits read back zero.
61:63	PS	Page size of the RPN in the IOPT entry. That is, this field describes the page size of the pages in this I/O segment. The page size is 4^{PS} KB. Only page sizes of 4 KB, 64 KB, 1 MB, and 16 MB are supported. PS values of 1, 3, 5, and 7 are supported. All four page sizes can be used, regardless of the sizes used by the PowerPC Processor Element (PPE) and Synergistic Processor Element (SPE).

5.3 IOC IOST Cache Invalidate Register (IOC_IOST_CacheInvd)

This register allows software to invalidate the IOST cache.

Register Short Name	IOC_IOST_CacheInvd	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘510900’	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC

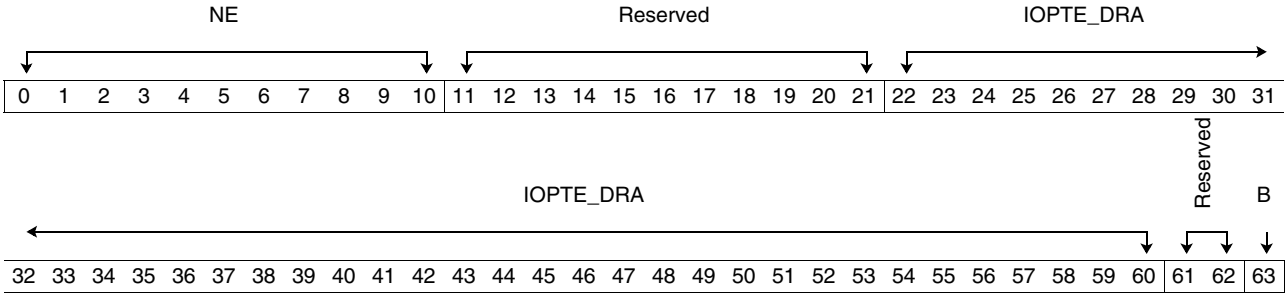


Bits	Field Name	Description
0:28	Reserved	Bits are not implemented; all bits read back zero.
29:35	IO_Segment_ID	This is the I/O segment ID to be invalidated in the IOST cache.
36:62	Reserved	Bits are not implemented; all bits read back zero.
63	B	Busy 0 The IOST segment invalidate operation is complete. 1 The IOST segment invalidate operation is in progress.

5.4 IOC IOPT Cache Invalidate Register (IOC_IOPT_CacheInvd)

This register allows software to invalidate entries in the IOPT cache that correspond to the specified IOPT entries.

Register Short Name	IOC_IOPT_CacheInvd	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'510908'	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	Xlate

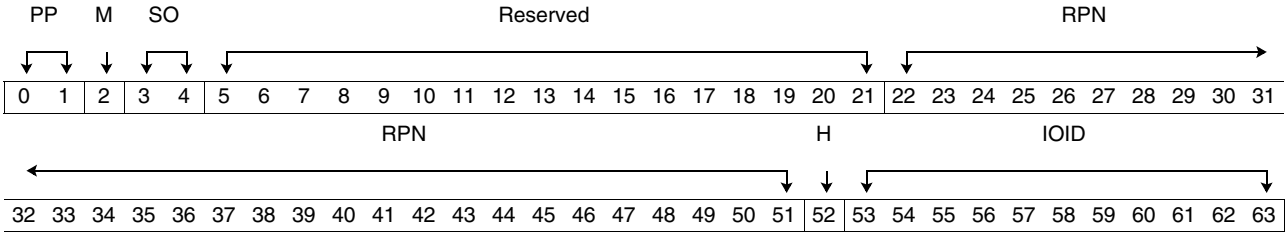


Bits	Field Name	Description
0:10	NE	Number of entries. The value in this field is one less than the number of entries in the IOPT for which IOPT cache entries are to be invalidated.
11:21	Reserved	Bits are not implemented; all bits read back zero.
22:60	IOPTE_DRA	I/O page table entry doubleword real address. (IOPTE_DRA concatenated with '000') is the real address of the first IOPT entry to be invalidated. All entries at or between (IOPTE_DRA concatenated with '000') and ((IOPTE_DRA + NE) concatenated with '000') are invalidated.
61:62	Reserved	Bits are not implemented; all bits read back zero.
63	B	Busy. 0 The IOPT cache invalidate operation is complete. 1 The IOPT cache invalidate operation is in progress.

5.5 IOC IOPT Cache Register (IOC_IOPT_Cache)

The fields in the IOPT cache register have the same meaning as the corresponding fields in the IOPT. However, only 30 bits of RPN are implemented. When the IOPT cache directory entry is written, the contents of the IOPT cache register are moved into the corresponding IOPT cache entry. When an IOPT cache directory entry is read, the IOPT cache register is loaded from the corresponding IOPT cache entry.

Register Short Name	IOC_IOPT_Cache	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘510910’	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC

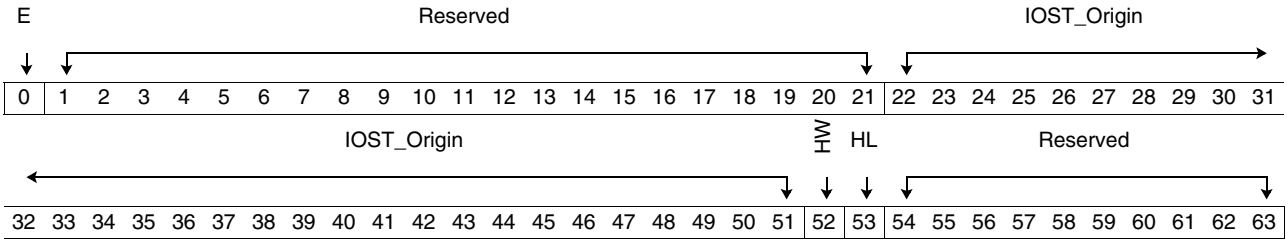


Bits	Field Name	Description
0:1	PP	Page Protection 00 No access 01 Read access 10 Write access 11 Read and write access
2	M	Coherency required. 0 Not required 1 Required
3:4	SO	Storage ordering. 00 Ordered only if the I/O device identifier (IOID), I/O virtual channel (VC), and I/O address match. 01 Ordered only if the IOID, I/O VC, and I/O address match. 10 Ordered only if the IOID, I/O VC, and I/O address match, or IOID and VC match and the previous command is a write. 11 Ordered if the IOID and VC match.
5:21	Reserved	Bits are not implemented; all bits read back zero.
22:51	RPN	Real page number of the translated I/O address.
52	H	Hint enable.
53:63	IOID	I/O device identifier. Only the I/O device specified can have its I/O address translated using this IOPT entry.

5.6 IOC IOST Origin Register (IOC_IOST_Origin)

This register sets up the I/O translation mechanism in the IOC. The IOST Size field is not implemented. When I/O translation is enabled and hardware miss handling is enabled, the IOST is treated as having a fixed size with 128 entries.

Register Short Name	IOC_IOST_Origin	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘510918’	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC



Bits	Field Name	Description
0	E	I/O translation enable. The IOC_IOCmd_Cfg[TE] must be set to the same value as this E bit. 0 Disabled. I/O addresses are treated as real addresses. The I/O address translation unit is disabled to save power. This is the only IOC address translation MMIO register that can be accessed. 1 Enabled. I/O addresses are translated using the IOST and IOPTs.
1:21	Reserved	Bits are not implemented; all bits read back zero.
22:51	IOST_Origin	The real page number (RPN) of the IOST. The IOST must be integrally aligned and must be aligned on a 4 KB page.
52	HW	Enable hardware miss handling for IOST and IOPT caches.
53	HL	Hint lock for IOST and IOPT caches. 0 The hint bit for an IOST or IOPT cache entry is treated as a hint for valid entries. If there are no invalid entries in the applicable congruence class, hardware miss handling should replace an entry whose hint bit is ‘0’ instead of an entry whose hint bit is ‘1’. If there are no invalid entries in the congruence class and all entries in the congruence class have hint bits equal to ‘1’, hardware miss handling replaces a cache entry whose hint bit is ‘1’. 1 The hint bit is treated as a lock. If hardware miss handling is enabled and an IOST or IOPT cache miss occurs, hardware must not replace a cache entry whose hint bit is ‘1’, even if the valid (V) bit in the entry is ‘0’.
54:63	Reserved	Bits are not implemented; all bits read back zero.

5.7 IOC I/O Exception Status Register (IOC_IO_ExcpStat)

When an I/O exception occurs, debug information is captured in this register, and then IOC_IO_ExcpStat[0] is set to '1'. Debug information for an I/O exception is only captured when IOC_IO_ExcpStat[0] is set to '0'. After handling an I/O exception, software should set IOC_IO_ExcpStat[0] to '0' to enable the capturing of information on a subsequent I/O exception.

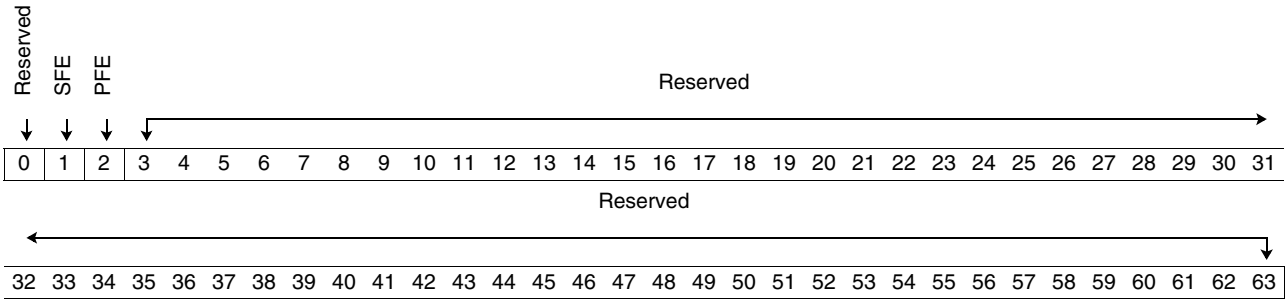
Register Short Name	IOC_IO_ExcpStat	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'510920'	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC

Bits	Field Name	Description
0	V	Valid. Software must set this bit to '0' after handling the exception. 0 Errors that occur when V is set to '0' are not captured. 1 The I/O Exception Status Register contains error status for the first error detected.
1:2	SPF	Segment or page fault exception. The following values assume that bits [1:2] were set to '00' before the exception occurred. 00 No I/O address translation fault occurred. 01 An I/O page fault occurred. 10 Undefined. 11 An I/O segment fault occurred.
3:28	Reserved	Bits are not implemented; all bits read back zero.
29:51	Address	Bits [7:29] of the 42-bit I/O address are used for the access that caused the I/O exception.
52	RW	Read or write type of I/O access. 0 Write 1 Read
53:63	IOID	I/O device identifier.

5.8 IOC I/O Exception Mask Register (IOC_IO_ExcpMask)

This register configures the mask for interrupts due to I/O exceptions. These masks do not affect the setting of the I/O Exception Status Register (IOC_IO_ExcpStat).

Register Short Name	IOC_IO_ExcpMask	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘510928’	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC

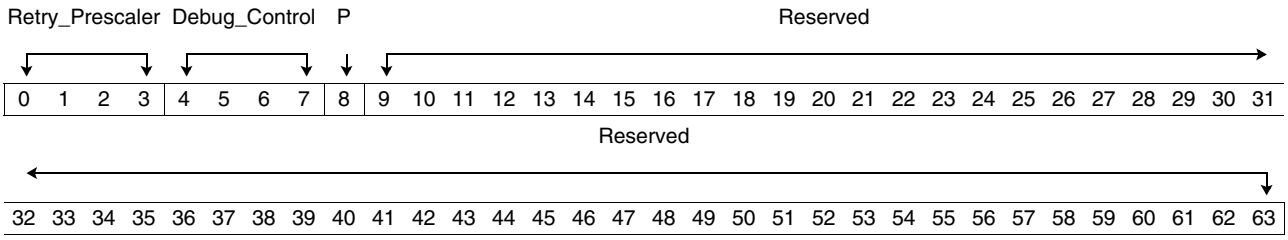


Bits	Field Name	Description
0	Reserved	Bits are not implemented; all bits read back zero.
1	SFE	I/O segment fault mask exception enable
2	PFE	I/O page fault mask exception enable
3:63	Reserved	Bits are not implemented; all bits read back zero.

5.9 IOC Translation Configuration Register (IOC_XlateCfg)

This register configures the retry backoff prescaler and power management of the I/O translation (Xlate) hardware.

Register Short Name	IOC_XlateCfg	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'510930'	**Memory Map Area**	IOC Address Translation
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IOC



Bits	Field Name	Description
0:3	Retry_Prescaler	Prescaler for the decrement value used to determine the amount of time before re-issuing an element interconnect bus (EIB) read after a Retry combined response. The first 2 bits of this field are used to select which decrement of the Linear Feedback Shift Register (LFSR) backoff to apply. The next 2 bits are used to select the LFSR sequence length. **Encode** **Decrement Rate** **LFSR Sequence Length** 00x 1:1 01x 1:2 10x 1:4 11x 1:8 x00 281 x01 317 x10 439 x11 487
4:7	Debug_Control	Debug control. Nonzero values indicate conditions that halt the I/O translation (Xlate) hardware. 0000 Never halt 0001 I/O segment fault or I/O page fault occurs 0010 I/O address translation exception interrupt is signalled to the internal interrupt controller (IIC) 0011 Write to the IOPT Cache Invalidate Register that invalidates way 0 0100 Write to the IOPT Cache Invalidate Register that invalidates way 1 0101 Write to the IOPT Cache Invalidate Register that invalidates way 2 0110 Write to IOPT Cache Invalidate Register that invalidates way 3 0111 Write to the IOST cache 1000 Write to the IOPT cache directory for the first way 1001 Write to the IOPT cache directory for the second way 1010 Write to the IOPT cache directory for the third way 1011 Write to the IOPT cache directory for the fourth way 1100 IOST cache miss or IOPT cache miss occurs on I/O translation 1101 Xlate hardware issues a "clear" to XIN logic All values not shown above are invalid.

Bits	Field Name	Description
8	P	Power management control for I/O translation (Xlate) hardware. 0 Enable latches. 1 Disable latches. When this bit is disabled, all latches in the Xlate hardware are disabled. The Xlate hardware can be re-enabled by asserting the HARD_RESET# signal to reset the Cell Broadband Engine.
9:63	Reserved	Bits are not implemented; all bits read back zero.

6. Internal Interrupt Controller (IIC) MMIO Registers

This section describes the IIC memory-mapped I/O (MMIO) registers. *Table 6-1* shows the IIC address translation MMIO memory map and lists the IIC registers. The internal interrupt controller (IIC) space starts at x‘508000’ and ends at x‘5084FF’. Offsets are from the start of the IIC register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

6.1 IIC MMIO Memory Map

Table 6-1. IIC Memory Map (Page 1 of 2)

Hexadecimal Offset (x‘508*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x‘000’ – x‘3FF’	Reserved			
x‘400’	*IIC Thread 0 Interrupt Pending Port Register (IIC_IPP0)* (nondestructive)	64	R	*Section 6.2.1* on page 147
x‘408’	*IIC Thread 0 Interrupt Pending Port Register (IIC_IPP0)* (destructive)	64	R	*Section 6.2.1* on page 147
x‘410’	*IIC Thread 0 Interrupt Generation Port Register (IIC_IGP0)*	64	W	*Section 6.2.2* on page 148
x‘418’	*IIC Thread 0 Current Priority Level Register (IIC_CPL0)*	64	R/W	*Section 6.2.3* on page 149
x‘420’	*IIC Thread 1 Interrupt Pending Port Register (IIC_IPP1)* (nondestructive)	64	R	*Section 6.2.4* on page 150
x‘428’	*IIC Thread 1 Interrupt Pending Port Register (IIC_IPP1)* (destructive)	64	R	*Section 6.2.4* on page 150
x‘430’	*IIC Thread 1 Interrupt Generation Port Register (IIC_IGP1)*	64	W	*Section 6.2.5* on page 151
x‘438’	*IIC Thread 1 Current Priority Level Register (IIC_CPL1)*	64	R/W	*Section 6.2.6* on page 152
x‘440’	*IIC Interrupt Routing Register (IIC_IR)*	64	R/W	*Section 6.2.7* on page 153
x‘448’	*IIC_Interrupt Status Register (IIC_IS)*	64	R/W	*Section 6.2.8* on page 154
x‘450’ – x‘4FF’	Reserved			

Table 6-1. IIC Memory Map (Page 2 of 2)

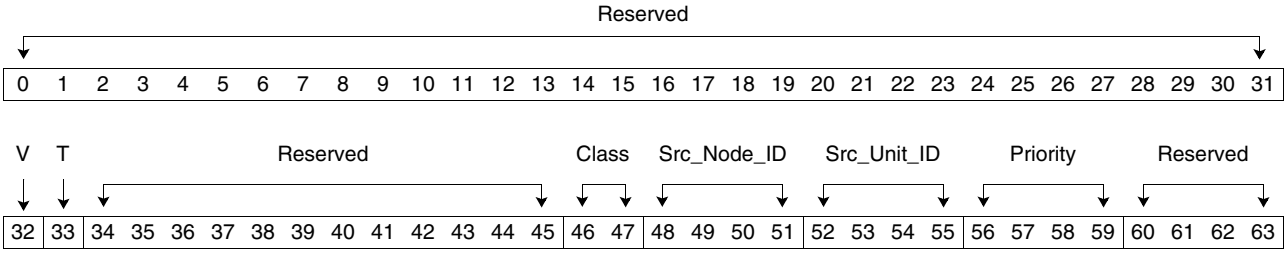
Hexadecimal Offset (x‘508*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x‘500’ x‘508’ x‘510’ x‘518’ x‘520’ x‘528’	*IOC Fault Isolation Register Reset* *IOC Fault Isolation Register Set* *IOC Checkstop Enable Register* *IOC Fault Isolation Error Mask Register* *IOC System Error Enable Register* *IOC Fault Isolation Register*	64	R/W	*Section A.11* on page 345
x‘530’ – x‘FFF’	Reserved			

6.2 IIC Register Descriptions

6.2.1 IIC Thread 0 Interrupt Pending Port Register (IIC_IPP0)

The IIC_IPP0 Register allows software to read the interrupt source and other information about any pending interrupts. After software reads the pending port, the next highest interrupt is loaded into the Interrupt Pending Port (IPP). When reading the IPP nondestructively, the value of the associated Current Priority Level (CPL) Register (IIC_CPL0) is not updated with the interrupt in the IPP. When reading the IPP destructively, the CPL Register takes on the priority of the interrupt in the IPP. The destructive read address offset is always 8 bytes beyond the nondestructive read offset.

Register Short Name	IIC_IPP0	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘508400’ Nondestructive x‘508408’ Destructive	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC



Bits	Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32	V	Interrupt Valid 0 No interrupt pending. 1 Interrupt pending.
33	T	Interrupt Type 0 Interrupt from Synergistic Processor Element (SPE), external device, or external interrupt controller 1 Interrupt from thread 0 Interrupt Generation Port
34:45	Reserved	Bits are not implemented; all bits read back zero.
46:47	Class	Interrupt Class. Returns zeros when T = ‘1’.
48:51	Src_Node_ID	Interrupt Source Cell Broadband Engine Interface (BIF) Node ID. Returns zeros when T = ‘1’.
52:55	Src_Unit_ID	Interrupt Source Unit ID. Returns zeros when T = ‘1’.
56:59	Priority	Interrupt Priority
60:63	Reserved	Bits are not implemented; all bits read back zero.

6.2.2 IIC Thread 0 Interrupt Generation Port Register (IIC_IGP0)

The IIC_IGP0 Register allows software to generate interrupts to PowerPC Processor Element (PPE) thread 0.

Register Short Name	IIC_IGP0	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'508410'	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC

Bits	Field Name	Description
0:55	Reserved	Bits are not implemented; all bits read back zero.
56:59	Priority	Interrupt Priority
60:63	Reserved	Bits are not implemented; all bits read back zero.

6.2.3 IIC Thread 0 Current Priority Level Register (IIC_CPL0)

The IIC_CPL0 Register allows software to mask interrupts of a specified priority. Software directly loads the CPL using an MMIO write of the register or indirectly using a destructive read of the IPP. When executing a destructive read of the IPP, the priority of the interrupt in the IPP is written to the CPL if that interrupt is valid. Otherwise, the content is unchanged.

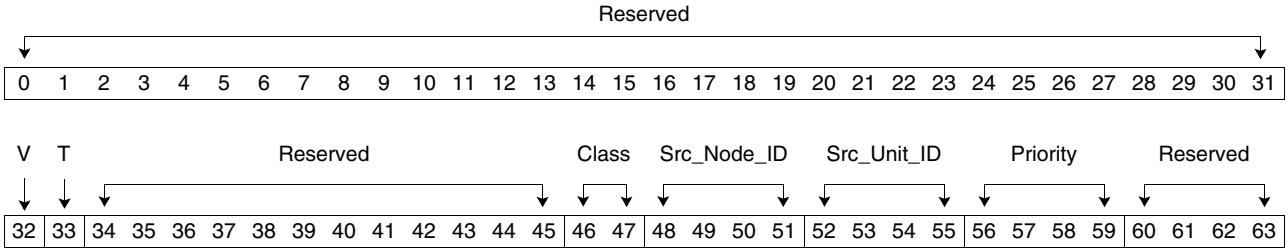
Register Short Name	IIC_CPL0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'508418'	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC

Bits	Field Name	Description
0:55	Reserved	Bits are not implemented; all bits read back zero.
56:60	Priority	If the priority of the interrupt in the IIC_IPP0 register is numerically less than the priority in this register, then the thread 0 external interrupt signal to the PPU is active.
61:63	Reserved	Bits are not implemented; all bits read back zero.

6.2.4 IIC Thread 1 Interrupt Pending Port Register (IIC_IPP1)

The IIC_IPP1 Register allows software to read the interrupt source and other information about any pending interrupts. After software reads the pending port, the next highest interrupt is loaded into the IPP. When reading the IPP nondestructively, the value of the associated Current Priority Level (CPL) Register (IIC_CPL1) is not updated with the interrupt in the IPP. When reading the IPP destructively, the CPL Register takes on the priority of the interrupt in the IPP. The destructive read address offset is always 8 bytes beyond the nondestructive read offset.

Register Short Name	IIC_IPP1	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘508420’ Nondestructive x‘508428’ Destructive	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC



Bits	Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32	V	Interrupt Valid 0 No interrupt pending 1 Interrupt pending
33	T	Interrupt Type 0 Interrupt from Synergistic Processor Element (SPE), external device, or external interrupt controller 1 Interrupt from thread 1 Interrupt Generation Port
34:45	Reserved	Bits are not implemented; all bits read back zero.
46:47	Class	Interrupt Class. Returns zeros when T = ‘1’.
48:51	Src_Node_ID	Interrupt Source Cell Broadband Engine Interface (BIF) Node ID. Returns zeros when T = ‘1’.
52:55	Src_Unit_ID	Interrupt Source Unit ID. Returns zeros when T = ‘1’.
56:59	Priority	Interrupt Priority
60:63	Reserved	Bits are not implemented; all bits read back zero.

6.2.5 IIC Thread 1 Interrupt Generation Port Register (IIC_IGP1)

The IIC_IGP1 Register allows software to generate interrupts to PowerPC Processor Element (PPE) thread 1.

Register Short Name	IIC_IGP1	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'508430'	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC

Bits	Field Name	Description
0:55	Reserved	Bits are not implemented; all bits read back zero.
56:59	Priority	Interrupt Priority
60:63	Reserved	Bits are not implemented; all bits read back zero.

6.2.6 IIC Thread 1 Current Priority Level Register (IIC_CPL1)

The IIC_CPL1 Register allows software to mask interrupts of a specified priority. Software directly loads the CPL using an MMIO write of the register or indirectly using a destructive read of the IPP. When executing a destructive read of the IPP, the priority of the interrupt in the IPP is written to the CPL if that interrupt is valid. Otherwise, the content is unchanged.

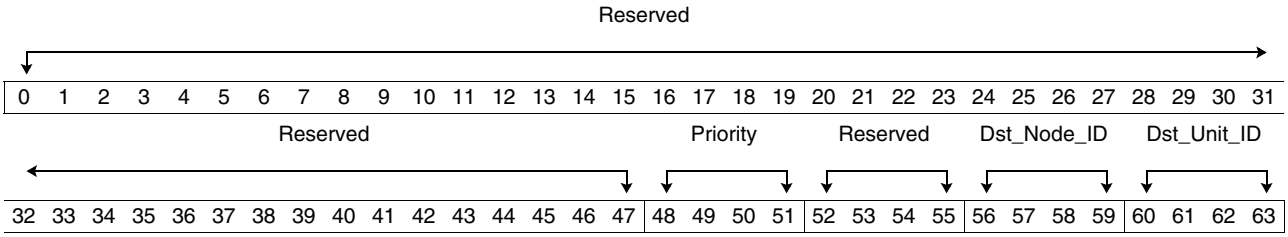
Register Short Name	IIC_CPL1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘508438’	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC

Bits	Field Name	Description
0:55	Reserved	Bits are not implemented; all bits read back zero.
56:60	Current_Priority	If the priority of the interrupt in the IIC_IPP1 is numerically less than the priority in this register, then the thread 1 external interrupt signal to the PPU is active.
61:63	Reserved	Bits are not implemented; all bits read back zero.

6.2.7 IIC Interrupt Routing Register (IIC_IR)

The IIC_IR Register configures the priority and destination of interrupts in the IIC_IS Register.

Register Short Name	IIC_IR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘508440’	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC



Bits	Field Name	Description
0:47	Reserved	Bits are not implemented; all bits read back zero.
48:51	Priority	Interrupt Priority. This field corresponds to the 4 bits of the priority field in the IIC_IPP0 and IIC_IPP1.
52:55	Reserved	Bits are not implemented; all bits read back zero.
56:59	Dst_Node_ID	Destination Node BIF ID for interrupts in the IIC_IS register.
60:63	Dst_Unit_ID	Destination Unit ID for interrupts in the IIC_IS register. Interrupt packets routed to I/O Interface 0 (IOIF0) have the same destination unit ID as I/O Controller 0 (IOC0). Those that are routed to I/O Interface 1 (IOIF1) have the same destination unit ID as IOC1. The valid values are: 0000 IOC0 1011 I/O Controller 1 (IOC1) 1110 Processor thread 0 1111 Processor thread 1

6.2.8 IIC_Interrupt Status Register (IIC_IS)

The IIC_IS Register records interrupt conditions from the memory interface controller (MIC), Element Interconnect Bus (EIB) unit, I/O address translation (Xlate), performance monitoring (PM), and token manager (TKM). When an interrupt occurs, the corresponding bit is set to '1'. If the Interrupt Status Register is nonzero, the IIC creates a class 1 interrupt that goes to either the EIB, IPP0, or IPP1 depending on how the IIC_IR is configured. After resetting the interrupt condition in the source unit, software resets the interrupt by writing '1' to the corresponding bit position in the Interrupt Status Register. Writing '0' to the corresponding bit has no effect on that bit. Writing this register must occur to confirm handling of interrupts in the IIC_IS Register.

Register Short Name	IIC_IS	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'508448'	**Memory Map Area**	Internal Interrupt Controller (IIC)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IIC

Bits	Field Name	Description
0:58	Reserved	All bits read back zero.
59	ELDI	EIB Possible Livelock Detection Interrupt 0 Inactive 1 Active **Note:** See EIB_Int[0] for additional information
60	MATBFI	Memory Interface Controller (MIC) Auxiliary Trace Buffer Full Interrupt 0 Inactive 1 Active
61	ATI	I/O Address Translation Interrupt Active
62	PMI	Performance Monitor Interrupt Active
63	TMI	Token Manager Interrupt Active

7. Memory Interface Controller (MIC) MMIO Registers

This section describes the MIC memory-mapped I/O (MMIO) registers. *Table 7-1* shows the MIC MMIO memory map and lists the MIC registers. The MIC unit shares a memory space with the token manager (TKM) unit. This shared space starts at x‘50A000’ and ends at x‘50AFFF’. Offsets are from the start of the MIC register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

Note: YRAC (Yellowstone Rambus ASIC Cell) is the old term for the external data representation (XDR) I/O Cell or XIO. The old term still exists in register names.

Note: The MIC interfaces to the XDR I/O Cell, which orders its bits in big-endian notation. Fields in the MIC registers that access the XDR I/O Cell facilities (buses, registers or bit numbering) have the ordering of the bits described explicitly in those fields. The names of the fields that access these facilities are in uppercase. The bit ranges of these fields are represented using big-endian notation (for example, ALL_CAPS[start-bit:endbit], where *startbit* is the most significant bit and *endbit* is the least significant bit).

7.1 MIC MMIO Memory Map

Table 7-1 provides the MIC MMIO memory map. Reserved registers return zero on read access. All the MIC registers have privileged mode access.

Table 7-1. MIC MMIO Memory Map (Page 1 of 3)

Hexadecimal Offset (x‘50A*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
MIC_CTL MMIO Registers				
x‘040’	*MIC Control Configuration Register 2 (MIC_Ctl_Cnfg2)*	64	R/W	*Section 7.2.1* on page 158
x‘048’	Reserved			
x‘050’	*MIC Auxiliary Trace Base Address Register (MIC_Aux_Trc_Base)*	64	R/W	*Section 7.2.2* on page 160
x‘058’	*MIC Auxiliary Trace Max Address Register (MIC_Aux_Trc_Max_Addr)*	64	R/W	*Section 7.2.3* on page 161
x‘060’	*MIC Auxiliary Trace Current Address Register(MIC_Aux_Trc_Cur_Addr)*	64	R/W	*Section 7.2.4* on page 162
x‘068’	*MIC Auxiliary Trace GRF Address (MIC_Aux_Trc_Grf_Addr)*	64	R/W	*Section 7.2.5* on page 163
x‘070’	*MIC Auxiliary Trace GRF Data (MIC_Aux_Trc_Grf_Data)*	64	R/W	*Section 7.2.6* on page 164
x‘078’	Reserved			

Table 7-1. MIC MMIO Memory Map (Page 2 of 3)

Hexadecimal Offset (x'50A*nnn*')	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x'080' x'1C0'	*MIC Control Configuration Register n [n = 0,1]* *(MIC_Ctl_Cnfg_n [n = 0,1])*	64	R/W	*Section 7.2.7* on page 165
x'088' x'1C8'	Reserved			
x'090' x'1D0'	*MIC Slow Mode Fast Timer Register n [n = 0,1]* *(MIC_Slow_Fast_Timer_n [n = 0,1])*	64	R/W	*Section 7.2.8* on page 166
x'098' x'1D8'	*MIC Slow Mode Next Timer Register n [n = 0,1]* *(MIC_Slow_Next_Timer_n [n = 0,1])*	64	R/W	*Section 7.2.9* on page 167
x'0A0' x'1E0'	*MIC Calibration Addresses n [n = 0,1]* *(MIC_Calibration_Addr_n [n = 0,1])*	64	R/W	*Section 7.2.10* on page 168
x'0A8' x'1E8'	*MIC Token Manager Threshold Levels n [n = 0,1]* *(MIC_TM_Threshold_n [n = 0,1])*	64	R/W	*Section 7.2.11* on page 169
x'0B0' x'1F0'	*MIC Queue Burst Sizes Register n [n = 0,1]* *(MIC_Que_BurstSize_n [n = 0,1])*	64	R/W	*Section 7.2.12* on page 170
x'0B8' x'1F8'	Reserved			
XDR DRAM Controller (YC) Registers				
x'0C0' x'180'	*MIC Device Configuration Register n [n = 0,1]* *(MIC_Dev_Cfg_n [n = 0,1])*	64	R/W	*Section 7.3.1* on page 171
x'0C8' x'188'	*MIC Memory Configuration Register n [n = 0,1]* *(MIC_Mem_Cfg_n [n = 0,1])*	64	R/W	*Section 7.3.2* on page 173
x'0D0' x'190'	*MIC tRCD and Precharge Register n [n = 0,1]* *(MIC_Trcd_Pchg_n [n = 0,1])*	64	R/W	*Section 7.3.3* on page 175
x'0D8' x'198'	*MIC Command Duration Register n [n = 0,1]* *(MIC_Cmd_Dur_n [n = 0,1])*	64	R/W	*Section 7.3.4* on page 177
x'0E0' x'1A0'	*MIC Command Spacing Register n [n = 0,1]* *(MIC_Cmd_Spc_n [n = 0,1])*	64	R/W	*Section 7.3.5* on page 178
x'0E8' x'1A8'	*MIC Dataflow Control Register n [n = 0,1]* *(MIC_DF_Ctl_n [n = 0,1])*	64	R/W	*Section 7.3.6* on page 180
Dataflow (DF) Registers				
x'0F0' x'1B0'	*MIC Dataflow XIO PTCal Register n [n = 0,1]* *(MIC_XIO_PTCal_Data_n [n = 0,1])*	128	R/W	*Section 7.4.1* on page 181
x'0F8' x'1B8'	*MIC Dataflow Error Correction Code Address Register n [n = 0,1] (MIC_Ecc_Addr_n [n = 0,1])*	64	R/W	*Section 7.4.2* on page 184
XDR DRAM Controller (YC) Registers				
x'100' x'140'	*YRAC Data Register n [n = 0,1]* *(Yreg_YRAC_Dta_n [n = 0,1])*	64	R/W	*Section 7.5.1* on page 185
x'108' x'148'	*YDRAM Data Register n [n = 0,1]* *(Yreg_YDRAM_Dta_n [n = 0,1])*	64	W	*Section 7.5.2* on page 187
x'110' x'150'	*MIC Status Register n [n = 0,1]* *(MIC_Yreg_Stat_n [n = 0,1])*	64	R/W	*Section 7.5.3* on page 189
x'118' x'158'	*Initialization Control Register n [n = 0,1]* *(Yreg_Init_Ctl_n [n = 0,1])*	64	R/W	*Section 7.5.4* on page 192
x'120' x'160'	*Initialization Constants Register n [n = 0,1]* *(Yreg_Init_Cnts_n [n = 0,1])*	64	R/W	*Section 7.5.5* on page 194

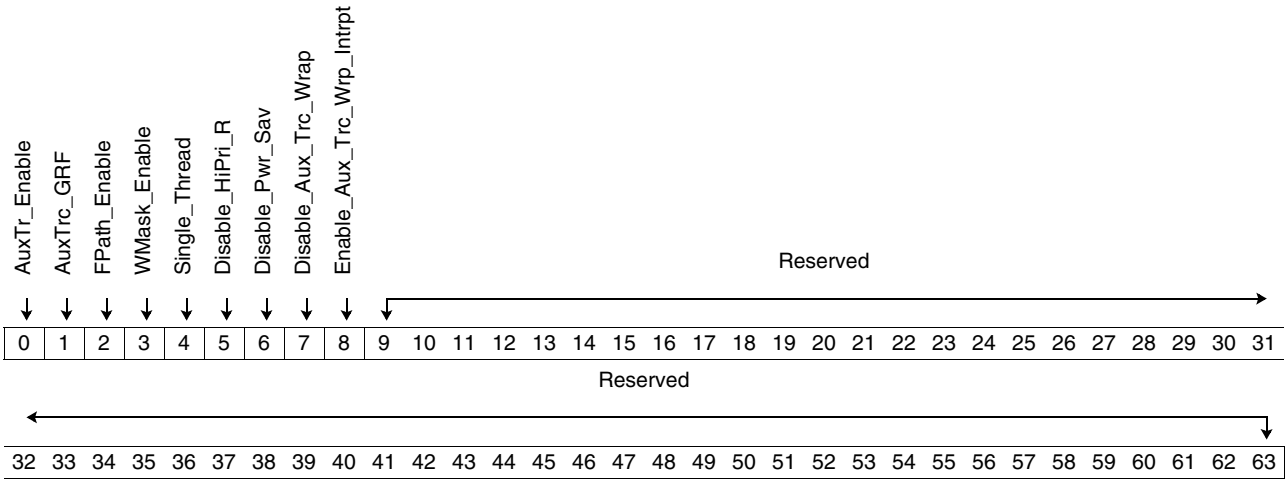
Table 7-1. MIC MMIO Memory Map (Page 3 of 3)

Hexadecimal Offset (x'50A*nnn*')	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x'128' x'168'	Reserved			
x'130' x'170'	*MIC Periodic Timing Calibration Address Register n [n = 0,1] (MIC_PTCal_Adr_n [n = 0,1])*	64	R/W	*Section 7.5.6* on page 196
x'138' x'178'	Reserved			
x'200'	*MIC Refresh and Scrub Register (MIC_Ref_Scb)*	64	R/W	*Section 7.5.7* on page 197
x'208'	*MIC Execute Register (MIC_Exc)*	64	R/W	*Section 7.5.8* on page 198
x'210'	*MIC Maintenance Config Register (MIC_Mnt_Cfg)*	64	R/W	*Section 7.5.9* on page 200
Dataflow (DF) Registers				
x'218'	*MIC Dataflow Configuration Register (MIC_DF_Config)*	64	R/W	*Section 7.6.1* on page 201
x'220' x'228'	Reserved			
Fault Isolation (FIR) Registers				
x'230'	*MIC Fault Isolation and Checkstop Enable Registers*	64	R/W	*Section 7.7* on page 203
x'238'	*MIC ErrorMask/RecErrorEnable/Debug Control Register (MIC_FIR_Debug)*	64	R/W	*Section 7.7.2* on page 205

7.2 MIC_CTL MMIO Registers

7.2.1 MIC Control Configuration Register 2 (MIC_Ctl_Cnfg2)

Register Short Name	MIC_Ctl_Cnfg2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A040’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

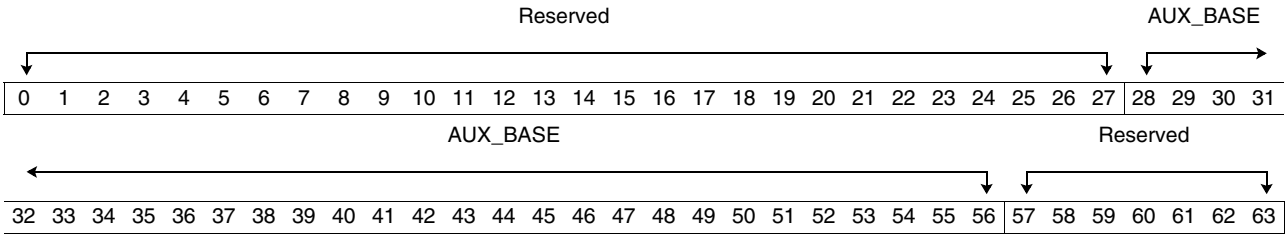


Bits	Field Name	Description
0	AuxTr_Enable	Enable auxiliary trace 0 Auxiliary trace is disabled. 1 Auxiliary trace is enabled. Auxiliary trace cannot be enabled in Slow Core Mode.
1	AuxTrc_GRF	Enable auxiliary trace GRF debug 0 Auxiliary trace GRF debug is disabled. 1 Auxiliary trace GRF debug is enabled. **Note:** If this bit is enabled, the MIC_Aux_Trc_Grf_Data register needs to be read twice to get the correct data for the current GRF address.
2	FPath_Enable	Enable Fast Path mode. Speculative read must be enabled if Fast Path is enabled (MIC_Ctl_Cnfg_n[1] = ‘0’). 0 Fast Path mode is disabled. 1 Fast Path mode is enabled.
3	WMask_Enable	Enable Write Masking 0 Write Masking is disabled. 1 Write Masking is enabled.

Bits	Field Name	Description
4	Single_Thread	Enable Single-Threaded mode In single-threaded mode, the MIC accepts and executes only one command at a time and the MIC retries all other commands it receives until that command has executed. When single-threaded mode is disabled, the MIC can accept up to 64 writes and 64 reads before retrying commands on the bus. **Note:** The MIC single-threaded mode is not related to the multithreading features of the PPE.
5	Disable_HiPri_R	Disable High Priority Reads 0 High Priority Reads are enabled. 1 High Priority Reads are disabled.
6	Disable_Pwr_Sav	Disable Power Savings Mode 0 Power Savings Mode is enabled. 1 Power Savings Mode is disabled. **Note:** Power Savings Mode must be disabled for Display/Alter (Auxiliary Trace is disabled).
7	Disable_Aux_Trc_Wrap	Disable Auxiliary Trace Wrap 0 Auxiliary Trace Wrap is enabled. 1 Auxiliary Trace Wrap is disabled. **Note:** Auxiliary Trace Wrap must be enabled for Display/Alter.
8	Enable_Aux_Trc_Wrp_Intrpt	Enable Auxiliary Trace Wrap Interrupt 0 Auxiliary Trace Wrap Interrupt is disabled. 1 Auxiliary Trace Wrap Interrupt is enabled.
9:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.2 MIC Auxiliary Trace Base Address Register (MIC_Aux_Trc_Base)

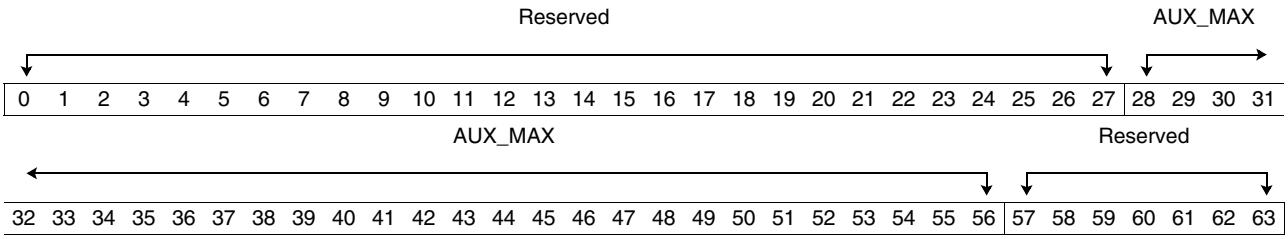
Register Short Name	MIC_Aux_Trc_Base	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A050’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)



Bits	Field Name	Description
0:27	Reserved	Bits are not implemented; all bits read back zero.
28:56	AUX_BASE	Base address of the auxiliary trace array
57:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.3 MIC Auxiliary Trace Max Address Register (MIC_Aux_Trc_Max_Addr)

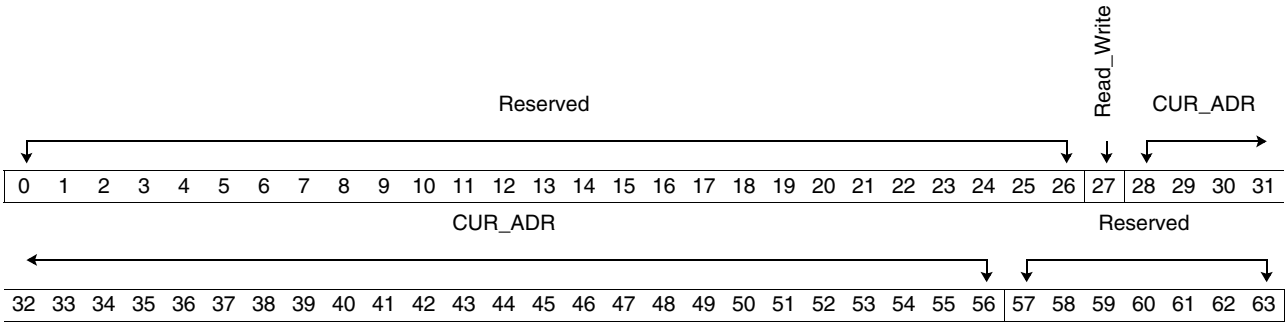
Register Short Name	MIC_Aux_Trc_Max_Addr	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A058’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)



Bits	Field Name	Description
0:27	Reserved	Bits are not implemented; all bits read back zero.
28:56	AUX_MAX	Maximum address of the auxiliary trace array When the current address matches the maximum address, the command has been sent to the rest of the CTL subunit, the data has been sent to the DF subunit if it is a write, and the current address is reset to the base address.
57:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.4 MIC Auxiliary Trace Current Address Register(MIC_Aux_Trc_Cur_Addr)

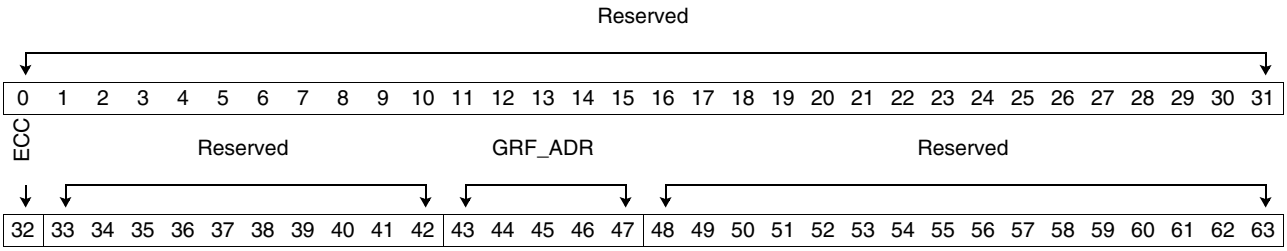
Register Short Name	MIC_Aux_Trc_Cur_Addr	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A060’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)



Bits	Field Name	Description
0:26	Reserved	Bits are not implemented; all bits read back zero.
27	Read_Write	Read and write field 0 Write. The store is sent to memory if there is a cache line of data in the data array. See the CUR_ADR field, MIC_Aux_Trc_Cur_Addr[28:56]. 1 Read. The read is sent to memory when this register is written. This register increments after every access to memory. See the CUR_ADR field, MIC_Aux_Trc_Cur_Addr[28:56].
28:56	CUR_ADR	Address of the next cache line. See the Read_Write field, MIC_Aux_Trc_Cur_Addr[27].
57:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.5 MIC Auxiliary Trace GRF Address (MIC_Aux_Trc_Grf_Addr)

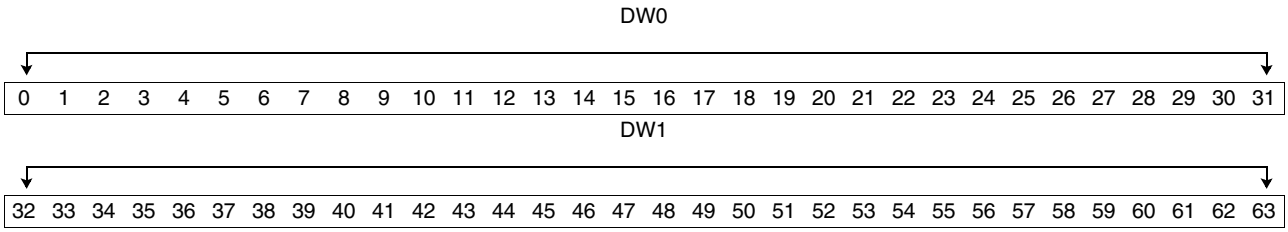
Register Short Name	MIC_Aux_Trc_Grf_Addr	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A068’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)



Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32	ECC	ECC write. 0 The 8 bytes of data in the MIC_Aux_Trc_Grf_Addr register are stored to the first 8 bytes of this entry in the auxiliary trace data array. 1 The first byte in the MIC_Aux_Trc_Grf_Addr register is stored to the ninth byte of this entry in the auxiliary trace data array.
33:42	Reserved	Bits are not implemented; all bits read back zero.
43:47	GRF_ADR	Address of the next entry in the auxiliary trace data array. This register is only used for display/alter (auxiliary trace disabled). The register itself does not auto-increment, but the GRFWrtAdr (which is set when this register is written) increments when data is stored in the data array. For the first cache line in the data array, the address should be set to ‘00000’. For the second cache line, it should be set to ‘10000’.
48:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.6 MIC Auxiliary Trace GRF Data (MIC_Aux_Trc_Grf_Data)

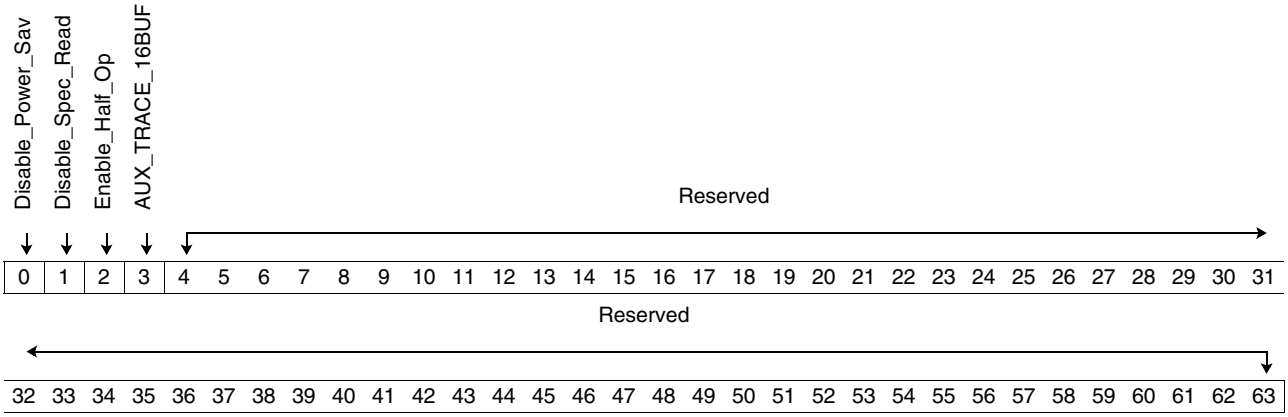
Register Short Name	MIC_Aux_Trc_Grf_Data	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A070’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)



Bits	Field Name	Description
0:31	DW0	Data to be stored into the auxiliary trace data array. If MIC_Aux_Trc_Grf_Addr[0], the ECC entry bit, is set, only bits [0:7] of this register are valid. **Note:** MIC_Ctl_Cnfg2[1] must be set to ‘1’ to read this register.
32:63	DW1	Data to be stored into the auxiliary trace data array. If MIC_Aux_Trc_Grf_Addr[0], the ECC entry bit, is set, only bits [0:7] of this register are valid. **Note:** MIC_Ctl_Cnfg2[1] must be set to ‘1’ to read this register.

7.2.7 MIC Control Configuration Register n [n = 0,1] (MIC_Ctl_Cnfg_n [n = 0,1])

Register Short Name	MIC_Ctl_Cnfg_0 MIC_Ctl_Cnfg_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A080' x'50A1C0'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

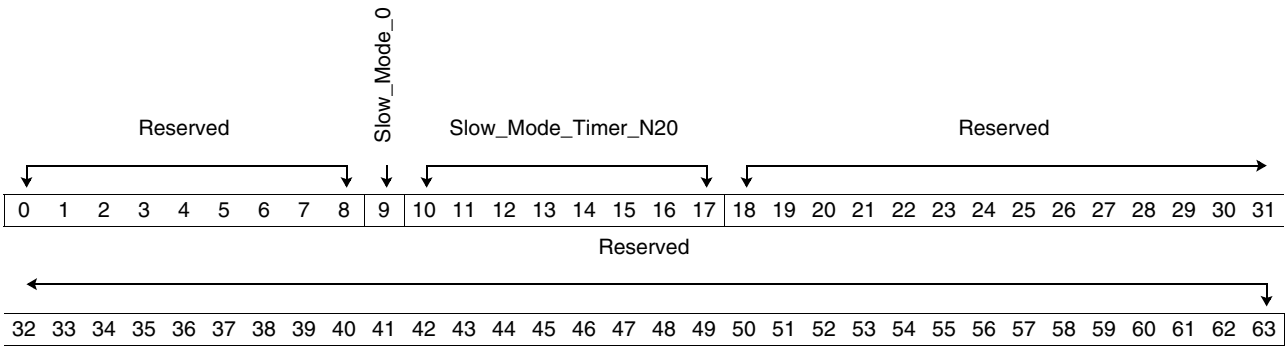


Bits	Field Name	Description
0	Disable_Power_Sav	Disable Power Savings Mode. Power Savings Mode must be disabled for Display/Alter (Auxiliary Trace is disabled). 0 Power savings mode is enabled. 1 Power savings mode is disabled.
1	Disable_Spec_Read	Disable Speculative Reads 0 Speculative reads are enabled. 1 Speculative reads are disabled.
2	Enable_Half_Op	Enable Half (64-byte) Operations 0 64-byte operations are disabled. 1 64-byte operations are enabled.
3	AUX_TRACE_16BUF	Reserve 16 queue entries for auxiliary trace 0 8 queue entries for auxiliary trace. 1 16 queue entries for auxiliary trace.
4	Reserved	Reserved. Software should ignore value read and write only zero.
5:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.8 MIC Slow Mode Fast Timer Register n [n = 0,1] (MIC_Slow_Fast_Timer_n [n = 0,1])

Bit-fields in this register must be set to the recommended values in order for the Power Management modes on the MIC to work.

Register Short Name	MIC_Slow_Fast_Timer_0 MIC_Slow_Fast_Timer_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A090’ x‘50A1D0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

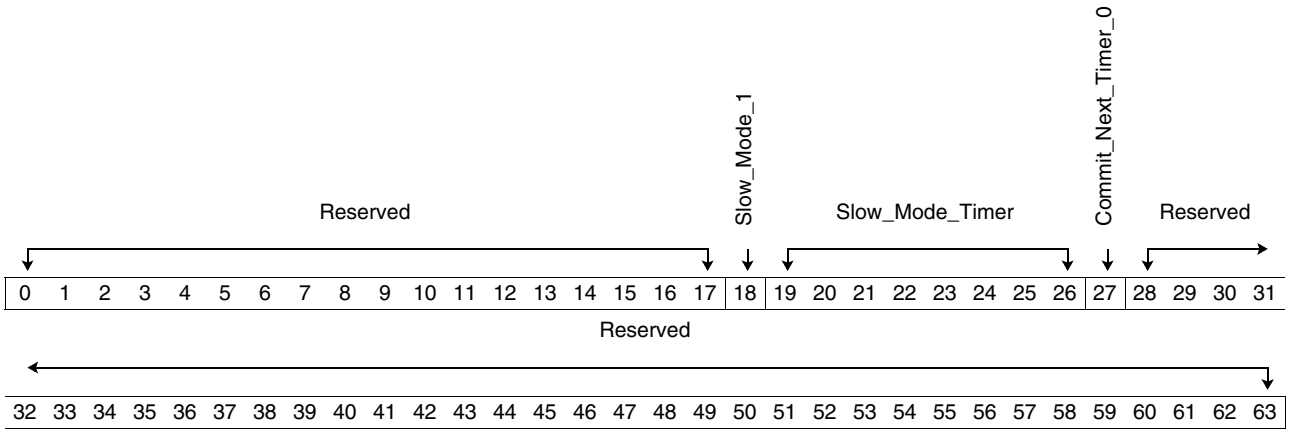


Bits	Field Name	Description
0:8	Reserved	Bits are not implemented. Bits read back the value of the Slow Mode Current Timer.
9	Slow_Mode_0	Slow Mode. The recommended setting for this bit is ‘1’. 0 Normal mode 1 Slow mode (when NClk < 2 × MiClk)
10:17	Slow_Mode_Timer_N20	Number of cycles between when commands can be selected for the slowest clock rate supported. The recommended setting for this field is x‘FF’. **Note:** The read and write masks for this register do not match. When this register is written, MIC_Slow_Fast_Timer[9:17] are written. When this register is read, MIC_Slow_Fast_Timer[0:8] of the read data are the Slow Mode Current Timer, MIC_Slow_Fast_Timer[9:17] are the Slow Mode N/20 Timer, and MIC_Slow_Fast_Timer[18:26] are the Slow Mode Next Timer.
18:26	Reserved	Bits are not implemented. Bits read back the value of the Slow Mode Next Timer.
27:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.9 MIC Slow Mode Next Timer Register n [n = 0,1] (MIC_Slow_Next_Timer_n [n = 0,1])

Bit fields in this register must be set to the recommended values in order for the Power Management modes on the MIC to work.

Register Short Name	MIC_Slow_Next_Timer_0 MIC_Slow_Next_Timer_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A098' x'50A1D8'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

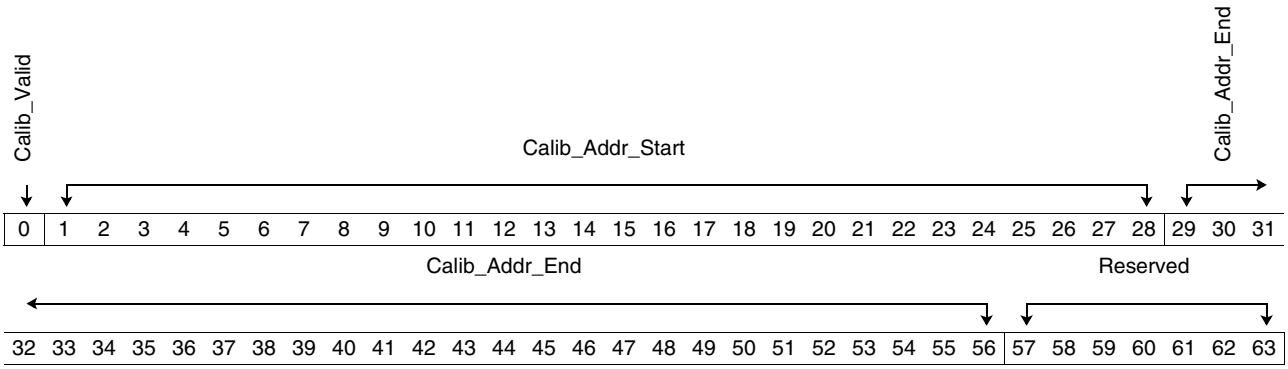


Bits	Field Name	Description
0:17	Reserved	Bits are not implemented. Bits [0:9] when read, return the Slow Mode Current Timer. Bits[9:17] return the value of the Slow Mode N/20 Timer.
18	Slow_Mode_1	Slow Mode. If the new clock frequency is NClk < 2 × MiClk, set this bit to a '1'; otherwise, set this bit to '0'. 0 Normal mode ('00010100'). 1 Slow mode (when NClk < 2 × MiClk)
19:26	Slow_Mode_Timer	Slow Mode Timer Number of cycles between the selection of commands for the next clock rate. This field should be set to 32 times the ratio of MiClk to NClk; it should never be set to less than 32 when you are in slow mode. The highest ratio supported is 8:1. Set to '00010100' for Normal mode. **Note:** The read and write masks for this register do not match. When this register is written, bits[18:26] are written. When this register is read, bits[0:8] of the read data are the Slow Mode Current Timer, bits[9:17] are the Slow Mode N/20 Timer, and bits[18:26] are the Slow Mode Next Timer.
27	Commit_Next_Timer_0	Write Only bit. Commits the values in Slow_Mode_1 and Slow_Mode_Timer bit fields. 0 Default 1 Commits the values in Slow_Mode_1 and Slow_Mode_Timer fields.
28:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.10 MIC Calibration Addresses n [n = 0,1] (MIC_Calibration_Addr_n [n = 0,1])

This register should only be written when scrubbing is disabled, because there is an asynchronous interface between where this register is located and the RLM that actually uses it.

Register Short Name	MIC_Calibration_Addr_0 MIC_Calibration_Addr_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0A0’ x‘50A1E0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

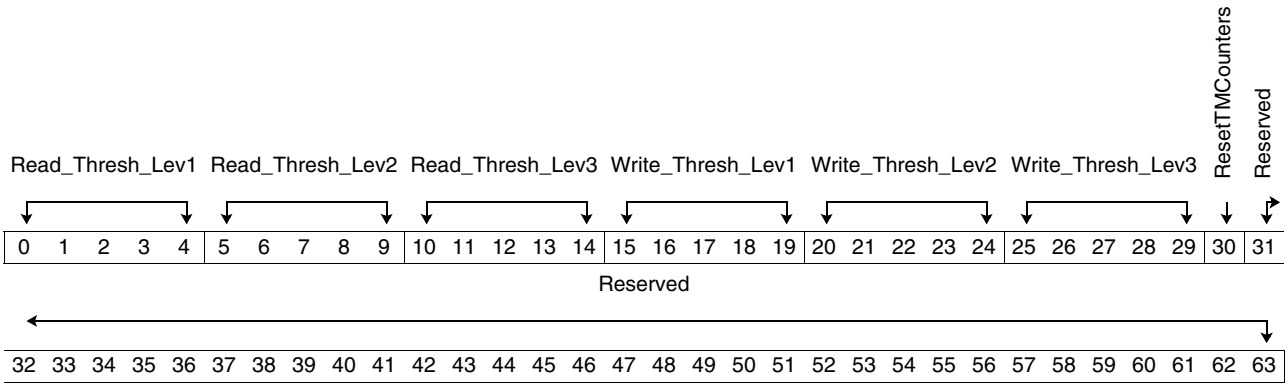


Bits	Field Name	Description
0	Calib_Valid	Calibration addresses valid. 0 Calibration range in this register is not enabled. 1 Calibration range is valid.
1:28	Calib_Addr_Start	Start address of range in memory that should not be scrubbed because it holds patterns for XIO cell periodic recalibration. This is the cache line address on this memory channel. See the programming note that follows this table.
29:56	Calib_Addr_End	Next address in memory that can be scrubbed. This is the end address of the memory range that should not be scrubbed plus 1. This field should be written only before scrubbing starts. By default, all bits are set to zero. This is the cache line address on this memory channel. See the programming note that follows this table. **Note:** MIC_Calibration_Addr[54:56] need to be equal to MIC_Calibration_Addr[26:28] in the Calib_Addr_Start field.
57:63	Reserved	Bits are not implemented; all bits read back zero.

Programming Note: Because MIC_Calibration_Addr[1:28] and MIC_Calibration_Addr[30:56] indicate the cache line address on this memory channel, the seven least significant bits are dropped off if a single memory channel is configured and the eight least significant bits are dropped off when two memory channels are present. When two memory channels are present, channel 0 has an implied ‘0’ appended to the right of the LSB of this value and channel 1 has an implied ‘1’ appended to the LSB of this value to form the effective cache-line offsets into the MIC.

7.2.11 MIC Token Manager Threshold Levels n [n = 0,1] (MIC_TM_Threshold_n [n = 0,1])

Register Short Name	MIC_TM_Threshold_0 MIC_TM_Threshold_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0A8’ x‘50A1E8’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

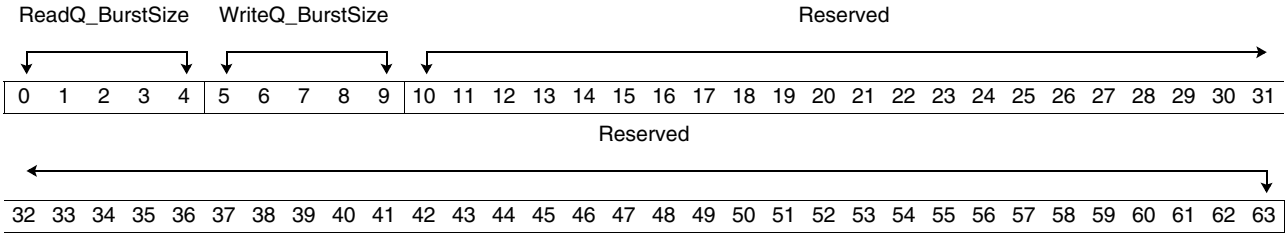


Bits	Field Name	Description
0:4	Read_Thresh_Lev1	Read threshold level 1 Five bits correspond to the queue entry level. There are 32 queues per XIO Cell. The highest level of the queues is sent to the token manager.
5:9	Read_Thresh_Lev2	Read threshold level 2 Five bits correspond to the queue entry level.
10:14	Read_Thresh_Lev3	Read threshold level 3 Five bits correspond to the queue entry level.
15:19	Write_Thresh_Lev1	Write threshold level 1 Five bits correspond to the queue entry level.
20:24	Write_Thresh_Lev2	Write threshold level 2 Five bits correspond to the queue entry level.
25:29	Write_Thresh_Lev3	Write threshold level 3 Five bits correspond to the queue entry level.
30	ResetTMCounters	Reset Token Manager Counters A write of ‘1’ resets the counters managing the token manager. A Read always returns 0.
31:63	Reserved	Bits are not implemented; all bits read back zero.

7.2.12 MIC Queue Burst Sizes Register n [n = 0,1] (MIC_Que_BurstSize_n [n = 0,1])

This register is used to select arbitration between reads and writes and must be set up during MIC Configuration.

Register Short Name	MIC_Que_BurstSize_0 MIC_Que_BurstSize_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0B0’ x‘50A1F0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)



Bits	Field Name	Description
0:4	ReadQ_BurstSize	Read queue burst size. Used in read versus write arbitration. When the number of reads queued inside the MIC is greater than ReadQ_BurstSize, the CTL design macro issues more reads in a row before switching to writes. If both ReadQ_BurstSize and WriteQ_BurstSize thresholds are exceeded, or neither has been exceeded, CTL issues an equal number of reads and writes in a row. The recommended value for this field is 4 (‘00100’).
5:9	WriteQ_BurstSize	Write queue burst size. Used in read versus write arbitration. When the number of writes queued inside the MIC is greater than WriteQ_BurstSize, the CTL design macro issues more writes in a row before switching to reads. If both ReadQ_BurstSize and WriteQ_BurstSize thresholds are exceeded, or neither has been exceeded, CTL issues an equal number of reads and writes in a row. The recommended value for this field is 12 (‘01100’).
10:63	Reserved	Bits are not implemented; all bits read back zero.

7.3 XDR DRAM Controller (YC) Registers

All XDR DRAM controller (YC) registers are in the MiClk domain (typically 1.6 GHz). All YC register read data is left-justified on the EIB.

Software must make sure that at least five MiClks have transpired before subsequent writes can be made to these control registers. If the clock frequencies cannot be determined, software can write and then read the register to ensure that the write has completed. This requirement does not apply to the Yreg_YDRAM_Dta and the Yreg_YRAC_Dta registers.

Configuration registers should be written first, followed by those that are used to configure the memory channel.

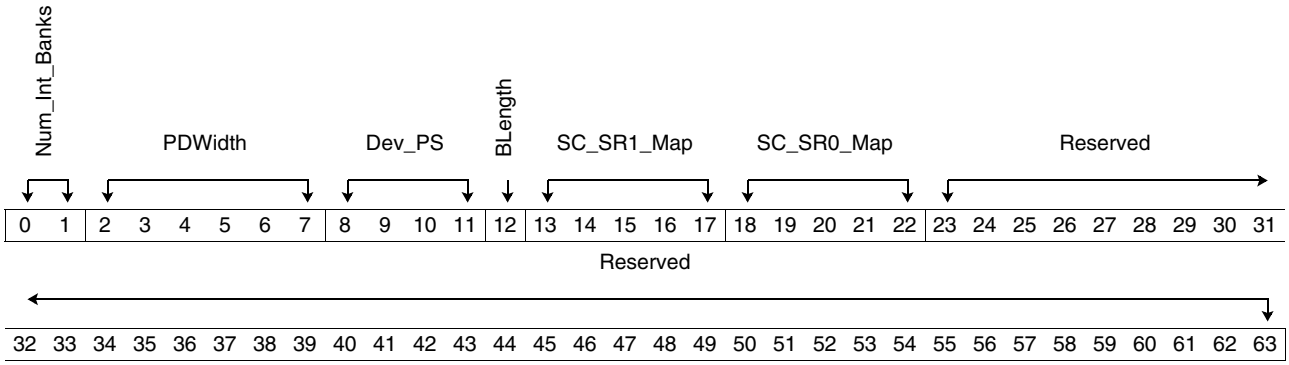
7.3.1 MIC Device Configuration Register n [n = 0,1] (MIC_Dev_Cfg_n [n = 0,1])

This register must be set during configuration of the MIC. These values should not be changed by hardware or software after configuration. Setting binary combinations not specified in the register description result in undefined behavior.

This register contains values pertaining to the XDR DRAM chip used. These values control the real-to-physical address mapping.

Both MIC_Dev_Cfg registers (one for each half) must be set to the same value.

Register Short Name	MIC_Dev_Cfg_0 MIC_Dev_Cfg_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A0C0' x'50A180'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0:1	Num_Int_Banks	Number of internal banks. 00 4 banks 01 8 banks 10 16 banks

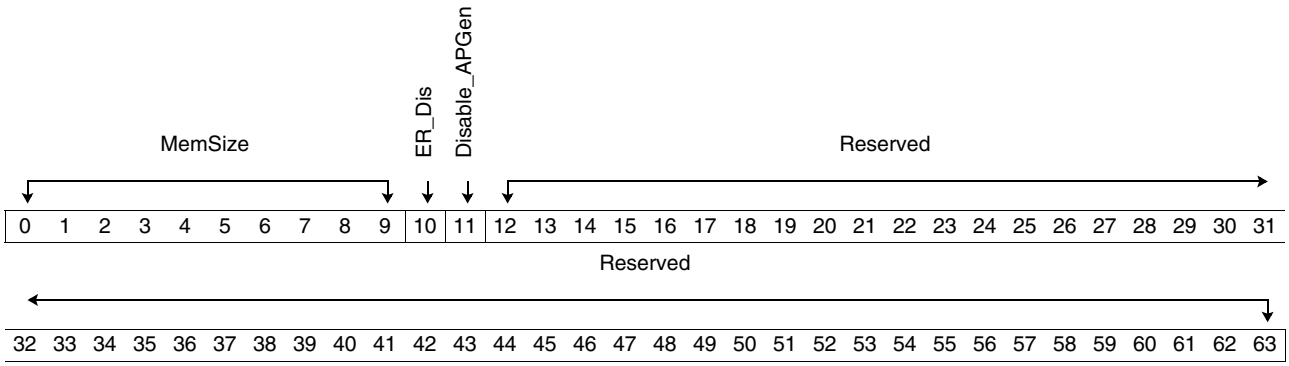
Bits	Field Name	Description
2:7	PDWidth	Programmed device width. 000001 by 1 (x1) 000010 by 2 (x2) 000100 by 4 (x4) 001000 by 8 (x8) 010000 by 16 (x16) 100000 by 32 (x32) This value is post-dynamic width adjustment (if used).
8:11	Dev_PS	Device page size. 0001 1 KB 0010 2 KB 0100 4 KB 1000 8 KB
12	BLength	Burst length (BL). 0 BL of 16 (tCC = 5 ns) 1 BL of 32 (tCC = 10 ns) where tCC is the column-to-column time
13:17	SC_SR1_Map	Subcolumn (SC) to subrow[1] (SR[1]) (MSB) mapping. 00000 Subpage activation is not used If subpage activation is used, choose one of the following: 10000 SC[4] is mapped to SR[1] 01000 SC[3] is mapped to SR[1] 00100 SC[2] is mapped to SR[1] 00010 SC[1] is mapped to SR[1] 00001 SC[0] is mapped to SR[1]
18:22	SC_SR0_Map	Subcolumn (SC) to subrow[0] (SR[0]) (LSB) mapping. 00000 Subpage activation is not used If subpage activation is used, choose one of the following: 10000 SC[4] is mapped to SR[0] 01000 SC[3] is mapped to SR[0] 00100 SC[2] is mapped to SR[0] 00010 SC[1] is mapped to SR[0] 00001 SC[0] is mapped to SR[0]
23:63	Reserved	Bits are not implemented; all bits read back zero.

7.3.2 MIC Memory Configuration Register n [n = 0,1] (MIC_Mem_Cfg_n [n = 0,1])

This register contains some miscellaneous memory configuration values. After startup, these values should not be changed by hardware or software.

The MIC_Mem_Cfg_0 and MIC_Mem_Cfg_1 registers must both be set to the same value.

Register Short Name	MIC_Mem_Cfg_0 MIC_Mem_Cfg_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0C8’ x‘50A188’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



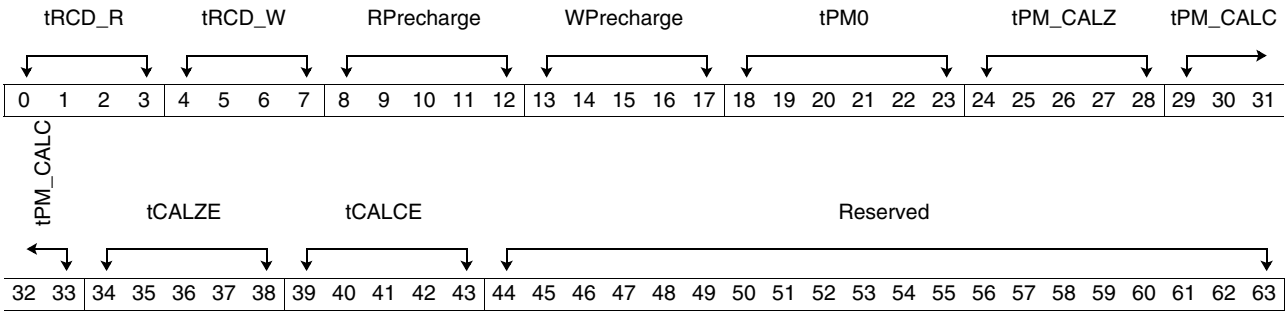
Bits	Field Name	Description
0:9	MemSize	Memory Size Enable vector. If only one memory channel is populated, make this vector match the memory capacity on the populated memory channel. If both memory channels are populated, make this vector match the memory capacity on a populated memory channel. The memory capacity of both memory channels must be the same. x‘000’ 32 MB/memory channel (both memory channels must be populated in this case) x‘001’ 64 MB/memory channel x‘003’ 128 MB/memory channel x‘007’ 256 MB/memory channel x‘00F’ 512 MB/memory channel x‘01F’ 1 GB/memory channel x‘03F’ 2 GB/memory channel x‘07F’ 4 GB/memory channel x‘0FF’ 8 GB/memory channel x‘1FF’ 16 GB/memory channel x‘3FF’ 32 GB/memory channel **Note:** This is on a per memory channel basis.
10	ER_Dis	Early Read Disable Set this bit under the following conditions: • You are using XDR DRAMs that support Early Read, but you do not want to use it. • You are using XDR DRAMs that do not support Early Read. When Early Read is enabled, tPP-D = 1 (fixed).

Bits	Field Name	Description
11	Disable_APGen	Disable Address Parity Generation Controls the generation of three odd parity bits over the address of each command. When ECC is used, Dataflow includes these three address parity bits in its ECC generation and checking.
12:63	Reserved	Bits are not implemented; all bits read back zero.

7.3.3 MIC tRCD and Precharge Register n [n = 0,1] (MIC_Trcd_Pchg_n [n = 0,1])

After startup configuration, these values should not be changed by hardware or software. The MIC_Trcd_Pchg register contains values that determine when the first COL command for an access is issued, as well as when the Precharge command is issued. Timings for calibration commands are also included in this register. Both MIC_Trcd_Pchg registers (MIC_Trcd_Pchg_0 and MIC_Trcd_Pchg_1) must be set to the same value.

Register Short Name	MIC_Trcd_Pchg_0 MIC_Trcd_Pchg_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0D0’ x‘50A190’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



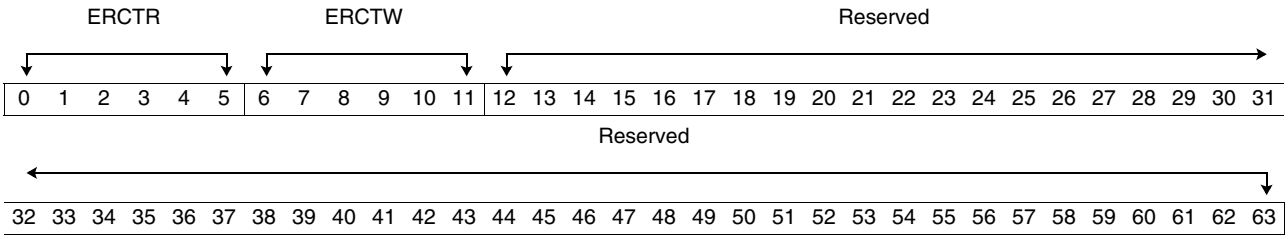
Bits	Field Name	Description
0:3	tRCD_R	tRCD-R value. 0010 If the tRCD-R value is 3 in the XDR DRAM datasheet, the controller uses a tRCD-R value of 3. 0100 If the tRCD-R value is 4 or 5 in the XDR DRAM datasheet, the controller uses a tRCD-R value of 5. 0110 If the tRCD-R value is 6 or 7 in the XDR DRAM datasheet, the controller uses a tRCD-R value of 7. 1000 If the tRCD-R value is 8 or 9 in the XDR DRAM datasheet, the controller uses a tRCD-R value of 9.
4:7	tRCD_W	tRCD-W value. 0010 If the tRCD-W value is less than or equal to 3 in the XDR DRAM datasheet, the controller uses a tRCD-W value of 3. 0100 If the tRCD-W value is 4 or 5 in the XDR DRAM datasheet, the controller uses a tRCD-W value of 5. 0110 If the tRCD-W value is 6 or 7 in the XDR DRAM datasheet, the controller uses a tRCD-W value of 7. 1000 If the tRCD-W value is 8 or 9 in the XDR DRAM datasheet, the controller uses a tRCD-W value of 9.
8:12	RPrecharge	Precharge for a read. This value is the greater of the following values: • tRAS - 1 • tRCD-R + 2 + tRDP - 1 if BL = 16; tRCD-R + tRDP - 1 if BL = 32 The minimum value is ‘00101’ (5).

Bits	Field Name	Description
13:17	WPrecharge	Precharge for a write. This value is the greater of the following values: • tRAS - 1 • tRCD-W + 2 + tWRP - 1 if BL = 16; tRCD-W + tWRP - 1 if BL = 32 The minimum value is ‘00101’ (5).
18:23	tPM0	tPM0 value. Set to tPM0 - 3. Then minimum value is ‘000101’ (5).
24:28	tPM_CALZ	tPM-CALZ value. Set to tPM-CALZ - 3. The minimum value is ‘00101’ (5).
29:33	tPM_CALC	tPM-CALC value. Set to tPM-CALC - 3. The minimum value is ‘00101’ (5).
34:38	tCALZE	tCALZE value. Set to tCALZE - 1. The minimum value is ‘00111’ (7).
39:43	tCALCE	tCALCE value. Set to tCALCE - 1. The minimum value is ‘00111’ (7).
44:63	Reserved	Bits are not implemented; all bits read back zero.

7.3.4 MIC Command Duration Register n [n = 0,1] (MIC_Cmd_Dur_n [n = 0,1])

After startup, these values should not be changed by hardware or software. This register contains values that determine the duration of various commands. Both MIC_Cmd_Dur registers must be set to the same value.

Register Short Name	MIC_Cmd_Dur_0 MIC_Cmd_Dur_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0D8’ x‘50A198’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



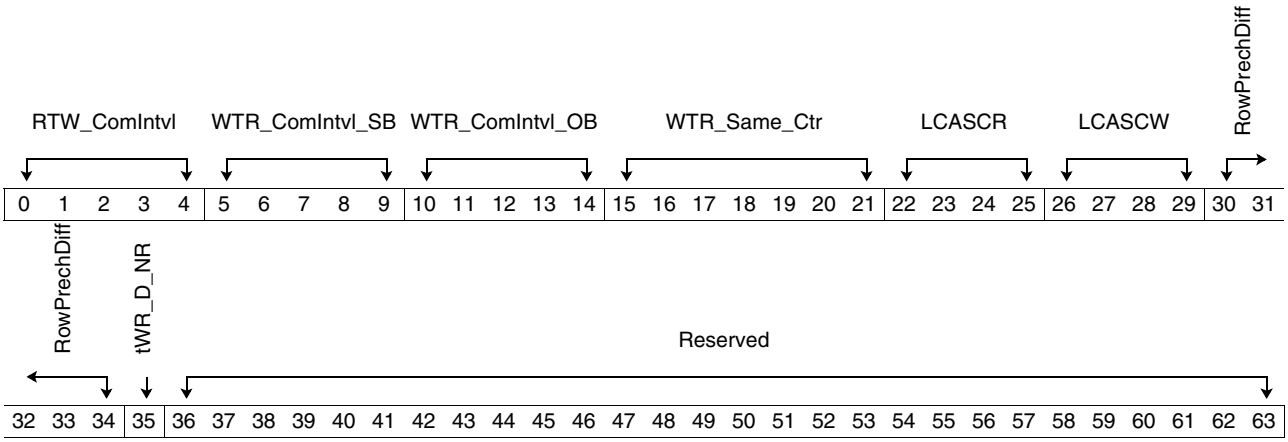
Bits	Field Name	Description
0:5	ERCTR	Effective row cycle time for a read. This value is the greatest of the following values: • tRC - 1 • tRAS + tRP - 1 • tRCD-R + 2 + tRDP + tRP - 1 for BL = 16; tRCD-R + tRDP + tRP - 1 for BL = 32 The minimum value is ‘000111’ (7). The maximum value is ‘111100’ (60).
6:11	ERCTW	Effective row cycle time for a write. This value is the greatest of the following values: • tRC - 1 • tRAS + tRP - 1 • tRCD-W + 2 + tWRP + tRP - 1 for BL = 16; tRCD-W + tWRP + tRP - 1 for BL = 32 The minimum value is ‘000111’ (7).
12:20	Reserved	Reserved. Software should ignore value read and write only zeros.
21:63	Reserved	Bits are not implemented; all bits read back zero.

7.3.5 MIC Command Spacing Register n [n = 0,1] (MIC_Cmd_Spc_n [n = 0,1])

This register contains values that control the spacing of reads and writes relative to one another. Both MIC_Cmd_Spc registers (MIC_Cmd_Spc_0 and MIC_Cmd_Spc_1) must be set to the same value. After startup, these values should not be changed by hardware or software.

Write-to-write, read-to-read, refresh-to-write, refresh-to-read, write-to-refresh, and read-to-refresh are determined by the tRR parameter of 4. Precharge-to-precharge is determined by the tPP parameter of 4 (1 for different bank sets when early read is enabled).

Register Short Name	MIC_Cmd_Spc_0 MIC_Cmd_Spc_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0E0’ x‘50A1A0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0:4	RTW_ComIntvl	Read-to-Write Command Interval When tCC = 2, tRCD-R + 2 + tΔRW - tRCD-W - 2 Else if tCC = 4, tRCD-R + tΔRW - tRCD-W - 2 Resulting values of 0 or 1 are treated as 2 because tRR,min is 4. If the resulting value is negative, enter zero (‘00000’). This value controls the minimum spacing between reads and writes being started. tΔRW includes the round-trip propagation delay.
5:9	WTR_ComIntvl_SB	Write-to-Read Command Interval, Same Bank Set/ER Disabled When tCC = 2, tRCD-W + 2 + tΔWR - tRCD-R - 2; Else if tCC = 4, tRCD-W + tΔWR - tRCD-R - 2 Resulting values of 0 or 1 are treated as 2 because tRR,min is 4. If the resulting value is negative, enter zero (‘00000’). This value controls the minimum spacing between writes and reads for the same bank sets. If Early Read is disabled, only this Write-to-Read value is used.

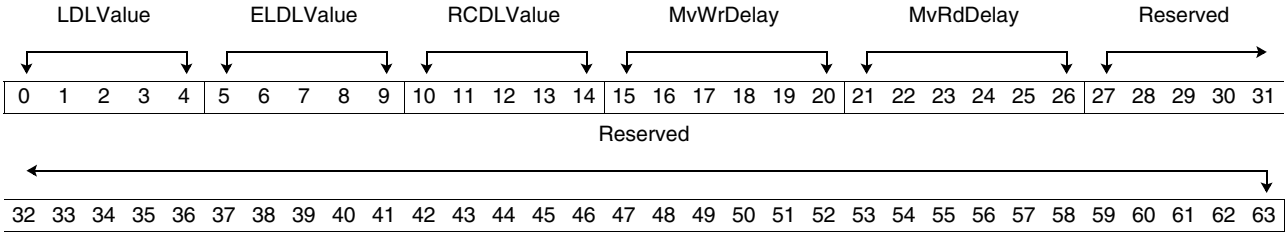
Bits	Field Name	Description
10:14	WTR_ComIntvl_OB	Write-to-Read Command Interval, Opposite Bank Set When tCC = 2, tRCD-W + 2 + tΔWR-D - tRCD-R - 2; Else if tCC = 4, tRCD-W + tΔWR-D - tRCD-R - 2 Resulting values of 0 or 1 are treated as 2 because tRR,min is 4. If the resulting value is negative, enter zero ('00000'). This value controls the minimum spacing between writes and reads for opposite bank sets. This value is only used when Early Read is enabled.
15:21	WTR_Same_Ctr	Write-to-Read Same Counter When tCC = 2, (tΔWR × 4) - 18 Else if tCC = 4, (tΔWR × 4) - 26 If this value is negative, enter zero ('0000000'). If Early Read is disabled, this value is a Don't Care. This assumes also that tRCD-W = tRCD-R. If this is not true, formula should be the results of MIC_Cmd_Spc_n [5:9] of this register × 4 - 18. For early reads, this is supposed to be true, therefore it should not be a concern.
22:25	LCASCR	Last CAS Counter for Read tRCD-R + 1 for Burst Length = 16 tRCD-R - 1 for Burst Length = 32 Controls when to switch to start a command at any time. Otherwise, avoids column commands.
26:29	LCASCW	Last CAS Counter for Write tRCD-W + 1 for Burst Length = 16 tRCD-W - 1 for Burst Length = 32 Controls when to switch to start a command at any time. Otherwise, avoids column commands.
30:34	RowPrechDiff	Precharge for a Write (WPrecharge) - Precharge for a Read (RPrecharge) - 2 If negative, enter zero ('00000') (See field descriptions in tRCD and Precharge Register description above.) Controls the Write Precharge Scoreboard location, so that the read precharges after a Write-to-Read turnaround in early read-after-write (ERAW) hardware are issued to meet tPP-D. tPP-D,min of 1 is supported.
35	tWR_D_NR	tΔWR-D Not Restricted Set this bit if the XDR DRAM has no restriction on the value of tΔWR-D between the values of tΔWR-D,min and tΔWR,min - 1. If the XDR DRAM has a restriction on the value of tΔWR-D, it is typically articulated in a footnote in the timing parameters section of the XDR DRAM spec. If there is a tΔWR-D restriction, then every other value after tΔWR-D,min until tΔWR,min is not allowed by the hardware. For example, if tΔWR-D,min = 2 and tΔWR,min = 9, then 2 is allowed for tΔWR-D, 3 is not, 4 is, 5 is not, 6 is, 7 is not, and 8 is.
36:37	Reserved	Reserved. Software should ignore value read and write only zeros.
38:63	Reserved	Bits are not implemented; all bits read back zero.

7.3.6 MIC Dataflow Control Register n [n = 0,1] (MIC_DF_Ctl_n [n = 0,1])

This register contains values that control the dataflow interface. After startup, these values should not be changed by hardware or software.

Both MIC_Df_Ctl registers, one for each half, must be set to the same value.

Register Short Name	MIC_DF_Ctl_0 MIC_DF_Ctl_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0E8’ x‘50A1A8’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0:4	LDLValue	Launch Delay Line Value. Set to tRCD-W + tQTD - 5.
5:9	ELDLValue	Expects Launch Delay Line Value. Set to tRCD-R + (tQRD - tERD) - 5.
10:14	RCDLValue	Read Complete Delay Line Value. Set to tQRD - 2.
15:20	MvWrDelay	Move Write Delay Delay before moving data out of the SRAM to the GRF. Set to [(tRCD-W + tQTD - 3) × 4] - 5 If negative, increase tRCD-W. This value has a different programming value and takes on a different meaning for Slow Core Mode.
21:26	MvRdDelay	Delay before moving expected read data out of the SRAM to the GRF. Set to [(tRCD-R + tQRD - tERD - 3) × 4] - 5 If negative, increase tRCD-R, following any aforementioned rules. This value has a different programming value for Slow Core Mode.
27:63	Reserved	Bits are not implemented; all bits read back zero.

7.4 Dataflow Registers

7.4.1 MIC Dataflow XIO PTCal Register n [n = 0,1] (MIC_XIO_PTCal_Data_n [n = 0,1])

These registers contains the XIO periodic timing calibration (PTCal) data and setup information. These registers are actually 128 bits wide, and are loaded by control bits [16:17] of the Dataflow Configuration Register. An entire cache line including ECC is 144 bytes. These 144 bytes are sent in groups of 9 in 16 MiClk periods. The 9 bytes in each group can be preset to receive the periodic timing calibration data (PTCal data) in register group A, B, or C.

During sixteen MiClk periods, data is taken from one of the three PTCal data slots. For periods 0 - 3, slot 0 is used; for periods 4 - 7, slot 1 is used; for periods 8 - 11, slot 2 is used; and for periods 12 - 15, slot 3 is used.

The net effect is that all DQ blocks receive the same pattern. Also, the pattern for each DQ pin on each block receives 4 bytes of data. For instance if a given pin was to receive data from group B, the output pattern would look like B0-B1-B2-B3: one byte for each of the four clock cycles.

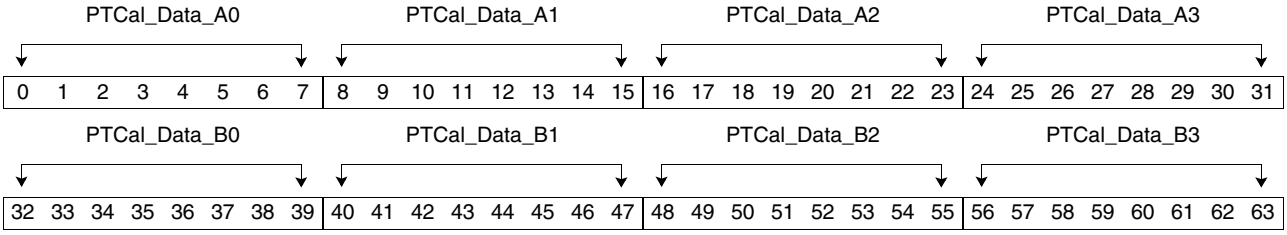
The TDATA bus from the MIC to XIO is ordered in big-endian notation: 71 to 0. The MIC fills the data in little-endian notation. Therefore, bit [0] (the MSB) of PTCal Data fields are actually sent last. For example, if slot B was set with the value B0-B1-B2-B3, the data sent over the pin would look like 0D-8D-4D-CD amounting to 32 bit times.

The following table shows the XIO DQ pin on which the selected byte is transmitted.

Selected Byte (A, B or C)	0	1	2	3	4	5	6	7	8
Output on DQ Pin	7	6	5	4	3	2	1	0	8

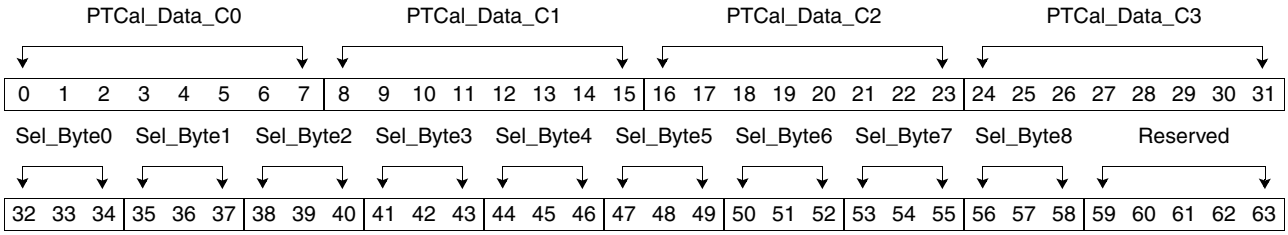
Register Short Name	MIC_XIO_PTCal_Data_0 MIC_XIO_PTCal_Data_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0F0’ x‘50A1B0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_DF macro)

First Half (bits [0:63])



Bits	Field Name	Description
0:7	PTCal_Data_A0	XIO PTCal data for slot A0
8:15	PTCal_Data_A1	XIO PTCal data for slot A1
16:23	PTCal_Data_A2	XIO PTCal data for slot A2
24:31	PTCal_Data_A3	XIO PTCal data for slot A3
32:39	PTCal_Data_B0	XIO PTCal data for slot B0
40:47	PTCal_Data_B1	XIO PTCal data for slot B1
48:55	PTCal_Data_B2	XIO PTCal data for slot B2
56:63	PTCal_Data_B3	XIO PTCal data for slot B3

Second Half (bits [64:127])



Bits	Field Name	Description
0:7	PTCal_Data_C0	XIO PTCal data for slot C0
8:15	PTCal_Data_C1	XIO PTCal data for slot C1
16:23	PTCal_Data_C2	XIO PTCal data for slot C2
24:31	PTCal_Data_C3	XIO PTCal data for slot C3
32:34	Sel_Byte0	Selection for Byte 0 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
35:37	Sel_Byte1	Selection for Byte 1 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
38:40	Sel_Byte2	Selection for Byte 2 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
41:43	Sel_Byte3	Selection for Byte 3 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
44:46	Sel_Byte4	Selection for Byte 4 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data

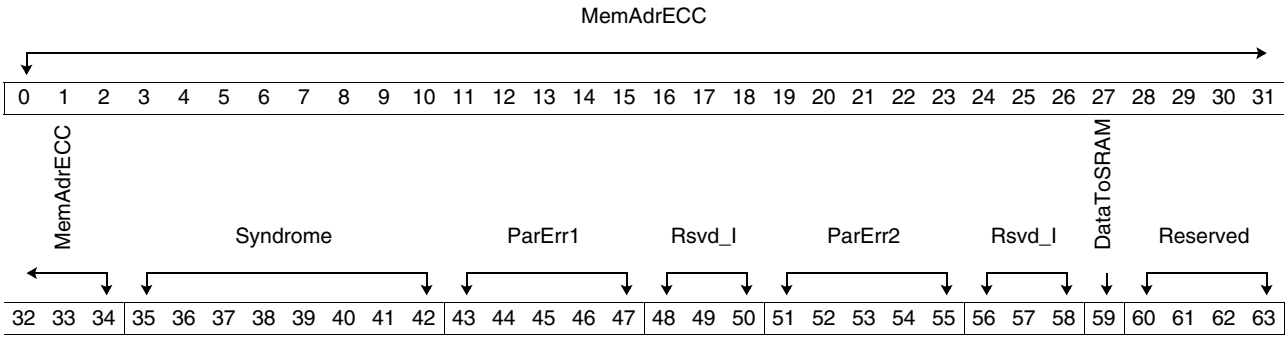
Bits	Field Name	Description
47:49	Sel_Byte5	Selection for Byte 5 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
50:52	Sel_Byte6	Selection for Byte 6 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
53:55	Sel_Byte7	Selection for Byte 7 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
56:58	Sel_Byte8	Selection for Byte 8 of PTCal data 100 Slot A data 010 Slot B data 001 Slot C data
59:63	Reserved	Bits not implemented; all bits read back zero.

TDATA[71:0] is defined in the XDR I/O Cell specification.

7.4.2 MIC Dataflow Error Correction Code Address Register n [n = 0,1] (MIC_Ecc_Addr_n [n = 0,1])

This register holds the memory address and syndrome for error correction code (ECC) errors.

Register Short Name	MIC_Ecc_Addr_0 MIC_Ecc_Addr_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A0F8’ x‘50A1B8’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_DF macro)



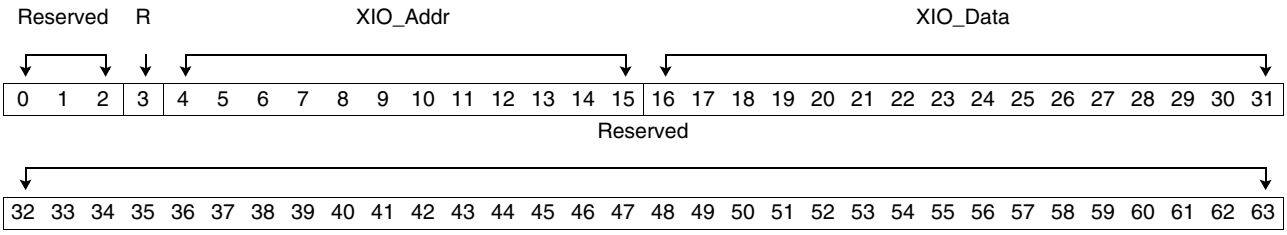
Bits	Field Name	Description
0:34	MemAdrECC	Memory address of ECC error
35:42	Syndrome	Syndrome for ECC error
43:47	ParErr1	Parity Error Address 1
48:50	Rsvd_I	Reserved. Latch bits are implemented; value read is the value written. Bits are reset to zero when an error is captured.
51:55	ParErr2	Parity Error Address 2
56:58	Rsvd_I	Reserved. Latch bits are implemented; value read is the value written. Bits are reset to zero when an error is captured.
59	DataToSRAM	Data to SRAM When this bit is set to a ‘1’, the MIC stores characterization data into a functional SRAM for debug. This is only used when MBL_Ctl[0] is set to a ‘1’. This bit causes data used in memory initialization to be stored in the MBL SRAM and await display alter to take it out. The entire MBL SRAM must be read out if you are debugging memory characterization.
60:63	Reserved	Bits are not implemented; all bits read back zero.

7.5 XDR DRAM Controller (YC) Registers

7.5.1 YRAC Data Register n [n = 0,1] (Yreg_YRAC_Dta_n [n = 0,1])

This register is used during the configuration of the memory channel by providing access to the XIO cell registers.

Register Short Name	Yreg_YRAC_Dta_0 Yreg_YRAC_Dta_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A100' x'50A140'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0:2	Reserved	Bits are not implemented; all bits read back zero.
3	R	Read 0 Writing this register causes an XIO cell register write. 1 This register is read in preparation for an XIO cell register write. A write to this register sets up the XIO cell address for a read. When this bit is set to '0', a write to this register causes an XIO cell register write.
4:15	XIO_Addr	XIO cell address [11:0]
16:31	XIO_Data	Value to be written or read [15:0]. The return value of these bits is determined by the underlying XIO cell register.
32:63	Reserved	Bits are not implemented; all bits read back zero.

To write an XIO cell register, make sure that Yreg_YRAC_Dta_n[3] = 0. This initiates a write to the XIO cell register specified in Yreg_YRAC_Dta_n[4:15]. The data to be written is in Yreg_YRAC_Dta_n[16:31].

To read an XIO cell register, a two-step process must be followed. First, the XIO cell register must be specified, so a write to this Yreg_YRAC_Dta register is performed with Yreg_YRAC_Dta_n[3] set to '1', Yreg_YRAC_Dta_n[4:15] specifying the XIO Cell register, and Yreg_YRAC_Dta_n[16:31] unused. A read to the Yreg_YRAC_Dta register is then performed, and the XIO cell register is accessed and the data returned in bits [16:31]. Reading the same XIO cell register can be performed over and over again. No other XIO cell or XDR DRAM accesses to the specified half are allowed during this operation. Each access to this register takes four PClks, which is nominally 10 ns.

Register accesses must be blocked during a register-reset operation. See the MIC_Yreg_Stat Register for information about detecting resets. All accesses are lost during this time period and the first one is captured. An error is recorded if this condition arises.

Software is responsible for the timing parameter tRESET_CMD. Eight reads of the MIC_Yreg_Stat Register after bits [18:19] of said register = ‘00’ guarantees that at least four PClks have transpired.

Hardware guarantees that XIO cell register accesses are blocked during calibration events, but these accesses complete even if there are back-to-back calibration events, as long as PAT_ENA gets back to zero.

REG_ADDR[11:00], REG_RD_DATA[15:0], REG_WR_DATA[15:0] are defined in the XDR I/O cell data specification.

Related Registers:

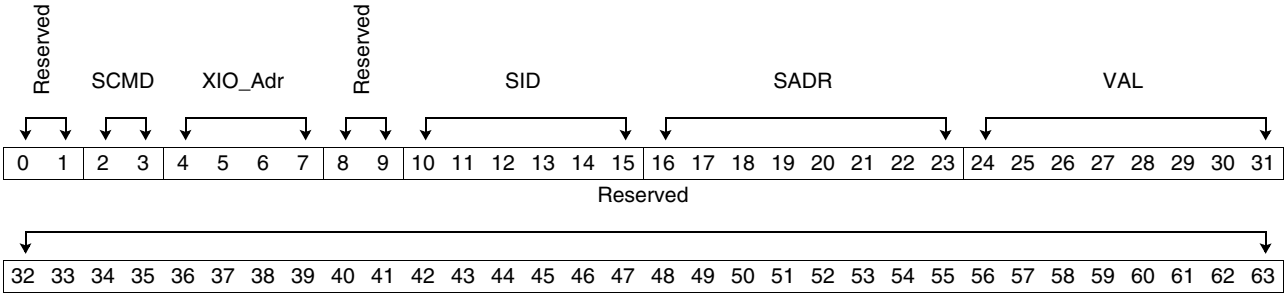
- Yreg_YDRAM_Dta
- MIC_Yreg_Stat
- Yreg_Init_Ctl

7.5.2 YDRAM Data Register n [n = 0,1] (Yreg_YDRAM_Dta_n [n = 0,1])

This register provides hardware assists to create the appropriate XIO cell register accesses that performs a serial XDR DRAM write access. This register is used during the configuration of the memory channel by providing write access XDR DRAM devices.

A write to this register with the SCMD[1:0] = SDW/SBW (Yreg_YDRAM_Dta[2] = 0) performs an XDR DRAM serial write transaction.

Register Short Name	Yreg_YDRAM_Dta_0 Yreg_YDRAM_Dta_1	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A108' x'50A148'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:3	SCMD	Serial Command SCMD[1:0] 00 SDW-Write to a particular device 01 SBW-Write to all the devices All other values are reserved.
4:7	XIO_Adr	XIO address bits REG_ADDR[11:8] for a register access to the RQ_SERIAL_CTL register. Use the value '0100' for this field. This addresses the independent RQ0 block.
8:9	Reserved	Bits are not implemented; all bits read back zero.
10:15	SID	Serial Chip ID: SID[5:0]
16:23	SADR	Serial Address: SADR[7:0]
24:31	VAL	Value to be written. SWD[7:0]. The return value depends on the XDR DRAM register.
32:63	Reserved	Bits are not implemented; all bits read back zero.

You cannot perform a serial read transaction using this register. A read returns the SCMD[1:0], SID[5:0], and SADR[7:0], and starts the serial transaction associated with these bits. This may not be desirable because XDR DRAM data field is unreliable. To perform a serial read, "Bit Bang" the RQ_SERIAL_CTL register of the XIO Cell with the correct sequence of commands and delay-to-account for the worst possible delay in the receive data for the number of XDR DRAM devices in the system.

Accessing this register initiates a series of 64-72 register reads and writes that send a scan command to the XDR DRAM chips. Only one XDR DRAM register access is allowed at any one time on any half. Also, no other XIO Cell or XDR DRAM register accesses are allowed during these operations.

An XIO Cell register access takes four PClks. Therefore, an XDR DRAM write requires a minimum of 256 PClks. A read to this register, while providing no useful information, requires a minimum of 288 PClks.

Register accesses must also be blocked during a register reset operation. See the MIC_Yreg_Stat status register for detecting resets. All read and write accesses during this time are lost. An error is recorded if this condition arises. The capture register describes the first access that was lost on each half, a read or a write.

The physical register implements only bits [2:23]. Only eight bits of data are written or read in each operation.

Software is responsible for the timing parameter tRESET_CMD. Performing eight successive reads of the MIC_Yreg_Stat Register after MIC_Yreg_Stat[18:19] returns ‘00’ guarantees that a minimum of four PClks have elapsed. This register uses the XIO Cell register access, thus requiring the minimum number of PClks.

Accesses to this register during normal operation can have an adverse effect on periodic calibration. Periodic calibrations wait until the XDR DRAM command has completed before starting. This effects tPEM and calibrations should be disabled before doing an XDR DRAM write if this parameter is of importance. Whenever PAT_ENA is ignored, normal read, writes, and refreshes occur without interruption.

Calibration events are held off during these operations.

REG_ADDR[11:8] is defined in the XDR I/O Cell specification.

SCMD[1:0], SID[5:0], SADR[7:0] and SWD[7:0] are defined in the XDR XDRAM specification.

Related Registers:
- Yreg_YRAC_Dta
- MIC_Yreg_Stat
- Yreg_Init_Ctl

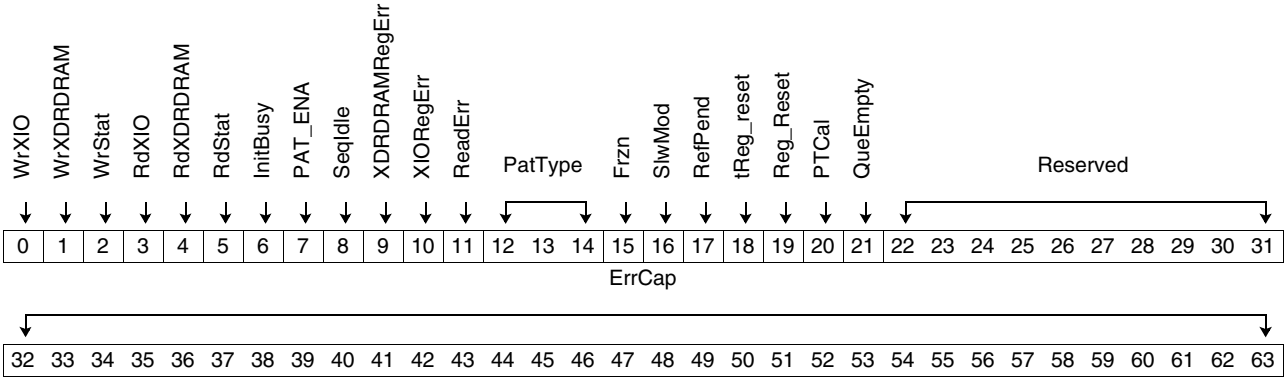
Additional Information: For more information, see the *XDR DRAM Datasheet*.

7.5.3 MIC Status Register n [n = 0,1] (MIC_Yreg_Stat_n [n = 0,1])

The register provides the status of XIO cell and XDR DRAM register accesses and other results. This register is used during the configuration of the memory channel by providing status during the initialization of the memory channel.

From an MMIO standpoint, you can check on the status of a XIO Cell or XDR DRAM write. An XIO cell write takes approximately 4 x 400 MHz cycles. XDR DRAM writes take 2 × 32 × 400 MHz cycles. XDR DRAM reads take (2 × 32 + 8) × 400 MHz cycles. Because of the nature of the byte bus, only one read can happen at a time. Therefore, a value of MIC_Yreg_Stat[0:7] = x‘41’ indicates that a XIO Cell write is still occurring. A value of MIC_Yreg_Stat[0:7] = x‘21’ indicates that an XDR DRAM access is occurring. A value of MIC_Yreg_Stat[0:7] = x‘04’ indicates the read of the status register.

Register Short Name	MIC_Yreg_Stat_0 MIC_Yreg_Stat_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A110’ x‘50A150’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0	WrXIO	Read only. Writing XIO. MMIO reads cannot see this bit set. 0 No access is occurring. 1 An XIO register access is occurring.
1	WrXDRDRAM	Read only. Writing XDR DRAM. 0 No XDR DRAM register access is occurring. 1 XDR DRAM register write is occurring. MMIO reads cannot see this bit set. Approximately 64 XIO cell register writes.
2	WrStat	Read only. Writing status. 0 A write is not occurring to this register. 1 Write of this register is occurring.
3	RdXIO	Read only. Reading XIO. 0 A read to an XIO register is not occurring. 1 An XIO register read access is occurring. **Note:** MMIO reads cannot see this bit set.

Bits	Field Name	Description
4	RdXDRDRAM	Read only. Reading XDR DRAM. 0 A read of the XDR DRAM is not occurring. 1 A read to the XDR DRAM register is occurring. **Note:** MMIO reads cannot see this bit set. XDR DRAM register reads using hardware acceleration are not supported.
5	RdStat	Read only. Reading status. 0 This register is not being read. 1 This register is being read. **Note:** This bit indicates that this register is being read.
6	InitBusy	Read only. Initialization sequencer busy. 0 The initialization sequencer is idle. 1 The initialization sequencer is busy.
7	PAT_ENA	Read only. PAT_ENA output of the XIO cell. 0 The XIO cell is not signaling a need for a calibration event. 1 The XIO cell is signaling a need for calibration.
8	SeqIdle	Read only. Sequencers are idle. 0 The main memory sequencers are not idle, they are busy. 1 The main memory sequencers are idle. There can be, however, commands in flight.
9	XDRDRAMRegErr	XDR DRAM write or read in progress with an XDR DRAM/XIO cell access or XIO cell reset occurring, error status. 0 No error has occurred during an XDR DRAM register access. 1 An error has occurred during an XDR DRAM register access.
10	XIORegErr	XIO cell write or read is in progress with an XDR DRAM or XIO cell access or XIO cell reset. This causes an error. 0 No error has occurred during an XIO cell register access. 1 An error has occurred during an XIO cell register access.
11	ReadErr	The error occurred on a read versus write. 0 If XIORegErr = 1 or XDRDRAMRegErr = 1, the access lost was a write. 1 If XIORegErr = 1 or XDRDRAMRegErr = 1, the access lost was a read.
12:14	PatType	Read only. PAT_TYPE[2:0]. 000 Initial or Periodic XIO CCAL/ZCAL is requested or is in progress. 001 Initial or Periodic XDR DRAM ZCAL is requested or is in progress. 010 Initial or Periodic XDR DRAM CCAL is requested or is in progress. 011 Periodic TCal is requested or is in progress. 100 Reserved. 101 Reserved. 110 Initial XIO RX TCAL is requested or is in progress. 111 Initial XIO TX TCAL is requested or is in progress.
15	Frzn	Read only. Frozen for initialization. 0 The MIC logic that drives the XIO cell interface is accepting commands. 1 The MIC logic that drives the XIO cell interface is not accepting commands.
16	SlwMod	Read only. Slow mode status. 0 The MIC is not in slow mode. 1 The MIC is in slow mode.
17	RefPend	Read only. Refresh pending. 0 There are no refreshes pending. 1 There are refreshes pending.

Bits	Field Name	Description
18	tReg_reset	Read only. The TCU is ordering a register reset. 0 The MIC is not being told to do an XIO cell register reset. 1 The TCU is telling the MIC to do an XIO cell register reset.
19	Reg_Reset	Read only. Reg_Reset. Waiting 200 PClks before deasserting Reg_Reset. 0 Reset complete. 1 Internal reset is in progress.
20	PTCal	Read only. Periodic timing calibration. If this bit is set to '1', this YC is responding to timing calibration.
21	QueEmpty	Read only. Queue empty. The store queues for this half are empty. No additional commands are coming. There is an asynchronous clock crossing to this register.
22:31	Reserved	Bits are not implemented; all bits read back zero.
32:63	ErrCap	Error capture. If MIC_Yreg_Stat[9] of this register is set to '1' , then Yreg_YDRAM_Dta[0:31] is copied to this location. If MIC_Yreg_Stat[10] is set to '1', then Yreg_YRAC_Dta[0:31] is copied to this location.

PAT_TYPE[2:0] is defined in the XDR I/O Cell specification.

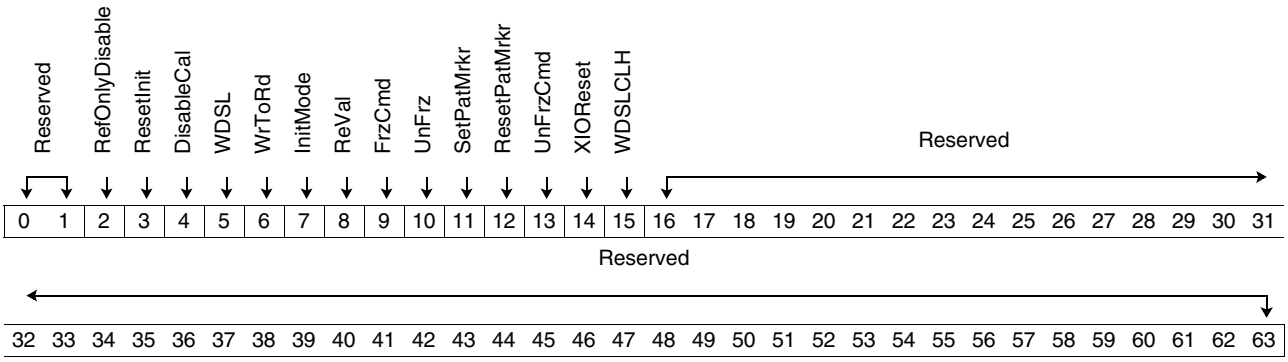
Related Registers:

- Yreg_YDRAM_Dta
- Yreg_YRAC_Dta
- Yreg_Init_Ctl
- Yreg_Init_Cnts

7.5.4 Initialization Control Register n [n = 0,1] (Yreg_Init_Ctl_n [n = 0,1])

This register controls the initialization sequencers. This register is used during the configuration of the memory channel by providing control of the initialization process.

Register Short Name	Yreg_Init_Ctl_0 Yreg_Init_Ctl_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A118' x'50A158'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2	RefOnlyDisable	Disable the Refresh Only function during initialization.
3	ResetInit	Reset initialization sequencer. (Lab only) 0 Do not reset the initialization sequencer. 1 Do a reset on the initialization sequencer. You must write a zero to return to normal mode.
4	DisableCal	Disables the sequencer aid in calibration. [make sequencer ignore pat_ena] (Lab Only) 0 Enable the initial calibration sequencer. 1 Disable the initial calibration sequencer. You must write a zero to return to normal mode.
5	WDSL	WDSL mode. Supports the Write Data Serial mode of the XDR DRAMs. 0 Normal command issuing mode. 1 Issue commands to support the WDSL sequence of the XDR initialization guide. You must write a zero to return to normal mode.
6	WrToRd	Write-to-read turns writes into reads with compares. (Lab only) 0 Normal command interpretation. 1 Writes are turned into reads with expect data. You must write a zero to return to normal mode.

Bits	Field Name	Description
7	InitMode	Initialization mode. Sent to the CTL partition during initial RX and TX calibration. 0 Normal execution mode. 1 Initial XIO RX and TX Calibration support. **Note:** To allow the control hardware to clean up its entries, wait at least 200 NClks from when this bit transitions from a '1' to a '0' at the end of calibration. No writes to the MIC MMIO space, main memory, Pervasive register space, or Token Manager should be sent during this time. If you have two halves of memory populated, you must write both Yreg_Init_Ctl registers sequentially with no other MMIO, read, or write between them. There needs to be 50 NClk cycles (in addition to the time between MMIO commands) between enabling InitMode on each half.
8	ReVal	Revalidates commands in the CTL partition. Read returns 0. (Lab only) 0 Do not revalidate the command queues. 1 Revalidate the command queues. Yreg_Init_Ctl[7] must be set to '1'. A single write causes this action.
9	FrzCmd	Freezes command issuing. Read returns 0. 0 Do not freeze the issuance of commands to the XIO cell. 1 Freeze issuing commands, including refreshes, to the XIO cell. You must write a zero to return to normal state.
10	UnFrz	Lab only. Unfreezes the take command interface. Read returns 0. 0 Do not unfreeze the command queue. 1 Unfreeze the command queue. A single write starts this action.
11	SetPatMrkr	Drives PAT_MRKR. Read returns 0. (Lab only) 0 Do not manually drive PAT_MRKR to the XIO cell. 1 Manual drive PAT_MRKR to the XIO cell. A single write starts this action.
12	ResetPatMrkr	Stop driving PAT_MRKR. Read returns 0. (Lab only) 0 Do not reset the driving of PAT_MRKR to the XIO cell. 1 Quit driving PAT_MRKR to the XIO cell. A single write starts this action.
13	UnFrzCmd	Unfreeze the command logic for any reason. Read returns 0. (Lab only) 0 Do not force the command logic to take new commands. 1 Force the command acceptance logic to take commands. A single write starts this action.
14	XIOReset	Cause a reg_reset to the XIO cell. If doing this as part of a soft reset, make sure this signal is asserted for at least the duration of an XDR DRAM register access. 0 Do not cause an XIO cell register reset. 1 Drive the XIO cell register signal. **Note:** Do not attempt a Yreg_YRAC_Dta access or a Yreg_YDRAM_Dta access during this time. All accesses are lost, the first access captured, and a recoverable error reported. You must write this bit back to zero to turn off refreshes.
15	WDSLCLH	Controls which half of a cache line is written during the XDR Write Data Serial Load (WDSL) pattern load step. This bit is effectively set to '0' when the WDSL Mode bit (Yreg_Init_Ctl[5]) is set to '0'. This bit should only be set when Burst Length (BL) = 16. 0 Access the first half (64 bytes) of a cache line. 1 Access the second half (64 bytes) of a cache line.
16:63	Reserved	Bits are not implemented; all bits read back zero.

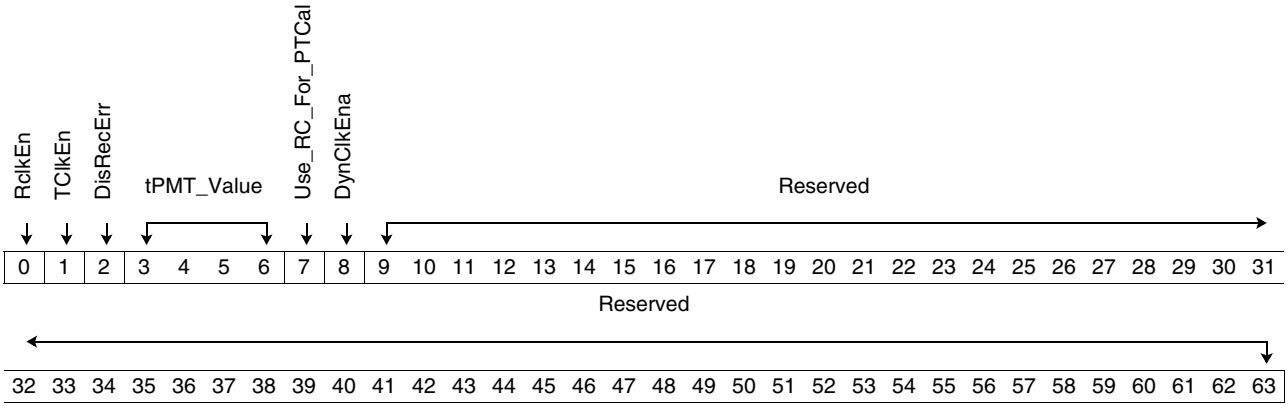
Related Registers:

- Yreg_YDRAM_Dta
- Yreg_YRAC_Dta
- MIC_Yreg_Stat
- Yreg_Init_Cnts

7.5.5 Initialization Constants Register n [n = 0,1] (Yreg_Init_Cnts_n [n = 0,1])

This configuration register controls certain constants that are used for initialization and for periodic timing calibration. Both Yreg_Init_Cnts registers (one for each half) must be set to the same value.

Register Short Name	Yreg_Init_Cnts_0 Yreg_Init_Cnts_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A120’ x‘50A160’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0	RclkEn	Receive Clock Enable drives the XIO cell pin. 0 Drive the RCLK_EN pin to a 0 (turn off receive). 1 Drive the RCLK_EN pin to a 1.
1	TClkEn	Transmit Clock Enable drives the XIO cell pin. 0 Drive the TCLK_EN pin to a 0 (turn off transmit). 1 Drive the TCLK_EN pin to a 1 During initialization and normal operation, this field is normally set to ‘1’. The type of clock enabling (constant or dynamic) is controlled by DynClkEna (Yreg_Init_Cnts[8]).
2	DisRecErr	Disables error reporting to the MIC FIR Register only. The Yreg_YDRAM_Dta Register still records the error. 0 Report any XIO cell or XDR DRAM register errors to the MIC_FIR Register. 1 Do not report any XIO cell or XDR DRAM register errors to the MIC_FIR Register.
3:6	tPMT_Value	tPMT - tRCD - W - tQTD - 2; if negative, enter ‘0000’.

Bits	Field Name	Description
7	Use_RC_For_PTCal	Use read complete for timing calibration. Only affects the read complete to dataflow. PTCal should be shut off by using XIO register writes before this bit is changed. PTCal can be turned on again after this bit has changed. Dataflow captures PTCal pattern data if it gets a read complete for a timing calibration. 0 Normal operation. 1 For debug purposes only.
8	DynClkEna	The dynamic clock gating function is only implemented for TClkEn. 0 Normal operation. Dynamic clock gating is not enabled. The transmit clock enable pin is driven to a constant 1. 1 Dynamic clock gating enabled. The transmit clock enable pin turns off for power savings when there is no activity on the memory interface. To take advantage of this power saving mode, Yreg_Init_Cnts[1] of this register should be set to '1' and Yreg_Init_Cnts[8] set to '1'. To disable this feature, set Yreg_Init_Cnts[8] to '0'. The system must not have any pending stores when Yreg_Init_Cnts[8] goes to '1' when enabling this mode. It is necessary to adhere to any other restrictions placed on these bits from the CBE XIO addendum to the XDR I/O Cell specification.
9:16	Reserved	Reserved. Software should ignore value read and write only zeros.
17:63	Reserved	Bits are not implemented; all bits read back zero.

Related Registers:

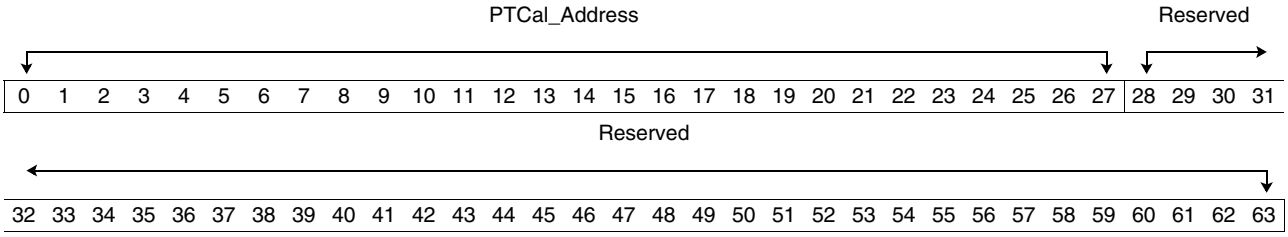
- MIC_Yreg_Stat

7.5.6 MIC Periodic Timing Calibration Address Register n [n = 0,1] (MIC_PTCal_Adr_n [n = 0,1])

The 28 bits of the MIC_PTCal_Adr Register specify the address of the cache line in memory to be set aside for periodic timing calibrations. When both memory channels are populated, there are two cache lines set aside for periodic timing calibrations. Typically, these two registers are set to the same value so that the two cache lines are contiguous in the system memory map. After startup, these values should not be changed by hardware or software.

The address is typically in the protected from scrub region (See MIC_Calibration_Addr_0 and MIC_Calibration_Addr_1).

Register Short Name	MIC_PTCal_Adr_0 MIC_PTCal_Adr_1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A130’ x‘50A170’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



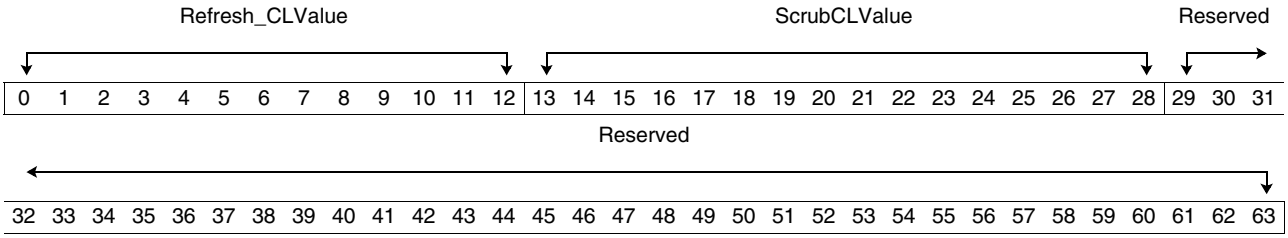
Bits	Field Name	Description
0:27	PTCal_Address	The 28 most significant real address bits of the periodic timing calibration cache line. Real (EIB) Address[28:55], if both memory channels are populated. Real (EIB) Address[29:56], if only one memory channel is populated. This value must fall within the specified range of memory.
28:63	Reserved	Bits are not implemented; all bits read back zero.

7.5.7 MIC Refresh and Scrub Register (MIC_Ref_Scb)

This register contains values that control the scheduling of refreshes and scrubs. If the MIC is in slow mode, the refresh or scrub rates cannot be increased to extreme values. If this happened, commands would be paced, and due to the slower NClk, the MIC would be unable to squeeze a command in between the refreshes or scrubs.

Note: After startup, these values should not be changed by hardware or software.

Register Short Name	MIC_Ref_Scb	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A200’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (MIC_CTL)

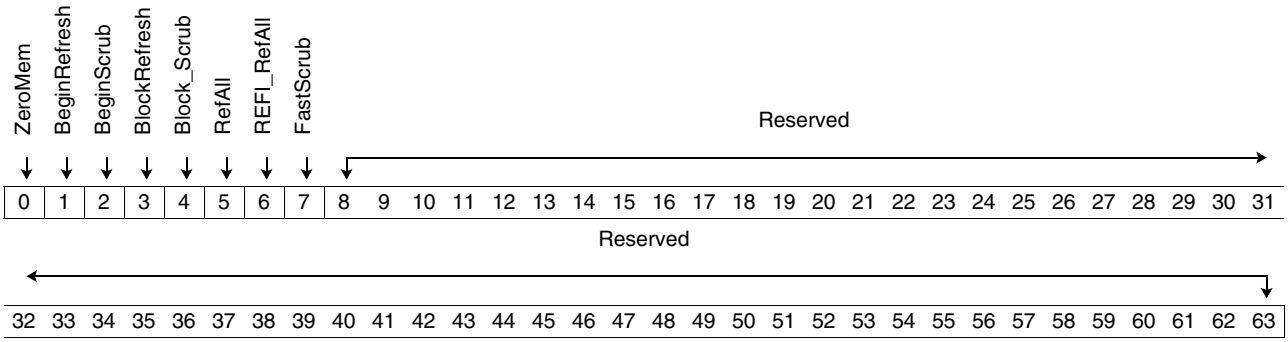


Bits	Field Name	Description
0:12	Refresh_CLValue	Refresh Counter Load Value. The value of bits [0:12] plus 1 multiplied by tCYCLE gives the period between Refresh commands to the XDR DRAMs. The minimum value allowed for this field is ‘1’. The hardware does not issue refreshes faster than the effective row cycle time of write or reads due to precharge command restrictions. The period between Refresh commands from the spec is tREF/(2^(# Bank Address bits + # Row Address Bits)).
13:28	ScrubCLValue	Scrub Counter Load Value. Assuming the 10 µs Timer is programmed to tick every 10 µs, the value of bits [13:28] plus 1 multiplied by 10 µs should equal the interval between scrubs. For example (and this is the maximum scrub interval supported), to scrub 64 MB every 2 days is equivalent to 262144 128-byte accesses per day, and the time per access is every 329.5 ms. This is the scrub interval. When defined as access time per 10 µs, 329.5 ms/10 µs = 32950 = x‘80B6’. Subtracting ‘1’ leaves the value x‘80B5’ to be written into the bits [13:28]. The minimum value allowed for this field is ‘1’.
29:63	Reserved	Bits are not implemented; all bits read back zero.

7.5.8 MIC Execute Register (MIC_Exc)

This register allows software to kick off and monitor more complex YC actions. After startup, these values can change. Software can write bits and hardware can clear certain bits.

Register Short Name	MIC_Exc	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A208’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



Bits	Field Name	Description
0	ZeroMem	Zero memory. Setting this bit initializes memory to all zeros. This bit may be set after the XDR DRAMs and XIO Cells have been initialized, and before normal operation. This bit automatically clears once all of memory is zeroed. Worst-case duration (zeroing two 32 GB halves in parallel) is approximately 2.68 s with nominal frequencies.
1	BeginRefresh	Begin refreshes. Software sets this bit when XIO cell and XDR DRAM initialization is complete. The rising edge starts scheduling refreshes. After this bit is written to ‘1’, it must always be written to ‘1’. Use MIC_Exc[3] to turn refreshes off and back on, if necessary.
2	BeginScrub	Begin scrubs. Software sets this bit when XIO cell and XDR DRAM initialization complete and scrubbing is needed. The rising edge starts scheduling scrubs. After this bit is written to ‘1’, it must always be written to ‘1’. Use MIC_Exc[3] to turn Scrubs off and back on, if necessary.
3	BlockRefresh	Block refresh pulses. Setting this bit does not block RefAll, an indication to refresh all banks sequentially, and is only used for initialization and for PDN entry and exit.
4	Block_Scrub	Block scrub pulses. Setting this bit does not block Fast Scrubbing.
5	RefAll	Refresh all. Setting this bit kicks off a refresh to each of the internal DRAM banks sequentially. Software may do this during initialization or during PDN entry or exit. Be aware that the RefAll does not start until the MicCtl read and write command queues are empty. This bit automatically clears once the last refresh of the sequence has been taken by YC. The last refresh is completed ~tRC after this event. For more precision, MIC_Yreg_Stat[8] can be monitored: when it becomes set following this event, the last refresh has been completed. Once the RefAll bit is set, no commands should be issued to the MIC until this bit clears (except, obviously, for the polling of this bit). See the ‘REFI at end of Ref All’ bit of this same register.

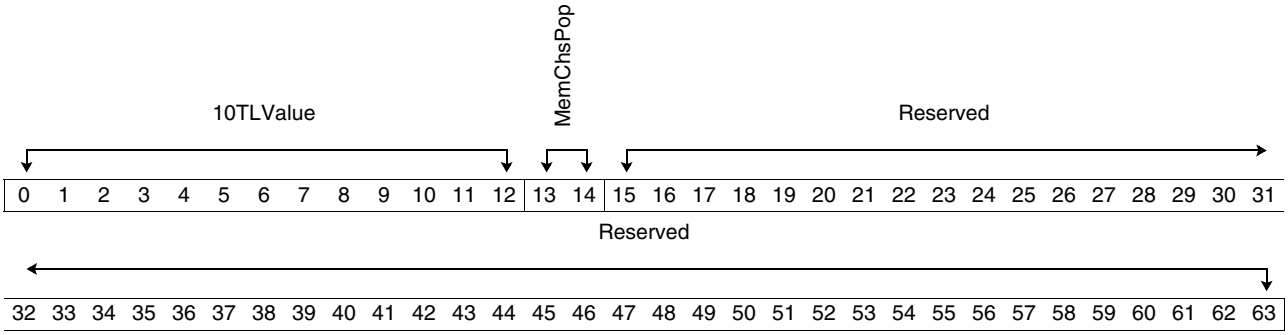
Bits	Field Name	Description
6	REFI_RefAll	REFI at end of Ref All. This bit should be written at the same time as the Ref All bit and not changed while Ref All is asserted. 0 No 1 Yes
7	FastScrub	Do Fast Scrub. Setting this bit scrubs all of memory as fast as possible. This is intended to be used after the XDR DRAMs are brought out of PDN, to correct single-bit errors that may have accumulated. This bit automatically clears once all of memory has been scrubbed.
8:63	Reserved	Bits are not implemented; all bits read back zero.

7.5.9 MIC Maintenance Config Register (MIC_Mnt_Cfg)

After startup, these values should not be changed by hardware or software.

This register supplements *Section 7.5.6 MIC Periodic Timing Calibration Address Register n [n = 0,1] (MIC_PTCal_Adr_n [n = 0,1])*.

Register Short Name	MIC_Mnt_Cfg	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A210'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_YC macro)



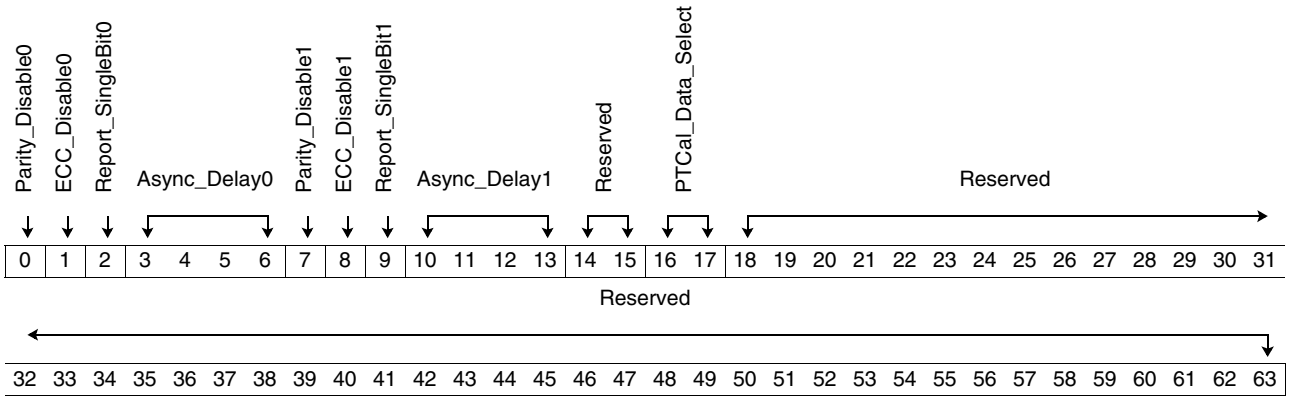
Bits	Field Name	Description
0:12	10TLValue	10-µsec Timer Load Value The 10-µsec timer is used to count scrub intervals. The minimum value allowed for this field is 1. (value of bits [0:12] + 1) × tCYCLE = 10 µsec
13:14	MemChsPop	Indicates which memory channel(s) is populated 00 Invalid 01 Only memory channel 0 is populated 10 Only memory channel 1 is populated 11 Both memory channels are populated **Note:** If a write to this register occurs that changes the MemChsPop field (call this X), another command to the MIC (read or write) cannot appear on the EIB until a minimum of 200 NClks after X appeared on the EIB. The Ctl partition requires this delay. Used to direct maintenance outputs to one or both halves of MIC, and to filter maintenance inputs from one or both halves of MIC.
15:63	Reserved	Bits are not implemented; all bits read back zero.

7.6 Dataflow (DF) Registers

7.6.1 MIC Dataflow Configuration Register (MIC_DF_Config)

This register sets user preferences for error checks and some timing for special modes.

Register Short Name	MIC_DF_Config	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A218'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_DF macro)



Bits	Field Name	Description
0	Parity_Disable0	Disable parity reporting for XIO0. 0 Enables parity checking when capturing auxiliary trace data. Also used for normal functional mode. 1 Disables parity reporting for XIO0.
1	ECC_Disable0	Disable ECC Off for XIO0 (includes correction and reporting).
2	Report_SingleBit0	Single bit or multibit error reporting for XIO0. 0 Multibit only. 1 Single or multibit.

Bits	Field Name	Description
3:6	Async_Delay0	Asynchronous read delay for XDR I/O Cell0. 0000 If MiClk/NClk ≥ 0.50 0001 If MiClk/NClk ≥ 0.45 0011 If MiClk/NClk ≥ 0.40 0101 If MiClk/NClk ≥ 0.35 0111 If MiClk/NClk ≥ 0.30 1111 If in slow memory mode (PLL BYPASS) **Note:** For results greater than 15, set this register to 15. For negative values, set this register to 0. When the NClk frequency changes for power management slow(n) mode, this register does not need to be changed. In other words, set this using the highest NClk frequency. For ratios not shown, use the following equation to determine the register value: delay = 18 - 38 × (MiClk/NClk) It is always safe to program this register to '1111'. A small read latency impact can be observed if the value '1111' is used. This value has a different programming value for slow core mode.
7	Parity_Disable1	Disable parity reporting for XIO1. 0 Enables parity checking when capturing auxiliary trace data. Also used for normal functional mode. 1 Disables parity reporting for XIO1.
8	ECC_Disable1	Disable ECC Off for XIO1 (includes correction and reporting).
9	Report_SingleBit1	Single bit or multibit error reporting for XIO1. 0 Multibit only. 1 Single or multibit.
10:13	Async_Delay1	Asynchronous read delay for XDR I/O Cell1 0000 If MiClk/NClk ≥ 0.50 0001 If MiClk/NClk ≥ 0.45 0011 If MiClk/NClk ≥ 0.40 0101 If MiClk/NClk ≥ 0.35 0111 If MiClk/NClk ≥ 0.30 1111 If in slow memory mode (PLL BYPASS). **Note:** For results greater than 15, set this register to 15. For negative values, set this register to 0. When the NClk frequency changes for power management slow(n) mode, this register does not need to be changed. In other words, set this using the highest NClk frequency. For MiClk frequencies lower than 1.2 GHz, or for combinations in which the NClk frequency is not 4 GHz, use the following equation to determine the register value: delay = 18 - 38 × (MiClk/NClk) It is always safe to program this register to '1111' A small read latency impact can be observed if the value '1111' is used. This value has a different programming value for slow core mode.
14:15	Reserved	Bits are not implemented; all bits read back zero.
16:17	PTCal_Data_Select	Selects which of the two halves of the MIC Dataflow XIO PTCal registers to write: bit 16: 0 write the first half of MIC_XIO_PTCal_Data_0 (memory channel 0) 1 write the second half of MIC_XIO_PTCal_Data_0 (memory channel 0) bit 17: 0 write the first half of MIC_XIO_PTCal_Data_1 (memory channel 1) 1 write the second half of MIC_XIO_PTCal_Data_1 (memory channel 1) See the *MIC Dataflow XIO PTCal Register n [n = 0,1] (MIC_XIO_PTCal_Data_n [n = 0,1])* for more information.
18:63	Reserved	Bits are not implemented; all bits read back zero.

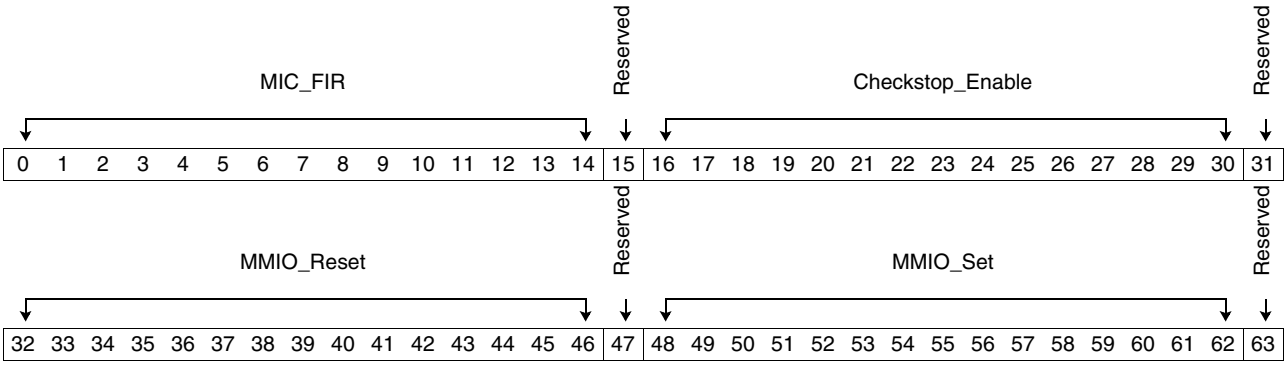
7.7 MIC Fault Isolation and Checkstop Enable Registers

7.7.1 MIC FIR/Checkstop Enable/Reset/Set Registers (MIC_FIR)

This section defines the MIC Fault Isolation Registers (FIRs). Because the reset, set, and direct write of the FIR are all in the same register, special care must be taken to ensure the correct function occurs. The following examples assume a checkstop enable setting of '1111_1101_0100_000':

- To direct write the FIR, the MMIO_Reset field must be all ones (1) and the MMIO_Set field must be all zeros (0). For a direct write of '1111_1111_1111_111', the register write is x'FFFEFD40_FFFE0000'.
- To use the MMIO_Reset field, the direct write and MMIO_Set fields should be all zeros (0). To reset the register using '0000_0000_0000_000', the register write is x'0000FD40_00000000'.
- To use the MMIO_Set field, the direct write field must be all zeros (0) and the MMIO_Reset field must be all ones (1), since this field is used in the feedback path from the FIR. To Set in '1111_1111_1111_111' using the MMIO_Set field, the register write is x'0000FD40_FFFEFFFE'.

Register Short Name	MIC_FIR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50A230’	**Memory Map Area**	MIC and TKM
Value at Initial POR	x‘00000000_FFFE0000’	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC



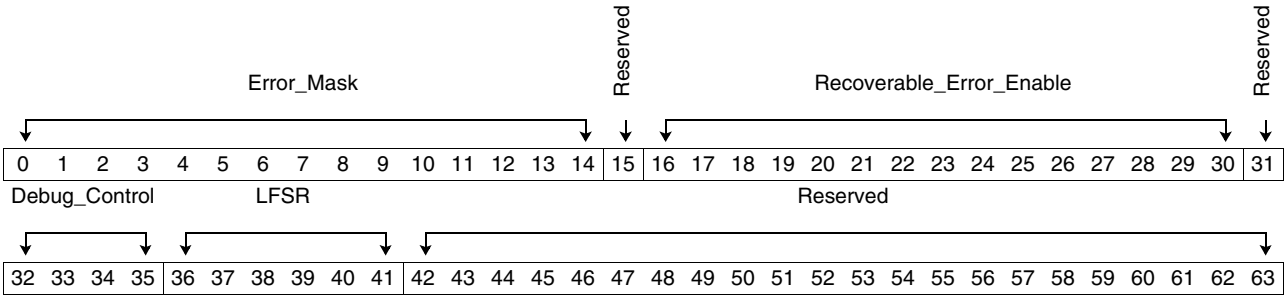
Bits	Description	Recommended Mask Settings	
		Error Mask	Checkstop Enable
0	Write SRAM parity error 0 (checkstop)	0	1
1	Write SRAM parity error 1 (checkstop)	0	1
2	Read SRAM parity error 0 (checkstop)	0	1
3	Read SRAM parity error 1 (checkstop)	0	1
4	Read SRAM parity error 2 (checkstop)	0	1
5	Data error (DERR) received on write data (checkstop)	0	1
6	Read data single-bit error correction code (ECC) error 0 (recoverable)	0	0

Bits	Description	Recommended Mask Settings	
		Error Mask	Checkstop Enable
7	Read data multiple-bit ECC error 0 (checkstop)	0	1
8	Read data single-bit ECC error 1 (recoverable)	0	0
9	Read data multiple-bit ECC error 1 (checkstop)	0	1
10	Byte bus master write error (recoverable)	0	0
11	Byte bus master read error (recoverable)	0	0
12	Rambus Register overflow 0 (recoverable)	0	0
13	Rambus Register overflow 1 (recoverable)	0	0
14	Auxiliary trace overflow error (recoverable)	0	0

Bits	Field Name	Description
0:14	MIC_FIR	Direct set of FIR. See previous table for errors corresponding to each bit. To directly set the FIR, set this field to the desired FIR value; and set the MMIO_Reset field to all ones and the MMIO_Set field to all zeros.
15	Reserved	Bit is not implemented; bit reads back zero.
16:30	Checkstop_Enable	Directs certain errors in the FIR to cause a checkstop. If the error is not a checkstop, it is considered recoverable.
31	Reserved	Bit is not implemented; bit reads back zero.
32:46	MMIO_Reset	MMIO reset is used to reset the FIR. x‘0000’ Resets the bit. x‘0001’ Maintains the original setting of the bit. **Note:** For this field to work properly, the FIR and MMIO_Set fields must be written with all zeros.
47	Reserved	Bit is not implemented; bit reads back zero.
48:62	MMIO_Set	MMIO set is used to set the FIR. x‘0000’ Maintains the original setting of the bit. x‘0001’ Sets the bit. **Note:** For this field to work properly, the FIR field must be written with all zeros and the MMIO_Reset field must be written with all ones.
63	Reserved	Bit is not implemented; bit reads back zero.

7.7.2 MIC ErrorMask/RecErrorEnable/Debug Control Register (MIC_FIR_Debug)

Register Short Name	MIC_FIR_Debug	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50A238'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MIC (YM_DF)



Bits	Field Name	Description
0:14	Error_Mask	Masks bits in FIR so that they do not cause errors (recoverable or checkstop). x'0000' Allows the error to be reported as a checkstop or recoverable. x'0001' Blocks the reporting of the error; however, the FIR bit is still set. See the MIC_FIR table for errors corresponding to each of the 15 bits.
15	Reserved	Bit is not implemented; bit reads back zero.
16:30	Recoverable_Error_Enable	Directs certain errors in the FIR to be counted in the Linear Feedback Shift Register (LFSR). The Checkstop Enable field of the MIC_FIR is used to set which errors cause a recoverable error. See table for the MIC_FIR for errors corresponding to each of the 15 bits.
31	Reserved	Bit is not implemented; bit reads back zero.
32:35	Debug_Control	Control for trace bus
36:41	LFSR	Direct set or reset of the LFSR counter
42:63	Reserved	Bits are not implemented; all bits read back zero.

8. Token Manager (TKM) MMIO Registers

This section describes the TKM memory map and lists the TKM registers. The shared MIC and TKM memory space starts at x‘50A000’ and ends at x‘50AFFF’. Offsets are from the start of the shared MIC and TKM register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

8.1 TKM MMIO Memory Map

Table 8-1. TKM MMIO Memory Map

Hexadecimal Offset (x‘50A*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x‘FC0’	Reserved			
x‘FC8’	*TKM Memory Bank Allocation Register (TKM_MBAR)*	64	R/W	*Section 8.2.1* on page 208
x‘FD0’	*TKM IOIF0 Allocation Register (TKM_IOIF0_AR)*	64	R/W	*Section 8.2.2* on page 210
x‘FD8’	*TKM IOIF1 Allocation Register (TKM_IOIF1_AR)*	64	R/W	*Section 8.2.3* on page 212
x‘FE0’	*TKM Priority Register (TKM_PR)*	64	R/W	*Section 8.2.4* on page 214
x‘FE8’	*TKM Interrupt Status Register (TKM_IS)*	64	R/W	*Section 8.2.5* on page 217
x‘FF0’	*TKM Performance Monitor Control Register (TKM_PMCR)*	64	R/W	*Section 8.2.6* on page 218

8.2 TKM MMIO Register Descriptions

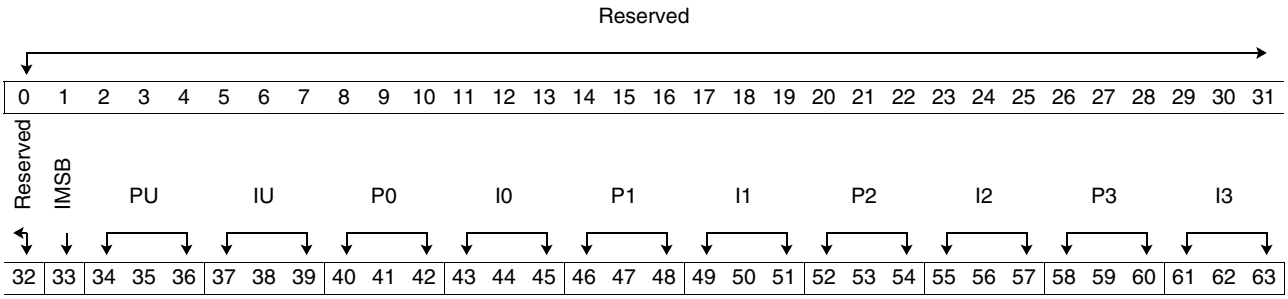
8.2.1 TKM Memory Bank Allocation Register (TKM_MBAR)

TKM Memory Bank Allocation Register defines the number of EIB clock cycles between the generation of memory tokens for each Research Allocation Group (RAG). This period is defined by a P*r* field and an I*r* field for each RAG *r*, where *r* is an element of {U, 0, 1, 2, 3}, and by the IMSB field. The number of EIB clock cycles between subsequent generation of memory bank tokens for RAG *r* is $(I + 1 + 8 \times ((P>0)\ OR\ TKM\_MBAR[33]) \times 2^{P})$ where I is the value of the I*r* field and P is the value of the P*r* field; *r* is an element of {U, 0, 1, 2, 3}.

Eight plus the value of an I*r* field is loaded into the RAG *r* memory rate decrementer when the Memory Bank Allocation Register 0 is written or when the RAG *r* memory rate decrementer expires. When a rate decrementer is zero and is to be decremented again, the rate decrementer expires. Not only does the rate decrementer have to be zero, but it also must be decremented again, at which time it is loaded again based on the allocation. The memory rate decrementer is decremented one out of 2^{P} EIB clocks. The RAG's memory rate decrementer is used to generate tokens if enabled by the corresponding TKM_CR Register Rate Counter bit.

The valid values for the P*r* field are decimal 0 to 7.

Register Short Name	TKM_MBAR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50AFC8'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TKM



Bits	Field Name	Description
0:32	Reserved	Bits are not implemented; all bits read back zero.
33	IMSB	Interval most significant bit. This bit allows higher allocation rates for one RAG. 0 When a RAG's memory rate decrementer would decrement below zero, the most significant bit of the 4-bit interval value is based on the RAG's prescaler value: --If the RAG's prescaler value is zero, a '0' is loaded. Only one of a RAG's prescaler values can be zero. Otherwise, memory is over-allocated and tokens may be lost. • If the RAG's prescaler value is nonzero, a '1' is loaded. 1 When a RAG's memory rate decrementer would decrement below zero, a '1' is loaded into the most significant bit of the 4-bit interval value loaded into each RAG's memory rate decrementer.
34:36	PU	Prescaler for RAG U, where RAG U Memory Rate Decrementer is decremented every one out of 2^{PU} EIB clock cycles, or NClk/2.

Bits	Field Name	Description
37:39	IU	Least significant 3 bits of the 4-bit interval value loaded into the RAG U Memory Rate Decrementer when the Memory Rate Decrementer U would decrement below zero.
40:42	P0	Prescaler for RAG 0, where RAG 0 Memory Rate Decrementer is decremented once every 2^{P0} EIB clock cycles.
43:45	I0	Least significant 3 bits of the 4-bit interval value loaded into the RAG 0 Memory Rate Decrementer when the Memory Rate Decrementer 0 would decrement below zero.
46:48	P1	Prescaler for RAG 1, where RAG 1 Memory Rate Decrementer is decremented once every 2^{P1} EIB clock cycles.
49:51	I1	Least significant 3 bits of the 4-bit interval value loaded into the RAG 1 Memory Rate Decrementer when the Memory Rate Decrementer 1 would decrement below zero.
52:54	P2	Prescaler for RAG 2, where RAG 2 Memory Rate Decrementer is decremented once every 2^{P2} EIB clock cycles.
55:57	I2	Least significant 3 bits of the 4-bit interval value loaded into the RAG 2 Memory Rate Decrementer when the Memory Rate Decrementer 2 would decrement below zero.
58:60	P3	Prescaler for RAG 3, where RAG 3 Memory Rate Decrementer is decremented once every 2^{P3} EIB clock cycles.
61:63	I3	Least significant 3 bits of the 4-bit interval value loaded into the RAG 3 Memory Rate Decrementer when the Memory Rate Decrementer 3 would decrement below zero.

Programming Note: The following sequence must be performed for the bank id counters for each RAG to be initialized correctly when changing TKM_MBAR.

1. MMIO Write TKM_CR, disable all Ri bits.
2. MMIO Write TKM_MBAR, set all to x‘0’.
3. MMIO Write TKM_MBAR, set new allocation values.
4. MMIO Write TKM_CR, reenable Ri bits of selected RAGs.

Also, for each type of token allocation (Memory, IOIF0In, IOIF0Out, IOIF1In, IOIF1Out), the relationship of token allocation rates between any two RAGs can prevent the sharing of tokens to the RAG with one requester. When X is (U,0,1,2,3), Y is (0,1,2,3), and RAGY has one requestor in its group, then the allocated percentage for RAGY should not be an intregral multiple of that for RAGX when RAGX is enabled to share its tokens with RAGY.

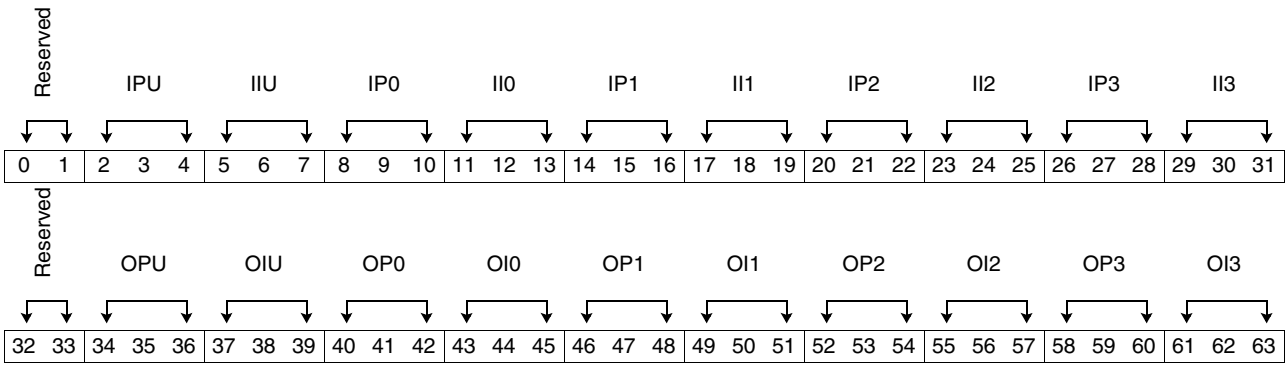
8.2.2 TKM IOIF0 Allocation Register (TKM_IOIF0_AR)

TKM IOIF0 Allocation Register defines the number of EIB clock cycles between the generation of IOIF0 In bus tokens and between IOIF0 Out bus tokens for each RAG. For IOIF0 In, this period is defined by a IPr field and an IIr field for each RAG *r*. For IOIF0 Out, this period is defined by a OPr field and an OIr field for each RAG r, where *r* is an element of {U, 0, 1, 2, 3}. The number of EIB clock cycles between subsequent generation of IOIF0 In tokens for RAG *r* is $(9 + I) \times 2^P$, where *I* is the value of the IIr field and *P* is the value of the IPr field; *r* is an element of {U, 0, 1, 2, 3}. Likewise, the number of EIB clock cycles between subsequent generation of IOIF0 Out tokens for RAG *r* is $(9 + I) \times 2^P$, where *I* is the value of the OIr field and *P* is the value of the OPr field; *r* is an element of {U, 0, 1, 2, 3}.

Eight plus the value of an IIr is loaded into RAG *r* IOIF0 In rate decrementer when the RAG *r* IOIF0 In rate decrementer expires. When a rate decrementer is zero and is to be decremented again, the rate decrementer expires. Not only does the rate decrementer have to be zero, but it also must be decremented again, at which time it is loaded again based on the allocation. The RAG *r* IOIF0 In rate decrementer is decremented one out of 2^P EIB clocks, where *P* is the value of the Prescaler field. The RAG *r* IOIF0 In rate decrementer is used to generate tokens if enabled by the corresponding TKM_CR Register RAG *r* Rate Counters Enable bit. The RAG *r* IOIF0 Out rate decrementer functions in a similar manner using the OIr and OPr fields.

Valid values for the prescaler fields are 0 to 7.

Register Short Name	TKM_IOIF0_AR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50AFD0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TKM



Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:4	IPU	IOIF0 In Prescaler for RAG U, where RAG U IOIF0 In Rate Decrementer is decremented once every 2^{IPU} EIB clock cycles.
5:7	IIU	Interval for RAG U, where (IIU + 8) is loaded into the RAG U IOIF0 In Rate Decrementer when RAG U IOIF0 In Rate Decrementer would be decremented below zero.
8:10	IP0	IOIF0 In Prescaler for RAG 0, where RAG 0 IOIF0 In Rate Decrementer is decremented every one out of 2^{IP0} EIB clock cycles

Bits	Field Name	Description
11:13	II0	Interval for RAG 0, where (II0 + 8) is loaded into the RAG 0 IOIF0 In Rate Decrementer when RAG 0 IOIF0 In Rate Decrementer would be decremented below zero.
14:16	IP1	IOIF0 In Prescaler for RAG 1, where RAG 1 IOIF0 In Rate Decrementer is decremented every one out of 2^{IP1} EIB clock cycles
17:19	II1	Interval for RAG 1, where (II1 + 8) is loaded into the RAG 1 IOIF0 In Rate Decrementer when RAG 1 IOIF0 In Rate Decrementer would be decremented below zero.
20:22	IP2	IOIF0 In Prescaler for RAG 2, where RAG 2 IOIF0 In Rate Decrementer is decremented every one out of 2^{IP2} EIB clock cycles
23:25	II2	Interval for RAG 2, where (II2 + 8) is loaded into the RAG 2 IOIF0 In Rate Decrementer when RAG 2 IOIF0 In Rate Decrementer would be decremented below zero.
26:28	IP3	IOIF0 In Prescaler for RAG 3, where RAG 3 IOIF0 In Rate Decrementer is decremented every one out of 2^{IP3} EIB clock cycles
29:31	II3	Interval for RAG 3, where (II3 + 8) is loaded into the RAG 3 IOIF0 In Rate Decrementer when RAG 3 IOIF0 In Rate Decrementer would be decremented below zero.
32:33	Reserved	Bits are not implemented; all bits read back zero.
34:36	OPU	IOIF0 Out Prescaler for RAG U, where RAG U IOIF0 Out Rate Decrementer is decremented every one out of 2^{OPU} EIB clock cycles
37:39	OIU	Interval for RAG U, where (OIU + 8) is loaded into the RAG U IOIF0 Out Rate Decrementer when RAG U IOIF0 Out Rate Decrementer would be decremented below zero.
40:42	OP0	IOIF0 Out Prescaler for RAG 0, where RAG 0 IOIF0 Out Rate Decrementer is decremented every one out of 2^{OP0} EIB clock cycles
43:45	OI0	Interval for RAG 0, where (OI0 + 8) is loaded into the RAG 0 IOIF0 Out Rate Decrementer when RAG 0 IOIF0 Out Rate Decrementer would be decremented below zero.
46:48	OP1	IOIF0 Out Prescaler for RAG 1, where RAG 1 IOIF0 Out Rate Decrementer is decremented every one out of 2^{OP1} EIB clock cycles
49:51	OI1	Interval for RAG 1, where (OI1 + 8) is loaded into the RAG 1 IOIF0 Out Rate Decrementer when RAG 1 IOIF0 Out Rate Decrementer would be decremented below zero.
52:54	OP2	IOIF0 Out Prescaler for RAG 2, where RAG 2 IOIF0 Out Rate Decrementer is decremented every one out of 2^{OP2} EIB clock cycles
55:57	OI2	Interval for RAG 2, where (OI2 + 8) is loaded into the RAG 2 IOIF0 Out Rate Decrementer when RAG 2 IOIF0 Out Rate Decrementer would be decremented below zero.
58:60	OP3	IOIF0 Out Prescaler for RAG 3, where RAG 3 IOIF0 Out Rate Decrementer is decremented every one out of 2^{OP3} EIB clock cycles
61:63	OI3	Interval for RAG 3, where (OI3 + 8) is loaded into the RAG 3 IOIF0 Out Rate Decrementer when RAG 3 IOIF0 Out Rate Decrementer would be decremented below zero.

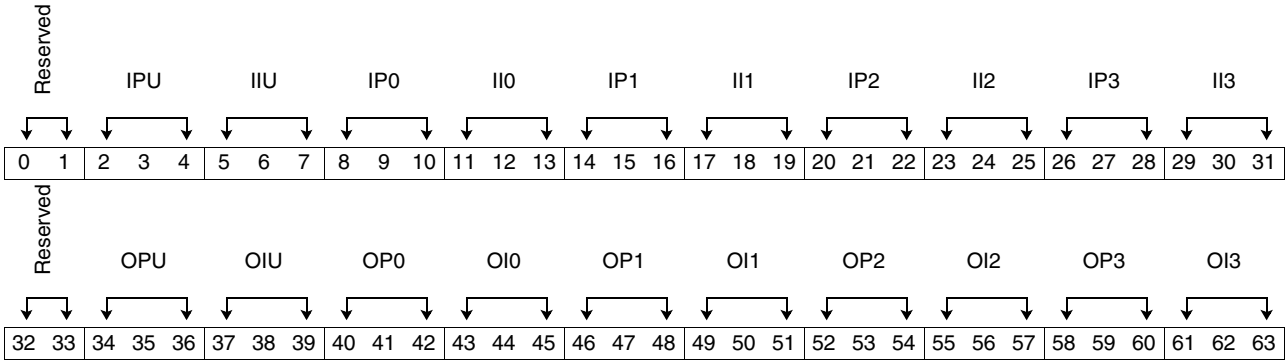
8.2.3 TKM IOIF1 Allocation Register (TKM_IOIF1_AR)

TKM IOIF1 Allocation Register is similar to TKM IOIF0 Allocation Register except that it is used to manage the IOIF1 In and IOIF1 Out bus and there is an additional implied prescaler. The number of EIB clock cycles between subsequent generation of IOIF1 In tokens for RAG *r* is (9 + *I*) x $2^{(P+2)}$, where *I* is the value of the IIr field and *P* is the value of the IPr field; *r* is an element of {U, 0, 1, 2, 3}. Likewise, the number of EIB clock cycles between subsequent generation of IOIF1 Out tokens for RAG *r* is (9 + *I*) x $2^{(P+2)}$, where *I* is the value of the OIr field and *P* is the value of the OPr field; *r* is an element of {U, 0, 1, 2, 3}.

Eight plus the value of an IIr is loaded into RAG *r* IOIF1 In rate decrementer when the RAG *r* IOIF1 In rate decrementer expires. When a rate decrementer is zero and is to be decremented again, the rate decrementer expires. Not only does the rate decrementer have to be zero, but it also must be decremented again, at which time it is loaded again based on the allocation. The RAG *r* IOIF1 In rate decrementer is decremented one out of $2^{(P+2)}$ EIB clocks, where *P* is the value of the IPr field. The RAG *r* IOIF1 In rate decrementer is used to generate tokens if enabled by the corresponding TKM_CR Register RAG *r* Rate Counters Enable bit. The RAG *r* IOIF1 Out rate decrementer functions in a similar manner using the OI*r* and OP*r* fields.

Valid values for the prescaler fields are 0 to 7.

Register Short Name	TKM_IOIF1_AR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50AFD8’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TKM



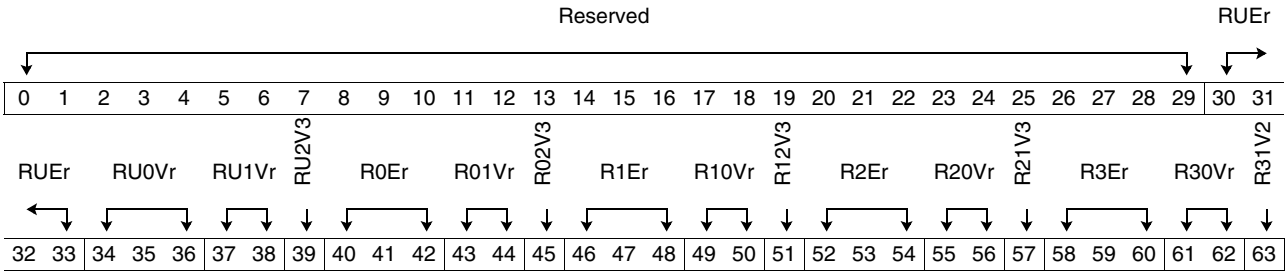
Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:4	IPU	IOIF1 In Prescaler for RAG U, where RAG U IOIF1 In Rate Decrementer is decremented every one out of 2^{IPU} EIB clock cycles
5:7	IIU	Interval for RAG U, where (IIU + 8) is loaded into the RAG U IOIF1 In Rate Decrementer when RAG U IOIF1 In Rate Decrementer would be decremented below zero.
8:10	IP0	IOIF1 In Prescaler for RAG 0, where RAG 0 IOIF1 In Rate Decrementer is decremented every one out of 2^{IP0} EIB clock cycles
11:13	II0	Interval for RAG 0, where (II0 + 8) is loaded into the RAG 0 IOIF1 In Rate Decrementer when RAG 0 IOIF1 In Rate Decrementer would be decremented below zero.

Bits	Field Name	Description
14:16	IP1	IOIF1 In Prescaler for RAG 1, where RAG 1 IOIF1 In Rate Decrementer is decremented every one out of 2^{IP1} EIB clock cycles
17:19	II1	Interval for RAG 1, where (II1 + 8) is loaded into the RAG 1 IOIF1 In Rate Decrementer when RAG 1 IOIF1 In Rate Decrementer would be decremented below zero.
20:22	IP2	IOIF1 In Prescaler for RAG 2, where RAG 2 IOIF1 In Rate Decrementer is decremented every one out of 2^{IP2} EIB clock cycles
23:25	II2	Interval for RAG 2, where (II2 + 8) is loaded into the RAG 2 IOIF1 In Rate Decrementer when RAG 2 IOIF1 In Rate Decrementer would be decremented below zero.
26:28	IP3	IOIF1 In Prescaler for RAG 3, where RAG 3 IOIF1 In Rate Decrementer is decremented every one out of 2IP3 EIB clock cycles
29:31	II3	Interval for RAG 3, where (II3 + 8) is loaded into the RAG 3 IOIF1 In Rate Decrementer when RAG 3 IOIF1 In Rate Decrementer would be decremented below zero.
32:33	Reserved	Bits are not implemented; all bits read back zero.
34:36	OPU	IOIF1 Out Prescaler for RAG U, where RAG U IOIF1 Out Rate Decrementer is decremented every one out of 2^{OPU} EIB clock cycles
37:39	OIU	Interval for RAG U, where (OIU + 8) is loaded into the RAG U IOIF1 Out Rate Decrementer when RAG U IOIF1 Out Rate Decrementer would be decremented below zero.
40:42	OP0	IOIF1 Out Prescaler for RAG 0, where RAG 0 IOIF1 Out Rate Decrementer is decremented every 1 out of 2^{OP0} EIB clock cycles
43:45	OI0	Interval for RAG 0, where (OI0 + 8) is loaded into the RAG 0 IOIF1 Out Rate Decrementer when RAG 0 IOIF1 Out Rate Decrementer would be decremented below zero.
46:48	OP1	IOIF1 Out Prescaler for RAG 1, where RAG 1 IOIF1 Out Rate Decrementer is decremented every 1 out of 2^{OP1} EIB clock cycles
49:51	OI1	Interval for RAG 1, where (OI1 + 8) is loaded into the RAG 1 IOIF1 Out Rate Decrementer when RAG 1 IOIF1 Out Rate Decrementer would be decremented below zero.
52:54	OP2	IOIF1 Out Prescaler for RAG 2, where RAG 2 IOIF1 Out Rate Decrementer is decremented every 1 out of 2^{OP2} EIB clock cycles
55:57	OI2	Interval for RAG 2, where (OI2 + 8) is loaded into the RAG 2 IOIF1 Out Rate Decrementer when RAG 2 IOIF1 Out Rate Decrementer would be decremented below zero.
58:60	OP3	IOIF1 Out Prescaler for RAG 3, where RAG 3 IOIF1 Out Rate Decrementer is decremented every 1 out of 2^{OP3} EIB clock cycles
61:63	OI3	Interval for RAG 3, where (OI3 + 8) is loaded into the RAG 3 IOIF1 Out Rate Decrementer when RAG 3 IOIF1 Out Rate Decrementer would be decremented below zero.

8.2.4 TKM Priority Register (TKM_PR)

This register defines which RAGs are granted unused tokens and unallocated tokens. If enabled by TKM_CR[UE], a RAG's unused token is made available to another RAG with an outstanding request. The RAG with the outstanding request is chosen based on priority values in this register. If enabled by TKM_CR[R*i*] where *i* = U, the Unallocated token is made available to a RAG with an outstanding request. The RAG with the outstanding request is chosen based on priority values in this register.

Register Short Name	TKM_PR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50AFE0'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TKM



Bits	Field Name	Description
0:29	Reserved	Bits are not implemented; all bits read back zero.
30:33	RUEr	RAG U tokens enable for RAG *r* to obtain these tokens, where *r* = 0 to 3 corresponding to bits [30:33], respectively.
34:36	RU0Vr	RAG U tokens: Priority of RAG 0 versus RAG *r*, where *r* = 1 to 3 corresponding to bits [34:36], respectively. For each bit: 0 RAG 0 has lower priority than RAG *r* for RAG U tokens 1 RAG 0 has higher priority than RAG *r* for RAG U tokens
37:38	RU1Vr	RAG U tokens: Priority of RAG 1 versus RAG *r*, where *r* = 2 to 3 corresponding to bits [37:38], respectively. For each bit: 0 RAG 1 has lower priority than RAG *r* for RAG U tokens 1 RAG 1 has higher priority than RAG *r* for RAG U tokens
39	RU2V3	RAG U tokens: Priority of RAG 2 versus RAG 3 0 RAG 2 has lower priority than RAG 3 for RAG U tokens 1 RAG 2 has higher priority than RAG 3 for RAG U tokens
40:42	R0Er	RAG 0 unused tokens: enable for RAG *r* to obtain these tokens, where *r* = 1 to 3 corresponding to bits [40:42], respectively.
43:44	R01Vr	RAG 0 tokens: Priority of RAG 1 versus RAG *r*, where *r* = 2 to 3 corresponding to bits [43:44], respectively. For each bit: 0 RAG 1 has lower priority than RAG *r* for RAG 0 unused tokens 1 RAG 1 has higher priority than RAG *r* for RAG 0 unused tokens
45	R02V3	RAG 0 tokens: Priority of RAG 2 versus RAG 3 0 RAG 2 has lower priority than RAG 3 for RAG 0 unused tokens 1 RAG 2 has higher priority than RAG 3 for RAG 0 unused tokens

Bits	Field Name	Description
46:48	R1Er	RAG 1 unused tokens: enable for RAG *r* to obtain these tokens, where *r* = 0, 2, or 3, corresponding to bits [46:48], respectively.
49:50	R10Vr	RAG 1 tokens: Priority of RAG 0 versus RAG *r*, where *r* = 2 to 3 corresponding to bits [49:50], respectively. For each bit: 0 RAG 0 has lower priority than RAG *r* for RAG 1 unused tokens 1 RAG 0 has higher priority than RAG *r* for RAG 1 unused tokens
51	R12V3	RAG 1 tokens: Priority of RAG 2 versus RAG 3 0 RAG 2 has lower priority than RAG 3 for RAG 1 unused tokens 1 RAG 2 has higher priority than RAG 3 for RAG 1 unused tokens
52:54	R2Er	RAG 2 unused tokens: enable for RAG r to obtain these tokens, where *r* = 0, 1, or 3, corresponding to bits [52:54], respectively.
55:56	R20Vr	RAG 2 tokens: Priority of RAG 0 versus RAG *r*, where *r* = 1 or 3 corresponding to bits [55:56], respectively. For each bit: 0 RAG 0 has lower priority than RAG *r* for RAG 2 unused tokens 1 RAG 0 has higher priority than RAG *r* for RAG 2 unused tokens
57	R21V3	RAG 2 tokens: Priority of RAG 1 versus RAG 3 0 RAG 1 has lower priority than RAG 3 for RAG 2 unused tokens 1 RAG 1 has higher priority than RAG 3 for RAG 2 unused tokens
58:60	R3Er	RAG 3 unused tokens: enable for RAG *r* to obtain these tokens, where *r* = 0, 1, or 2, corresponding to bits [58:60], respectively.
61:62	R30Vr	RAG 3 tokens: Priority of RAG 0 versus RAG *r*, where *r* = 1 to 2 corresponding to bits [61:62], respectively. For each bit: 0 RAG 0 has lower priority than RAG *r* for RAG 3 unused tokens 1 RAG 0 has higher priority than RAG *r* for RAG 3 unused tokens
63	R31V2	RAG 3 tokens: Priority of RAG 1 versus RAG 2 0 RAG 1 has lower priority than RAG 2 for RAG 3 unused tokens 1 RAG 1 has higher priority than RAG 2 for RAG 3 unused tokens

There are two types of fields in this register.

- RiEj enable field:
  - If *i* = U: RiEj indicates whether RAG j is enabled to obtain RAG U (Unallocated) tokens. If RiEj = 1, RAG j is enabled to obtain the RAG U tokens; otherwise, it is disabled.
  - For *i* = 0 to 3: RiEj indicates whether RAG j is enabled to obtain RAG i unused tokens. If RiEj = 1, RAG j is enabled to obtain the RAG i unused tokens; otherwise, it is disabled.
- RijVk priority field: This field is only used if RiEj = 1 and RiEk = 1.
  - If *i* = U: RijVk indicates whether RAG j has higher priority than RAG k in obtaining RAG U (Unallocated) tokens. If RijVk = 1, RAG j has higher priority than RAG k in obtaining RAG U tokens; otherwise, RAG j has lower priority than RAG k in obtaining RAG U tokens.
  - For *i* = 0 to 3: RijVk indicates whether RAG j has higher priority than RAG k in obtaining RAG i unused tokens. If RijVk = 1, RAG j has higher priority than RAG k in obtaining RAG i unused tokens; otherwise, RAG j has lower priority than RAG k in obtaining RAG i unused tokens.

If a circular priority order for RAG i unused or unallocated tokens is established such that some RAG n has higher priority than RAG m and RAG m has higher priority than RAG n and RiEn = 1 and RiEm = 1, which RAG is given the token is undefined. For example, if RAGs 0 – 2 are enabled for RAG 3 unused tokens, RAG 0 has higher priority than RAG 1, RAG 1 has higher priority than RAG 2, and RAG 0 has lower priority than RAG 2, which RAG is given RAG 3 unused token is undefined.

The following example demonstrates assigning values for RAG 1 unused tokens:

```
R1E0 = 1; R1E2 = 1; R1E3 = 1 (all other RAGs are enabled to obtain RAG 1 unused tokens)
R102 = 1 (RAG 0 has higher priority than 2)
R103 = 0 (RAG 0 has lower priority than 3)
R123 = 0 (RAG 2 has lower priority than 3)
```

The priority order from highest to lowest is 3, 0, 2. If a RAG 1 token becomes an unused token, and if all other RAGs have an outstanding request for the token, RAG 3 is given the token since it has higher priority than the other two RAGs.

8.2.5 TKM Interrupt Status Register (TKM_IS)

The TKM_IS Register indicates which TKM events caused an external interrupt condition. If feedback for a managed resource indicates that the command queue is full, the corresponding TKM_IS Register bit is set. When a TKM_IS Register bit is set to '1' and the corresponding interrupt is enabled, an exception signal from the TKM to the IIC is active. When this signal is active, the IIC sets the TKM exception bit in the IIC_IS register to '1'.

Reads of the TKM_IS Register are nondestructive. Once a TKM_IS Register bit is set to '1', it remains set until software resets the bit. To reset the TKM_IS Register bit, software must write a binary one to the corresponding bit.

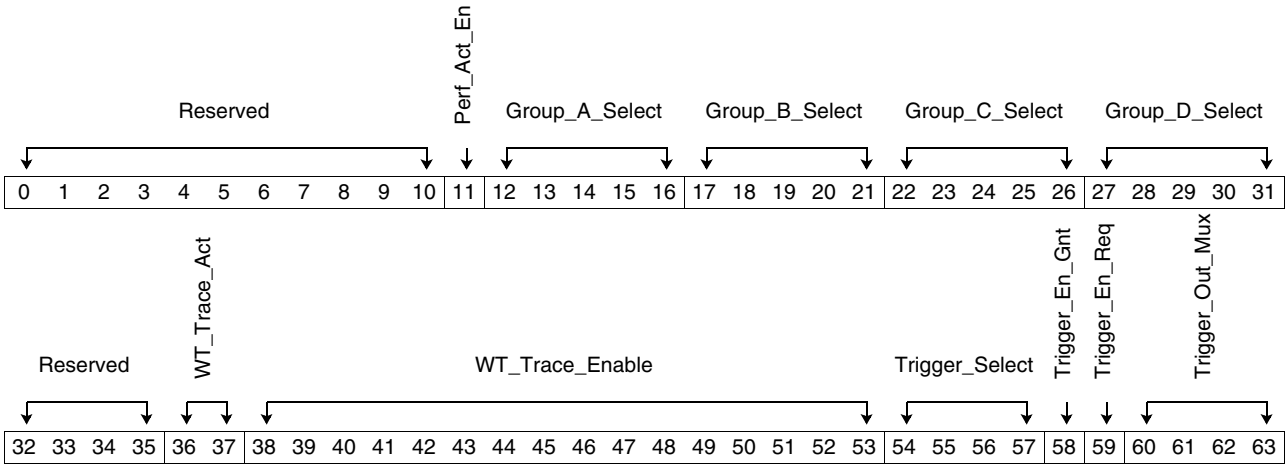
Register Short Name	TKM_IS	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'50AFE8'	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TKM

Bits	Field Name	Description
0:58	Reserved	Bits are not implemented; all bits read back zero.
59	M	MIC feedback Set to '1' if MIC feedback indicates that the command queue is full.
60	I0	In IOIF0 feedback Set to '1' if IOIF0 In feedback indicates that the command queue is full.
61	O0	Out IOIF0 feedback Set to '1' if IOIF0 Out feedback indicates that the command queue is full.
62	I1	In IOIF1 feedback Set to '1' if IOIF1 In feedback indicates that the command queue is full.
63	O1	Out IOIF1 feedback Set to '1' if IOIF1 Out feedback indicates that the command queue is full.

8.2.6 TKM Performance Monitor Control Register (TKM_PMCR)

This register controls trace and performance monitor functions in the token manager.

Register Short Name	TKM_PMCR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘50AFF0’	**Memory Map Area**	MIC and TKM
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TKM



Bits	Field Name	Description
0:10	Reserved	Bits are not implemented; all bits read back zero.
11	Perf_Act_En	TKM performance monitor latches enable. Disable to save power when not in use. 0 Disabled 1 Enabled
12:16	Group_A_Select	Performance monitor group A select for debug bus bits [0:15]. Values not listed are invalid. 00000 Disable Group A 10000 Select Signal Group A0 01000 Select Signal Group A1 00100 Select Signal Group A2 00010 Select Signal Group A3 00001 Select Signal Group A4
17:21	Group_B_Select	Performance monitor group B select for debug bus bits [16:31]. Values not listed are invalid. 00000 Disable Group B 10000 Select Signal Group B0 01000 Select Signal Group B1 00100 Select Signal Group B2 00010 Select Signal Group B3 00001 Select Signal Group B4

Bits	Field Name	Description
22:26	Group_C_Select	Performance monitor group C select for debug bus bits [64:79]. Values not listed are invalid. 00000 Disable Group C 10000 Select Signal Group C0 01000 Select Signal Group C1 00100 Select Signal Group C2 00010 Select Signal Group C3 00001 Select Signal Group C4
27:31	Group_D_Select	Performance monitor group D select for debug bus bits [80:95]. Values not listed are invalid. 00000 Disable Group D 10000 Select Signal Group D0 01000 Select Signal Group D1 00100 Select Signal Group D2 00010 Select Signal Group D3 00001 Select Signal Group D4
32:35	Reserved	Bits are not implemented; all bits read back zero.
36:37	WT_Trace_Act	Enables for performance monitor latches in the EIB that are used for token manager performance monitor, trace, triggers, and events. Disable to save power when not in use. 00 Disabled 01 Enabled
38:53	WT_Trace_Enable	Trace Bus byte enables, which must be set to ‘1’ to propagate the Trace Bus byte from other units as well as from token manager. For n = 0 to 15, TKM_PMCR[38 + n] corresponds to Trace Bus byte n. x‘0000’ Disabled x‘0001’ Enabled
54:57	Trigger_Select	Select for token request and grant on Trigger Bus. Values not listed are invalid. 0000 IOC0 token request and grant 0001 IOC1 token request and grant 0010 SPE2 token request and grant 0011 SPE3 token request and grant 0100 SPE4 token request and grant 0101 SPE5 token request and grant 0110 SPE6 token request and grant 0111 SPE7 token request and grant 1000 SPE0 token request and grant 1001 SPE1 token request and grant 1010 PPE token request and grant
58	Trigger_En_Gnt	Enable Token Grant on Trigger Bus. 0 Disabled 1 Enabled
59	Trigger_En_Req	Enable token request on Trigger Bus. 0 Disabled 1 Enabled

Bits	Field Name	Description
60:63	Trigger_Out_Mux	Selects for request or grant triggers on Trigger Bus bits. Trigger_Out_Mux[0:1] selects position of request trigger when enabled. 00 Trigger Bus bit [0] 01 Trigger Bus bit [1] 10 Trigger Bus bit [2] 11 Trigger Bus bit [3] Trigger_Out_Mux[2:3] selects position of grant trigger when enabled. 00 Trigger Bus bit [0] 01 Trigger Bus bit [1] 10 Trigger Bus bit [2] 11 Trigger Bus bit [3] Setting Trigger_Out_Mux[0:1] = Trigger_Out_Mux[2:3] when both request and grant triggers are enabled will OR the triggers into the same bit position.

9. CBE Distribution (BED) of I/O MMIO Registers

This section describes the BED memory-mapped I/O (MMIO) registers. *Table 9-1* shows the BED MMIO memory map and lists the BED registers. Though these registers are part of the BED function, they tie into the BIC BClk MMIO ring on slice 1. All BED MMIO registers are part of the Slice 1 BClk MMIO ring. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

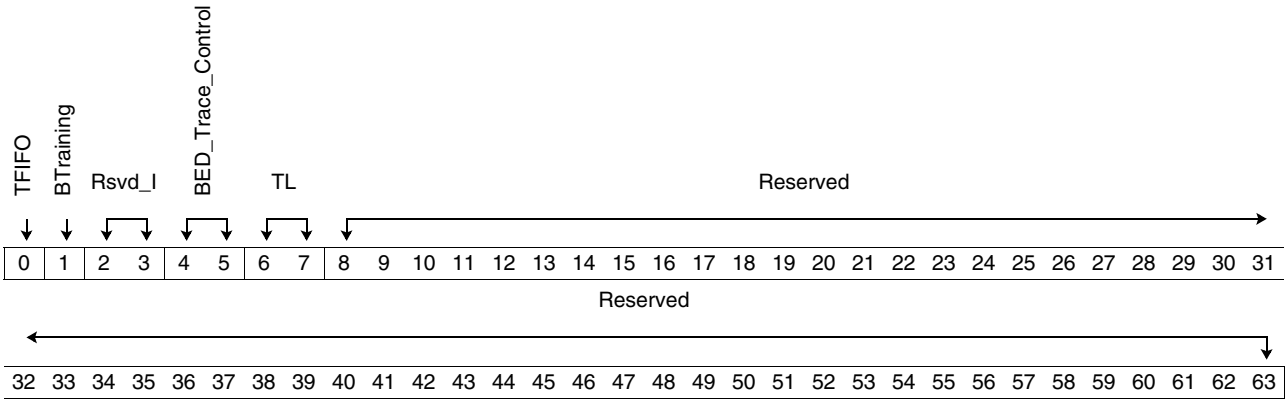
Table 9-1. BED MMIO Memory Map

Hexadecimal Offset (x‘513 *nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x‘600’ – Link 0 x‘608’ – Link 1	*BED Link n [n = 0, 1] Transmit Byte Training Control Registers (BED_Lnk0_TransBytTrngCntl, BED_Lnk1_TransBytTrngCntl)*	64	R/W	*Section 9.1* on page 222
x‘610’ – Link 0 x‘618’ – Link 1	*BED Link n [n = 0, 1] Receive Byte Training Control Registers (BED_RecBytTrngCntl_Lnk0, BED_RecBytTrngCntl_Lnk1)*	64	R/W	*Section 9.2* on page 223
x‘620’	*BED RRAC Register Control Register (BED_RRAC_RegCntl)*	64	R/W	*Section 9.3* on page 225
x‘628’	*BED RRAC Register Read Data Register (BED_RRAC_RegRdDat)*	64	R/W	*Section 9.4* on page 226

9.1 BED Link n [n = 0, 1] Transmit Byte Training Control Registers (BED_Lnk0_TransBytTrngCntl, BED_Lnk1_TransBytTrngCntl)

This register is used by software to enable the transmit first-in first-outs (FIFOs) and to send training patterns. The typical value after training is done is x‘8200000’.

Register Short Name	BED_Lnk0_TransBytTrngCntl BED_Lnk1_TransBytTrngCntl	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘513600’ x‘513608’	**Memory Map Area**	BIC 1 BClk
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BED

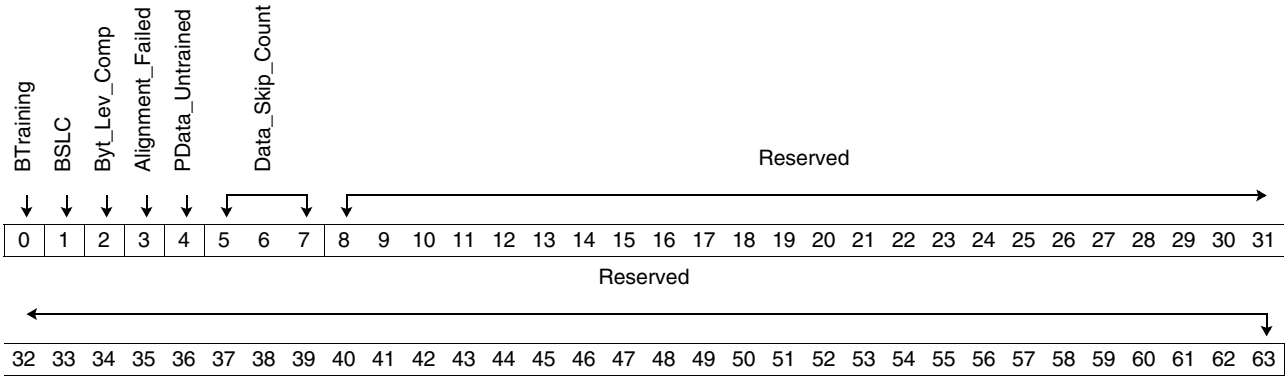


Bits	Field Name	Description
0	TFIFO	Enable transmit FIFO Set by software after bit levelization is complete in the FlexIO. Setting this bit enables transmission of idle characters on all bytes of the link. Resetting (clearing) this bit resets the transmit FIFO. Do not set this bit until FlexIO initialization is complete, and the clocks from the FlexIO are stable.
1	BTraining	Enable byte training Set by software after bit levelization is complete in the FlexIO and the Enable Transmit FIFO bit is set. Setting this bit enables transmission of the hardware byte-training sequence. Resetting (clearing) this bit stops transmission of the training sequence.
2:3	Rsvd_I	Reserved. Latch bits are implemented; value read is the value written.
4:5	BED_Trace_Control	**This field is only implemented in BED_Lnk0_TransBytTrngCntl.** In BED_Lnk0_TransBytTrngCntl, bits 4 and 5 are the BED Trace Control. Set either or both bits to ‘1’ to enable BED trace data. In BED_Lnk1_TransBytTrngCntl, bits 4 and 5 are reserved. Latch bits are implemented; value read is the value written.
6:7	TL	Transmit latency. The default is ‘00’. For the CBE, this field must be set to ‘10’ (2). Software should only modify this field before the transmit FIFO is enabled. This field is intended for use during debug and characterization and should only be changed with values provided by the hardware designers. Setting this field increases the latency for data transmitted on the link. In hardware, it controls the separation of the write and read pointers for the transmit FIFO.
8:63	Reserved	Bits are not implemented; all bits read back zero.

9.2 BED Link n [n = 0, 1] Receive Byte Training Control Registers (BED_RecBytTrngCntl_Lnk0, BED_RecBytTrngCntl_Lnk1)

This register is used by software to enable byte training on the receive path. The typical value after training is done is x‘66000000’.

Register Short Name	BED_RecBytTrngCntl_Lnk0 BED_RecBytTrngCntl_Lnk1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘513610’ x‘513618’	**Memory Map Area**	BIC 1 BClk
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BED



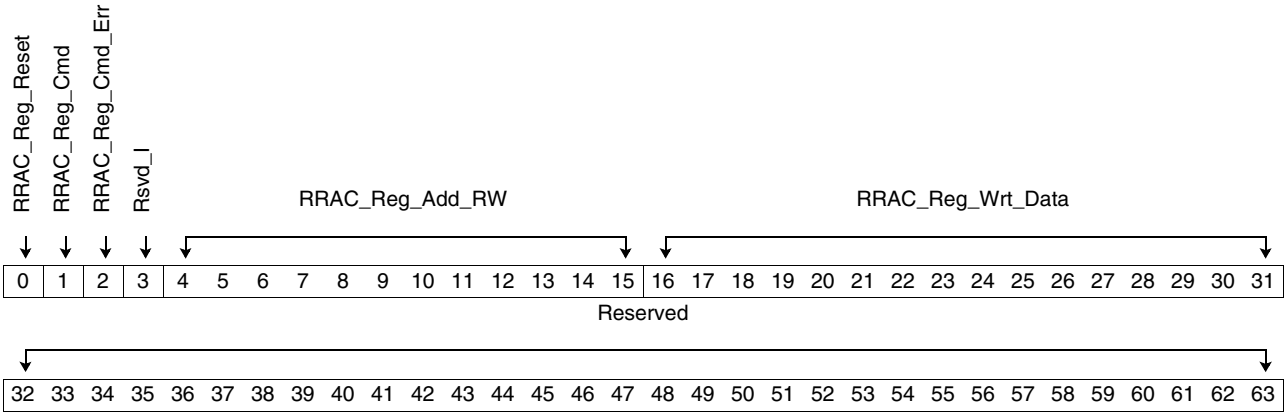
Bits	Field Name	Description
0	BTraining	Enable byte training Set by software after the Enable Byte Training bit is set at the source (TX) end of the link. This bit initiates the hardware hunt for the correct training pattern. It must be reset after training is complete, and before starting normal traffic on the link.
1	BSLC	Byte shift learning complete Set by hardware after the correct byte alignment has been found on all bytes of the link. On a 2:1 FlexIO no byte shifting is required, therefore this bit is set immediately after the Start Byte Training bit is set. This is a status bit that indicates byte training progress.
2	Byt_Lev_Comp	Byte levelization complete Set by hardware after the receiving FIFOs are all adjusted to align all bytes within the link. Software polls this bit to detect completion of byte training.
3	Alignment_Failed	Alignment failed Set by hardware to indicate a failure during byte training. If this bit is detected as set during byte training, software must clear the RX and TX training registers, and restart the training sequence. Set to ‘1’ if nine BCLK cycles have elapsed since the time an 8-bit physical link received x‘EE00’ without receiving x‘EE00’ on all 8-bit links used for one device. Software only modifies this bit when byte training is not active. For example, software modifies this bit when the [Byte_Training] bit in this register is already set to ‘0’, or if the [Byte_Training] bit had been set to ‘1’ and the hardware has set either the [Byt_Lev_Comp] or [Alignment_Failed] bit to ‘1’.

Bits	Field Name	Description
4	PData_Untrained	Enable Untrained Pass Data Set this bit to allow the receive FIFOs to pass data through without being trained. It should only be set for debug purposes. **Note:** Complete the following steps to enable the bypass mode. 1. Set this register to x‘8E000000’. 2. Wait 16 cycles. 3. Set bit [Byte_Training] to ‘0’. Do not modify the other bits (x‘6E000000’).
5:7	Data_Skip_Count	Data skip count This field is set by software. Hardware uses it to control the number of FIFO entries skipped before starting to read the FIFO when byte levelization completes. FIFO entries must be skipped in some designs due to the number of cycles of delay lost to communicate alignment between different 8-bit physical links. The FIFO read pointer must start at an offset based on this number of bclk cycles in the Cell Broadband Engine. The design implementation dictates the value required for this field. 000 Default 110 For the Cell Broadband Engine (CBE), this field must be set to ‘110’ (6). Software should only modify this field before byte training is performed.
8:63	Reserved	Bits are not implemented; all bits read back zero.

9.3 BED RRAC Register Control Register (BED_RRAC_RegCntl)

This register is used by software to access FlexIO register space.

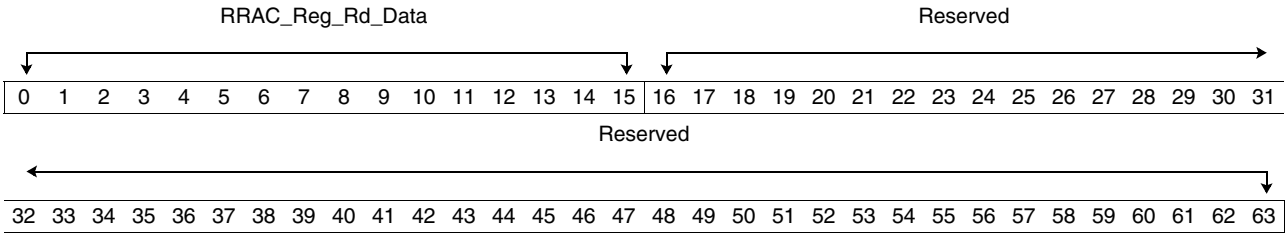
Register Short Name	BED_RRAC_RegCntl	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘513620’	**Memory Map Area**	BIC 1 BClk
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BED



Bits	Field Name	Description
0	RRAC_Reg_Reset	Enable FlexIO Register Reset Set by hardware during register reset and cleared after the reset sequence completes. Software can set this bit to force a FlexIO register reset sequence to start.
1	RRAC_Reg_Cmd	FlexIO register command 0 Read 1 Write
2	RRAC_Reg_Cmd_Err	FlexIO register command error. A write to the FlexIO register control register was attempted while FlexIO register command in progress. The default is 0. Hardware sets this to 1 if a write to a FlexIO register happens while there is already a FlexIO register command in progress.
3	Rsvd_I	Reserved. Behavior is unpredictable.
4:15	RRAC_Reg_Add_RW	FlexIO register address to be written or read Bits[4:15] correspond to FlexIO Reg_Addr[0:11]
16:31	RRAC_Reg_Wrt_Data	FlexIO register write data Bits[16:31] correspond to FlexIO Reg_Wr_Data[0:15]
32:63	Reserved	Bits are not implemented; all bits read back zero.

9.4 BED RRAC Register Read Data Register (BED_RRAC_RegRdDat)

Register Short Name	BED_RRAC_RegRdDat	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘513628’	**Memory Map Area**	BIC 1 BClk
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BED



Bits	Field Name	Description
0:15	RRAC_Reg_Rd_Data	FlexIO register read data These bits correspond to FlexIO Reg_Rd_Data[0:15]. The default is x‘0000’. When software reads this register, the BED issues the read to the FlexIO register address specified by the register control register. The BED returns the FlexIO register value when the read sequence is complete.
16:63	Reserved	Bits are not implemented; all bits read back zero.

10. Element Interconnect Bus (EIB) MMIO Registers

This section describes the EIB IOC memory-mapped I/O (MMIO) registers. *Table 10-1* shows the EIB MMIO memory map and lists the EIB registers. The EIB register space starts at x‘511800’ and ends at x‘511BFF’. Offsets are from the start of the EIB register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.

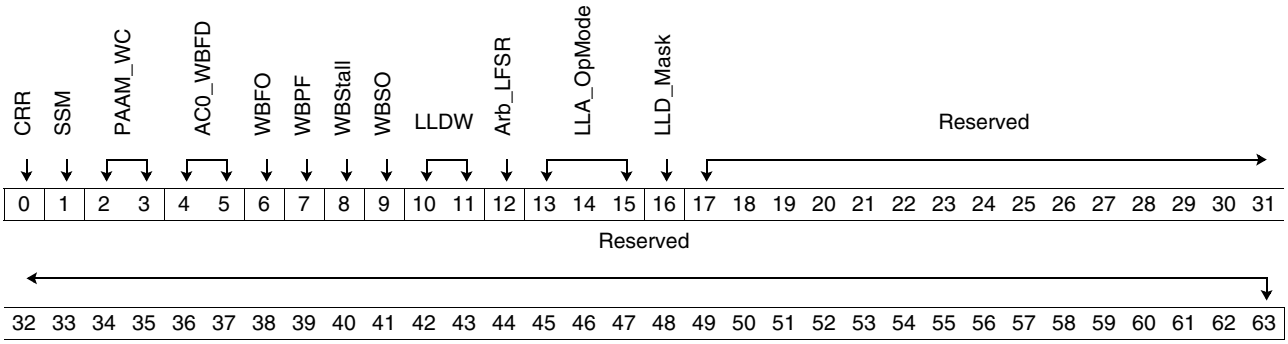
Table 10-1. EIB MMIO Memory Map

Hexadecimal Offset from BE_MMIO_Base (x‘511*nnn*’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x‘800’	*EIB AC0 Control Register (EIB_AC0_CTL)*	64	R/W	*Section 10.1* on page 228
x‘808’	Reserved			
x‘810’	*EIB Interrupt Register (EIB_Int)*	64	R/W	*Section 10.2* on page 230
x‘840’	*EIB Local Base Address Register 0 (EIB_LBAR0)*	64	R/W	*Section 10.3* on page 231
x‘848’	*EIB Local Base Address Mask Register 0 (EIB_LBAMR0)*	64	R/W	*Section 10.4* on page 232
x‘850’	*EIB Local Base Address Register 1 (EIB_LBAR1)*	64	R/W	*Section 10.4.1* on page 233
x‘858’	*EIB Local Base Address Mask Register 1 (EIB_LBAMR1)*	64	R/W	*Section 10.4.2* on page 234
x‘860’ – x‘868’	Reserved			
x‘870’	*EIB AC/Darb Configuration Register (EIB_Cfg)*	64	R/W	*Section 10.4.3* on page 235
x‘878’ – x‘BFF’	Reserved			

10.1 EIB AC0 Control Register (EIB_AC0_CTL)

This register is used to control several address concentrator 0 (AC0) functions.

Register Short Name	EIB_AC0_CTL	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511800'	**Memory Map Area**	EIB
Value at Initial POR	x'88060000_00000000'	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	EIB



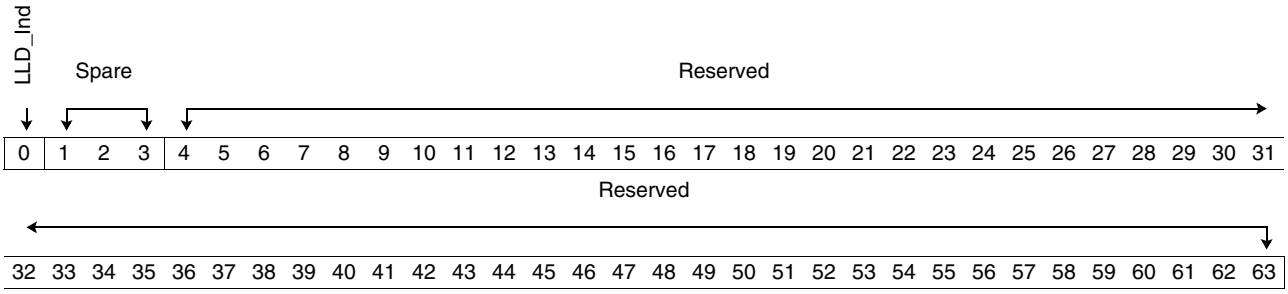
Bits	Field Name	Description
0	CRR	Command Reflection Rate 0 Disable 1 Enables AC0 to reflect commands to AC1 at the maximum rate of one command per cycle. AC0 attempts to reflect certain types of commands every cycle when enabled (M = 0, TType = '*nnn*10'. In a multiCBE system, this bit should be set to the same value in all the CBE processors. (default value)
1	SSM	Single Step Mode 0 Disable 1 Sets AC0 into Single Step mode. In this mode, AC0 reflects one command and waits for the snoop response to return before it reflects another command. **Note:** This bit is for debug use only. This bit has no effect if AC0 is off chip.
2:3	PAAM_WC	PAAM Window Control These bits set the PAAM window. 00 2 cycles (default) 01 3 cycles 10 4 cycles 11 5 cycles This bit delays the invalidation of the CAM entry and may have minimal effect on the EIB that has AC0 off chip (two-CBE configuration).

Bits	Field Name	Description
4:5	AC0_WBFD	AC0 Wait Buffer Flush Delay These bits control when AC0 stops accepting new commands from AC1 and then flush the wait buffer before again accepting new commands from AC1. The purpose of these bits is to prevent a livelock. A livelock occurs when many EIB masters are making accesses to the same cache line and commands stored in the Wait Buffer cannot make forward progress in a reasonable amount of time. The hardware detects when wait buffer commands leave the wait buffer, hit the PAAM window, and reenter the wait buffer again. A wait buffer empty condition resets all active counters. 00 Never flush the Wait Buffer. 01 Flush after 16 commands have reentered the Wait Buffer. 10 Flush after 64 commands have reentered the Wait Buffer. (default value) 11 Flush after 256 commands have reentered the Wait Buffer.
6	WBFO	Wait Buffer Flush Override 0 Normal operation 1 Flush immediately when not empty
7	WBPF	Disable Wait Buffer Prefetch This field slows down the wait buffer fetch rate.
8	WBStall	Disable Wait Buffer Stall This field allows new commands to mix with wait buffer commands.
9	WBSO	Disable Wait Buffer Shutoff Logic This field allows new commands to end up at the head of the wait queue.
10:11	LLDW	Livelock Detection Window 00 fewer than 16 M = 1 commands needed to refill the wait buffer since it was last full (default setting) 01 <32 M = 1 commands to refill 10 <64 M = 1 commands to refill 11 <128 M = 1 commands to refill
12	Arb_LFSR	AC0 Arbitration control between internal and off-chip commands 0 Randomize the selection of internal and off-chip commands using an 8-bit LFSR function. (default) 1 Alternate between internal and off-chip commands.
13:15	LLA_OpMode	LiveLock Avoidance Operating Mode 000 Mixed mode, exit after 8K M = 1 commands. 001 Mixed mode, exit after 64K M = 1 commands. 010 Mixed mode, exit when LiveLock Interrupt bit is reset. 011 Reflect and Retry mode (wait buffer disabled, LiveLock interrupt disabled). 10x Wait Buffer mode (Current Pass1 operating mode, LiveLock interrupt disabled). 110 Default. CBC mode with interrupt issued upon possible livelock detected. 111 CBC mode with interrupt disabled. Reflect and Retry mode-Any command that gets a PAAM Hit during Reflect and Retry mode is reflected instead of going into the PAAM wait buffer as happens during normal operation; AC0 then forces the Retry bit for said command to be active in the combined snoop response. Mixed mode-AC0 runs normally with the PAAM wait buffer until a possible livelock is detected, then switches to Reflect and Retry mode. AC0 exits Reflect and Retry mode when the programmed exit condition occurs. CBC mode-Cycle-By-Cycle mode. Run in Wait Buffer mode until the PAAM wait buffer fills up. At that point, rather than stall the command pipe like normal Wait Buffer mode would, switch to Reflect and Retry mode until a wait buffer entry frees up. Then return back to Wait Buffer mode.
16	LLD_Mask	EIB Possible Livelock Detection Interrupt Mask 0 Interrupt enabled. When EIB_Int[0] is set to ‘1’, an interrupt is sent to IIC_IS[59]. 1 Default. Interrupt masked. Register bit EIB_Int[0] can still be set to ‘1’, but no interrupt is sent to IIC_IS[59].
17:63	Reserved	Bits are not implemented; all bits read back zero.

10.2 EIB Interrupt Register (EIB_Int)

This register is used to log that a possible livelock condition has been detected on the EIB.

Register Short Name	EIB_Int	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511810'	**Memory Map Area**	EIB
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	EIB



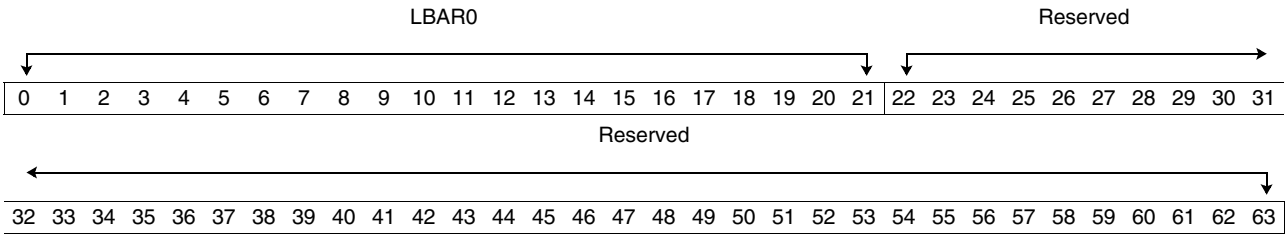
Bits	Field Name	Description
0	LLD_Ind	EIB Possible LiveLock Detection Indicator 0 None detected 1 A possible livelock condition (overuse of the previous adjacent address match (PAAM) waitbuffer) has been detected. Contact your IBM technical support representative for more information about the EIB Possible Livelock Detection Interrupt. **Note:** An MMIO write that sets this bit to '1' when it was previously '0' causes IIC_IS[59] to be set, provided EIB_AC0_CTL[16] is '0', and regardless of the state of EIB_AC0_CTL[13:15]. EIB_AC0_CTL[16] can be used to mask this interrupt. Additionally, the LLD_Ind bit only sends an interrupt when it transitions from '0' to '1'. If it is not reset to '0', it will not issue another interrupt.
1:3	Spare	Spare Bits This field can be read from or written to without side effects.
4:63	Reserved	Bits are not implemented; all bits read back zero.

10.3 EIB Local Base Address Register 0 (EIB_LBAR0)

This register contains the base address of the primary noncoherent address range. This base address is used in conjunction with the EIB_LBAMR0 register to determine whether a command address falls within noncoherent range 0.

Read and write commands in this address range are considered to be local commands. Local commands have the M bit set to '0' and are reflected back to bus masters after arbitration.

Register Short Name	EIB_LBAR0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511840'	**Memory Map Area**	EIB
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	EIB



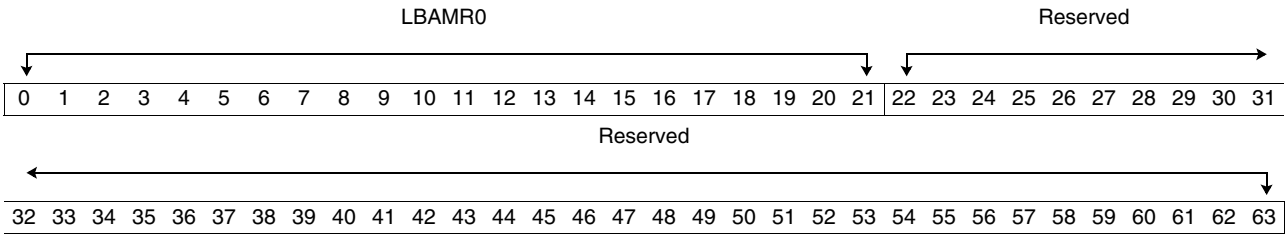
Bits	Field Name	Description
0:21	LBAR0	Defines the primary noncoherent command address range. Bits [0:21] of this register correspond to bits [22:43] of the real address used on the EIB.
22:63	Reserved	Bits are not implemented; all bits read back zero.

10.4 EIB Local Base Address Mask Register 0 (EIB_LBAMR0)

This register is used to hold the mask (bit enables) for the primary noncoherent address range. This mask is used to specify which bits from the command address should be compared to the EIB_LBAR0 register to determine whether the command address falls within noncoherent range 0.

Read and write commands in the noncoherent range are considered to be local commands. Local commands have the M bit set to '0' and are reflected back to bus masters after arbitration.

Register Short Name	EIB_LBAMR0	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511848'	**Memory Map Area**	EIB
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	EIB



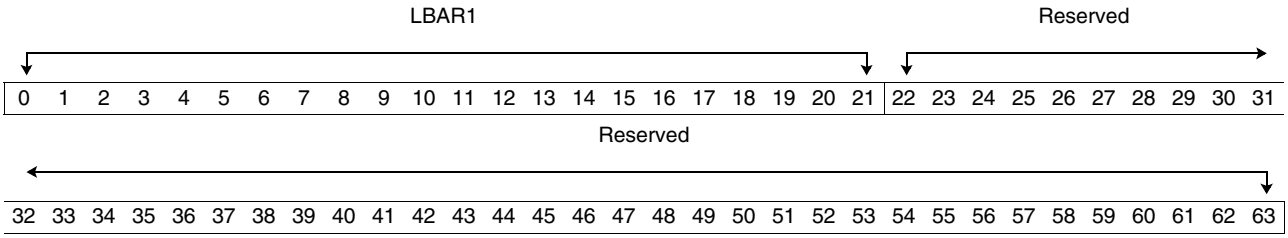
Bits	Field Name	Description
0:21	LBAMR0	Local Base Address Mask Register 0 Mask for use with Local Base Address Register 0
22:63	Reserved	Bits are not implemented; all bits read back zero.

10.4.1 EIB Local Base Address Register 1 (EIB_LBAR1)

This register contains the base address of the secondary noncoherent address range. This base address is used in conjunction with the EIB_LBAMR1 register to determine if a command address falls within noncoherent range 1.

Read and write commands in this address range are considered to be local commands. Local commands have the M bit set to '0' and are reflected back to bus masters after arbitration.

Register Short Name	EIB_LBAR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511850'	**Memory Map Area**	EIB
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	EIB



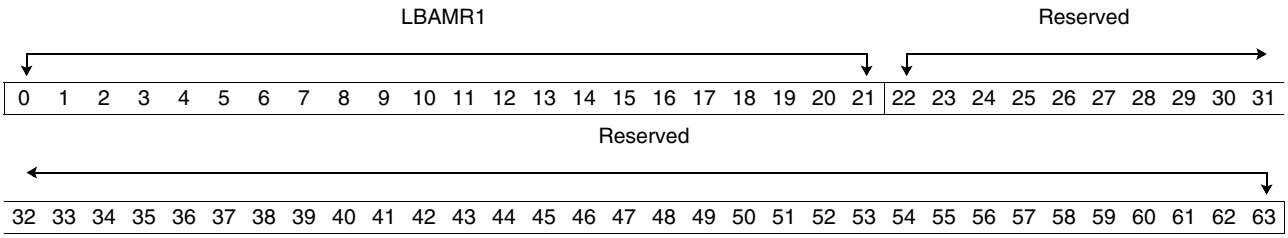
Bits	Field Name	Description
0:21	LBAR1	Local Base Address Register 1 Defines secondary noncoherent command address range. Bits [0:21] of this register correspond to bits [22:43] of the Real Address used on the EIB.
22:63	Reserved	Bits are not implemented; all bits read back zero.

10.4.2 EIB Local Base Address Mask Register 1 (EIB_LBAMR1)

This register contains the mask (bit enables) for the secondary noncoherent address range. This mask is used to specify which bits from the command address should be compared to the EIB_LBAR1 register to determine if the command address falls within noncoherent range 1.

Read and write commands in the noncoherent range are considered to be local commands. Local commands have the M bit set to '0' and are reflected back to bus masters after arbitration.

Register Short Name	EIB_LBAMR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511858'	**Memory Map Area**	EIB
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	EIB

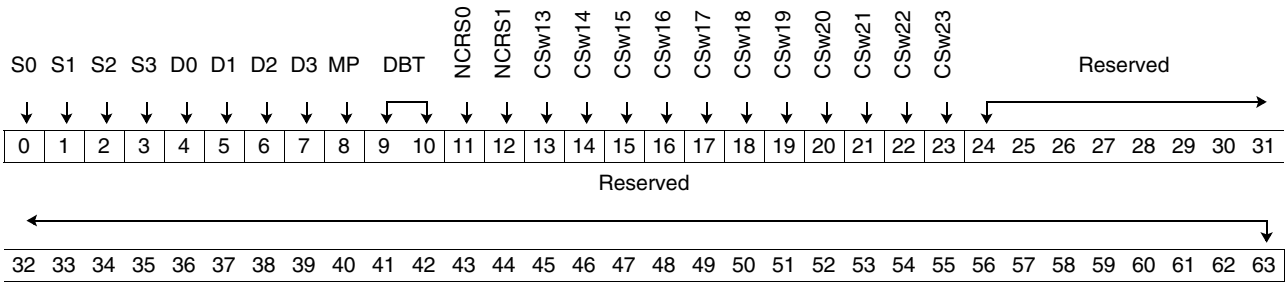


Bits	Field Name	Description
0:21	LBAMR1	Local Base Address Mask Register 1 Mask for use with Local Base Address Register 1
22:63	Reserved	Bits are not implemented; all bits read back zero.

10.4.3 EIB AC/Darb Configuration Register (EIB_Cfg)

Use this register to control configuration options and control switches in the address concentrators and data arbiter.

Register Short Name	EIB_Cfg	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'511870'	**Memory Map Area**	EIB
Value at Initial POR	x'00080000_00000000'	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	EIB



Bits	Field Name	Description
0	S0	Enable Single Thread Mode for Data Ring 0
1	S1	Enable Single Thread Mode for Data Ring 1
2	S2	Enable Single Thread Mode for Data Ring 2
3	S3	Enable Single Thread Mode for Data Ring 3
4	D0	Disable Data Ring 0
5	D1	Disable Data Ring 1
6	D2	Disable Data Ring 2
7	D3	Disable Data Ring 3 **Caution:** A minimum of one even data ring and one odd data ring must be enabled at all times, or deadlock might occur. Illegal combinations for bits [4:7] (in hexadecimal) are: 5, 7, A, B, D, E, and F.
8	MP	MIC Priority 0 MIC always wins over any other unit 1 Treat MIC as an equal
9:10	DBT	Destination Busy Timeout These bits determine how long to wait before resetting a destbusy latch. (EIB livelock prevention) 00 12 - 23 cycles 01 16 - 31 cycles 10 32 - 63 cycles 11 Never (EIB denial-of-service livelock possible)
11	NCRS0	Noncoherent (NC) Range Switch 0 0 Make local NC Range 0 local 1 Make local NC Range 0 global

Bits	Field Name	Description
12	NCRS1	Noncoherent (NC) Range Switch 1 0 Make local NC Range 1 local 1 Make local NC Range 1 global
13	CSw13	Disable Credit Switch 13 Turns off command credits to the bus masters for unit 0 (BIC), which locks them out of the EIB.
14	CSw14	Disable Credit Switch 14 Turns off command credits to the bus masters for unit 1 (SPE6), which locks them out of the EIB.
15	CSw15	Disable Credit Switch 15 Turns off command credits to the bus masters for unit 2 (SPE4), which locks them out of the EIB.
16	CSw16	Disable Credit Switch 16 Turns off command credits to the bus masters for unit 3 (SPE2), which locks them out of the EIB.
17	CSw17	Disable Credit Switch 17 Turns off command credits to the bus masters for unit 4 (SPE0), which locks them out of the EIB.
18	CSw18	Disable Credit Switch 18 Turns off command credits to the bus masters for unit 5 (PPU), which locks them out of the EIB.
19	CSw19	Disable Credit Switch 19 Turns off command credits to the bus masters for unit 7 (SPE1), which locks them out of the EIB.
20	CSw20	Disable Credit Switch 20 Turns off command credits to the bus masters for unit 8 (SPE3), which locks them out of the EIB.
21	CSw21	Disable Credit Switch 21 Turns off command credits to the bus masters for unit 9 (SPE5), which locks them out of the EIB.
22	CSw22	Disable Credit Switch 22 Turns off command credits to the bus masters for unit A (SPE7), which locks them out of the EIB.
23	CSw23	Disable Credit Switch 23 Turns off command credits to the bus masters for unit B (IOC), which locks them out of the EIB.
24:63	Reserved	Bits are not implemented; all bits read back zero.

11. Pervasive MMIO Registers

This section describes the pervasive Performance Monitor, Power Management, and Thermal Management memory-mapped I/O (MMIO) registers. *Table 11-1* shows the pervasive MMIO memory map and lists the pervasive registers. The pervasive MMIO Registers area starts at x‘509000’ and ends at x‘509FFF’. Offsets are from the start of the pervasive register space. For the complete CBE MMIO memory map, see *Section 1 Cell Broadband Engine Memory-Mapped I/O Registers* on page 19.

The following notes apply to the register bit definitions:

- Multiple address offsets for a register indicate that there are multiple instances of this register.
- The *Privilege Type* of all MMIO registers is recommended by the *Cell Broadband Engine Architecture*, but is not enforced in hardware.
- The *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the power-on reset (POR) sequence.

11.1 Pervasive MMIO Memory Map

Note: Within the (RAS) test control unit (TCU) shared memory space, the 32-bit registers are defined on even word addresses. This means that these 32-bit registers are defined on bits [0:31] of the MMIO double-word.

Table 11-1. Pervasive Registers (Page 1 of 3)

Hexadecimal Offset (x‘509000’ – x‘509FFF’)	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
Fault Isolation Registers				
x‘C00’	*Global Fault Isolation Register (checkstop_fir)*	32	R	*Section 11.2.1* on page 240
x‘C08’	*Global Fault Isolation Register for Recoverable Errors (recoverable_fir)*	32	R	*Section 11.2.2* on page 241
x‘C10’	*Global Fault Isolation Register For Special Attention And Machine Check (spec_att_mchk_fir)*	32	R/R	*Section A.11* on page 345
x‘C18’	*Global Fault Isolation Mode Register (fir_mode_reg)*	32	R/W	*Section A.11* on page 345
x‘C20’	*Global Fault Isolation Error Enable Mask Register (fir_enable_mask)*	32	R/W	*Section A.11* on page 345
x‘C28’ – x‘C30’	Reserved			
x‘C38’	*SPE Available Partial Good Register (SPE_available)*	32	R	*Section 11.2.3* on page 242
x‘C40’ – x‘C58’	Reserved			
x‘C80’	*Serial Number (serial_number)*	64	R	*Section 11.2.4* on page 243
x‘C88’ – x‘CB8’	Reserved			
Performance Monitor Registers shared with the TLA				
x‘008’	*Group Control Register (group_control)*	32	W	*Section 11.3.1* on page 244
x‘0A8’	*Debug Bus Control Register (debug_bus_control)*	32	W	*Section 11.3.2* on page 245
x‘108’	*Trace Buffer High Doubleword Register (0 to 63) (trace_buffer_high)*	64	R	*Section 11.3.3* on page 247

Table 11-1. Pervasive Registers (Page 2 of 3)

Hexadecimal Offset (x'509000' – x'509FFF')	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
x'110'	*Trace Buffer Low Doubleword Register (64 to 127) (trace_buffer_low)*	64	R	*Section 11.3.4* on page 248
x'118'	*Trace Address Register (trace_address)*	64	R/W	*Section 11.3.5* on page 249
x'120'	*External Trace Timer Register (ext_tr_timer)*	64	W	*Section 11.3.6* on page 250
Performance Monitor Only Registers				
x'400'	*Performance Monitor Status/Interrupt Mask Register (pm_status)*	32	R/W	*Section 11.4.1* on page 251
x'408'	*Performance Monitor Control Register (pm_control)*	32	W	*Section 11.4.2* on page 252
x'410'	*Performance Monitor Interval Register (pm_interval)*	32	R/W	*Section 11.4.3* on page 255
x'418'	*Performance Monitor Counter Pairs Registers (pmM_N)*	32	R/W	*Section 11.4.4* on page 256
x'420'				
x'428'				
x'430'				
x'438'	*Performance Monitor Start Stop Register (pm_start_stop)*	32	W	*Section 11.4.5* on page 257
x'440'	*Performance Monitor Counter Control Registers (pmN_control)*	32	W	*Section 11.4.6* on page 259
x'448'				
x'450'				
x'458'				
x'460'				
x'468'				
x'470'				
x'478'				
Power Management Control Registers				
x'880'	*Power Management Control Register (PMCR)*	64	R/W	*Section 11.5.1* on page 261
x'888'	*Power Management Status Register (PMSR)*	64	R	*Section 11.5.2* on page 263

Table 11-1. Pervasive Registers (Page 3 of 3)

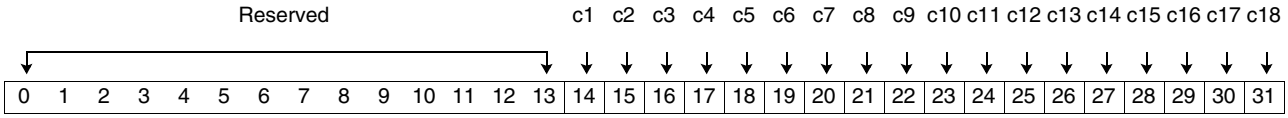
Hexadecimal Offset (x'509000' – x'509FFF')	Register Name and (Short Name)	Width (Bits)	Read/ Write	Additional Information
Thermal Management MMIO Registers				
x'800'	*Thermal Sensor Current Temperature Status Register 1 (TS_CTSR1)*	64	R	*Section 11.6.1* on page 265
x'808'	*Thermal Sensor Current Temperature Status Register 2 (TS_CTSR2)*	64	R	*Section 11.6.2* on page 267
x'810'	*Thermal Sensor Maximum Temperature Status Register 1 (TS_MTSR1)*	64	R	*Section 11.6.3* on page 268
x'818'	*Thermal Sensor Maximum Temperature Status Register 2 (TS_MTSR2)*	64	R	*Section 11.6.4* on page 270
x'820'	*Thermal Sensor Interrupt Temperature Register 1 (TS_ITR1)*	64	R/W	*Section 11.6.5* on page 271
x'828'	*Thermal Sensor Interrupt Temperature Register 2 (TS_ITR2)*	64	R/W	*Section 11.6.6* on page 273
x'830'	*Thermal Sensor Global Interrupt Temperature Register (TS_GITR)*	64	R/W	*Section 11.6.7* on page 274
x'838'	*Thermal Sensor Interrupt Status Register (TS_ISR)*	64	R/W	*Section 11.6.8* on page 275
x'840'	*Thermal Sensor Interrupt Mask Register (TS_IMR)*	64	R/W	*Section 11.6.9* on page 276
x'848'	*Thermal Management Control Register 1 (TM_CR1)*	64	R/W	*Section 11.6.10* on page 278
x'850'	*Thermal Management Control Register 2 (TM_CR2)*	64	R/W	*Section 11.6.11* on page 279
x'858'	*Thermal Management System Interrupt Mask Register (TM_SIMR)*	64	R/W	*Section 11.6.12* on page 280
x'860'	*Thermal Management Throttle Point Register (TM_TPR)*	64	R/W	*Section 11.6.13* on page 281
x'868'	*Thermal Management Stop Time Register 1 (TM_STR1)*	64	R/W	*Section 11.6.14* on page 283
x'870'	*Thermal Management Stop Time Register 2 (TM_STR2)*	64	R/W	*Section 11.6.15* on page 284
x'878'	*Thermal Management Throttle Scale Register (TM_TSR)*	64	R/W	*Section 11.6.16* on page 285
Time Base Registers				
x'890'	*Time Base Register (TBR)*	64	R/W	*Section 11.6.17* on page 286

11.2 Fault Isolation Registers

11.2.1 Global Fault Isolation Register (checkstop_fir)

After this checkstop_fir register is written, the bits cannot be cleared except by rebooting the chip. The checkstop_fir register collects all uncorrectable (clock stop) errors received from the local units.

Register Short Name	checkstop_fir	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509C00’	**Memory Map Area**	Pervasive: RAS
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TCU

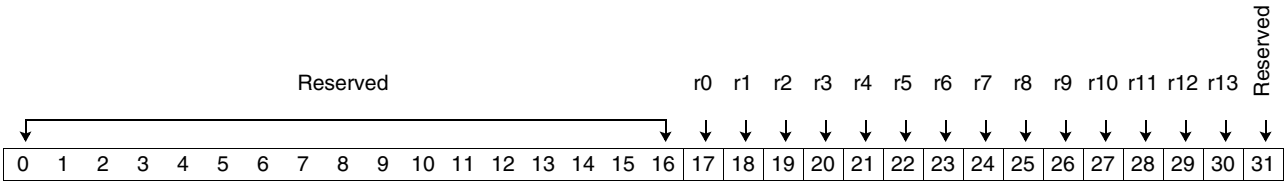


Bits	Field Name	Description
0:12	Reserved	Bits are not implemented; all bits read back zero.
13	Reserved	Reserved. Software should ignore value read and write only zero.
14	c1	PowerPC Processor Element (PPE)-PowerPC Processor Unit (PPU)
15	c2	Level 2 (L2) Cache
16	c3	Bus Interface Unit (BIU)
17	c4	Core Interface Unit (CIU)
18	c5	Memory Interface Controller (MIC)
19	c6	Bus Interface Controller (BIC), I/O Command (IOC)
20	c7	Synergistic Processor Element (SPE) 0
21	c8	SPE 1
22	c9	SPE 2
23	c10	SPE 3
24	c11	SPE 4
25	c12	SPE 5
26	c13	SPE 6
27	c14	SPE 7
28	c15	External Checkstop C4 pin
29	c16	Checkstop on trigger
30	c17	Any Local Recoverable Error Counter
31	c18	Checkstop by “CBE is quiesced”

11.2.2 Global Fault Isolation Register for Recoverable Errors (recoverable_fir)

The recoverable_fir register collects all correctable errors received from the local units. Recovery actions are executed by the units. This global register is updated by the local partitions.

Register Short Name	recoverable_fir	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509C08’	**Memory Map Area**	Pervasive: RAS
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	TCU

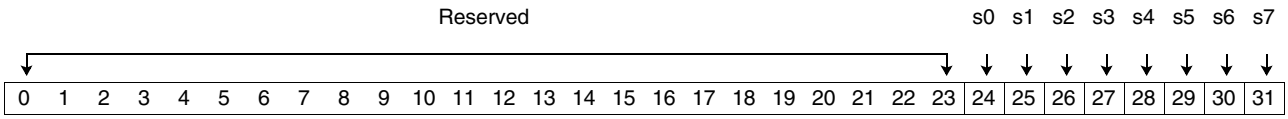


Bits	Field Name	Description
0:16	Reserved	Bits are not implemented; all bits read back zero.
17	r0	PPE-PPU
18	r1	L2
19	r2	BIU
20	r3	CIU
21	r4	MIC
22	r5	BIC, IOC
23	r6	SPE0
24	r7	SPE1
25	r8	SPE2
26	r9	SPE3
27	r10	SPE4
28	r11	SPE5
29	r12	SPE6
30	r13	SPE7
31	Reserved	Reserved. Software should ignore value read and write only zero.

11.2.3 SPE Available Partial Good Register (SPE_available)

The SPE_available register bits are read from the pervasive section of the configuration ring at POR. This register specifies whether any SPEs are disabled on the CBE chip.

Register Short Name	spe_available	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509C38’	**Memory Map Area**	Pervasive: RAS
Value at Initial POR	N/A	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	TCU

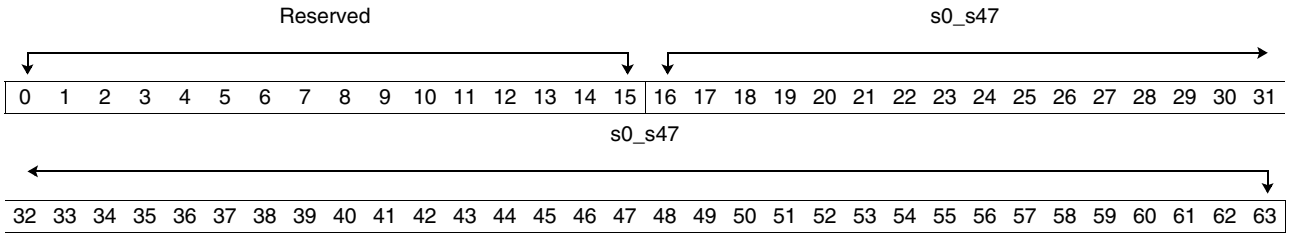


Bits	Field Name	Description
0:23	Reserved	Bits are not implemented; all bits read back zero.
24	s0	SPE0 available, partial good 0 SPU is disabled 1 SPU is available
25	s1	SPE1 available, partial good 0 SPU is disabled 1 SPU is available
26	s2	SPE2 available, partial good 0 SPU is disabled 1 SPU is available
27	s3	SPE3 available, partial good 0 SPU is disabled 1 SPU is available
28	s4	SPE4 available, partial good 0 SPU is disabled 1 SPU is available
29	s5	SPE5 available, partial good 0 SPU is disabled 1 SPU is available
30	s6	SPE6 available, partial good 0 SPU is disabled 1 SPU is available
31	s7	SPE7 available, partial good 0 SPU is disabled 1 SPU is available

11.2.4 Serial Number (serial_number)

This is a 48-bit serial number.

Register Short Name	serial_number	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509C80’	**Memory Map Area**	Pervasive: RAS
Value at Initial POR	x‘0000SSSS_SSSSSSSS’ (S = s0_s47)	**Value During POR Set By**	Configuration ring
Specification Type	Implementation-specific register	**Unit**	TCU



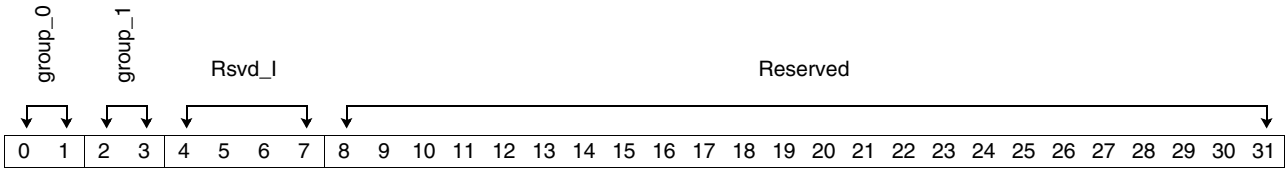
Bits	Field Name	Description
0:15	Reserved	Bits are not implemented; all bits read back zero.
16:63	s0_s47	48-bit customer ID

11.3 Performance Monitor Registers Shared with the Trace Logic Analyzer

11.3.1 Group Control Register (group_control)

Group 0 selects which debug bus word goes to the counter input multiplexer (mux) bits [0–31]. Group 1 selects which debug bus word goes to the counter input multiplexer bits [32–63]. See the debug_bus_control register for determination of which units are connected to word 0 through word 3.

Register Short Name	group_control	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509008’	**Memory Map Area**	Pervasive: Trace Logic Analyzer
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

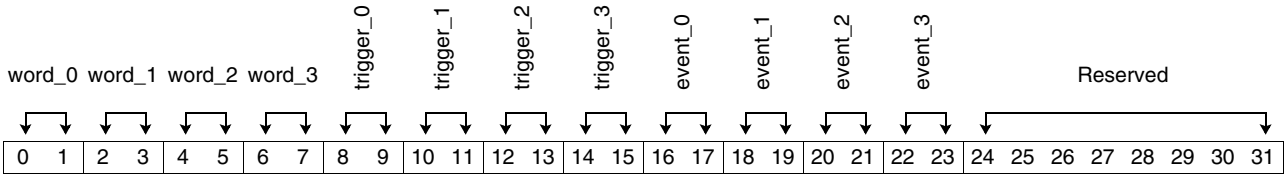


Bits	Field Name	Description
0:1	group_0	Counter input mux bits[0:31] Select the island’s word (as determined by the debug_bus_control register) to feed the performance monitor bus bits[0:31]. 00 word 0 01 word 1 10 word 2 11 word 3
2:3	group_1	Counter input mux bits[32:63] Select the sel_data segment (as determined by the debug_bus_control register) to feed the performance monitor bus bits[32:63]. 00 word 0 01 word 1 10 word 2 11 word 3
4:7	Rsvd_I	Used only by the Trace Logic Analyzer; all bits read back zero
8:31	Reserved	Bits are not implemented; all bits read back zero.

11.3.2 Debug Bus Control Register (debug_bus_control)

This register is shared with the TLA. This register selects which set of units can have their signals counted in the performance monitor.

Register Short Name	debug_bus_control	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x'5090A8'	**Memory Map Area**	Pervasive: Trace Logic Analyzer
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



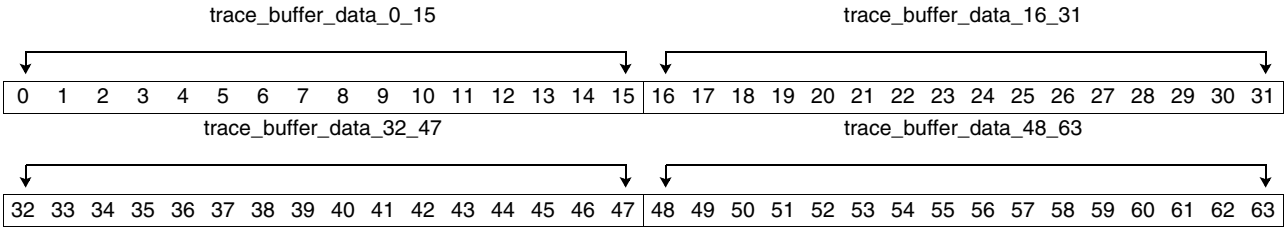
Bits	Field Name	Description
0:1	word_0	00 Full frequency PPU or MFC 01 Full frequency SPU (word 0) 10 Half frequency 11 Reserved
2:3	word_1	Ignored if word 0 = SPU 00 Full frequency; PPU or MFC 01 Reserved 10 Half frequency 11 Reserved
4:5	word_2	00 Full frequency PPU or MFC 01 Full frequency SPU (word 1) 10 Half frequency 11 Reserved
6:7	word_3	Ignored if word 2 = SPU 00 Full frequency; PPU or MFC 01 Reserved 10 Half frequency 11 Reserved
8:9	trigger_0	00 Full frequency PPU or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
10:11	trigger_1	00 Full frequency PPU or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
12:13	trigger_2	00 Full frequency PPU or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved

Bits	Field Name	Description
14:15	trigger_3	00 Full frequency PPU or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
16:17	event_0	00 Full frequency PPU or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
18:19	event_1	00 Full frequency PPC core or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
20:21	event_2	00 Full frequency PPC core or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
22:23	event_3	00 Full frequency PPC core or MFC 01 Full frequency SPU 10 Half frequency 11 Reserved
24:31	Reserved	Bits are not implemented; all bits read back zero

11.3.3 Trace Buffer High Doubleword Register (0 to 63) (trace_buffer_high)

This register is shared with the TLA.

Register Short Name	trace_buffer_high	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509108'	**Memory Map Area**	Pervasive: Trace Logic Analyzer
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

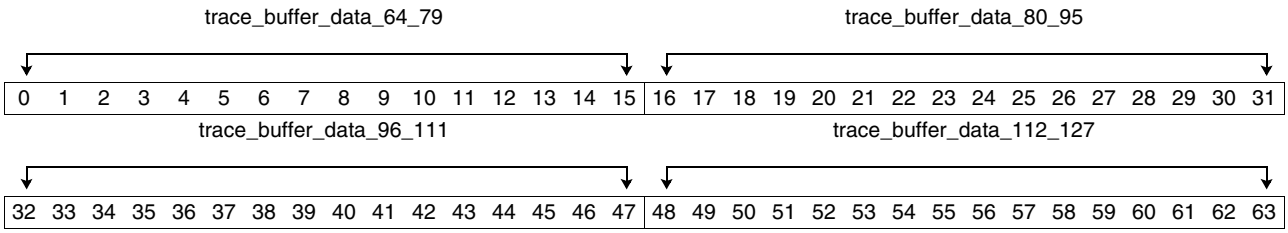


Bits	Field Name	Description
0:15	trace_buffer_data_0_15	PM0 when configured as 16-bit counters; PM0[0:15] when configured as a 32-bit counter.
16:31	trace_buffer_data_16_31	PM4 when configured as 16-bit counters; PM0[16:31] when configured as a 32-bit counter.
32:47	trace_buffer_data_32_47	PM1 when configured as 16-bit counters; PM1[0:15] when configured as a 32-bit counter.
48:63	trace_buffer_data_48_63	PM5 when configured as 16-bit counters; PM1[16:31] when configured as a 32-bit counter.

11.3.4 Trace Buffer Low Doubleword Register (64 to 127) (trace_buffer_low)

This register is shared with the TLA. The read address is incremented on a read of this register, so this register must be read with a 64-bit MMIO read.

Register Short Name	trace_buffer_low	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509110'	**Memory Map Area**	Pervasive: Trace Logic Analyzer
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



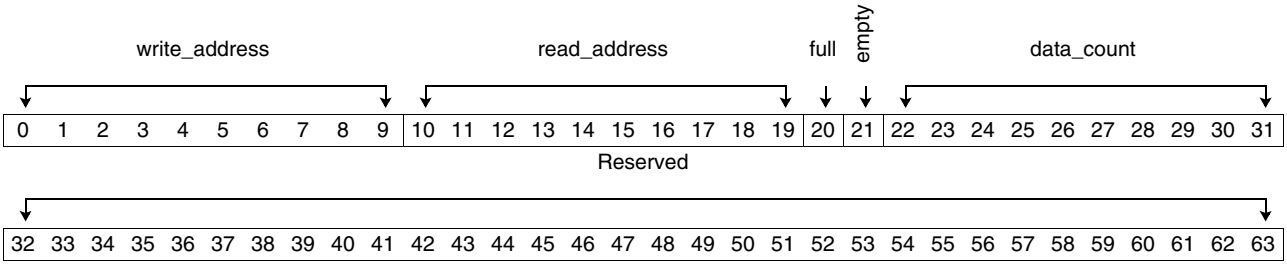
Bits	Field Name	Description
0:15	trace_buffer_data_64_79	PM2 when configured as 16-bit counters; PM2[0:15] when configured as a 32-bit counter.
16:31	trace_buffer_data_80_95	PM6 when configured as 16-bit counters; PM2[16:31] when configured as a 32-bit counter.
32:47	trace_buffer_data_96_111	PM3 when configured as 16-bit counters; PM3[0:15] when configured as a 32-bit counter.
48:63	trace_buffer_data_112_127	PM7 when configured as 16-bit counters; PM3[16:31] when configured as a 32-bit counter.

11.3.5 Trace Address Register (trace_address)

This register is shared with the TLA.

The trace_address register must be written to zero before enabling the performance monitor in performance data trace modes. Since the performance-monitor logic controls the trace arrays as a hardware FIFO during operation, it is unnecessary and undesirable to write the trace address registers. When in the trace_buffer_overwrite mode, the write and read pointers wrap when the maximum count is reached. A read from the FIFO with no data in the FIFO returns invalid data and sets a trace buffer underflow interrupt condition. See *Section 11.4.1 Performance Monitor Status/Interrupt Mask Register (pm_status)* on page 251.

Register Short Name	trace_address	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509118'	**Memory Map Area**	Pervasive: Trace Logic Analyzer
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

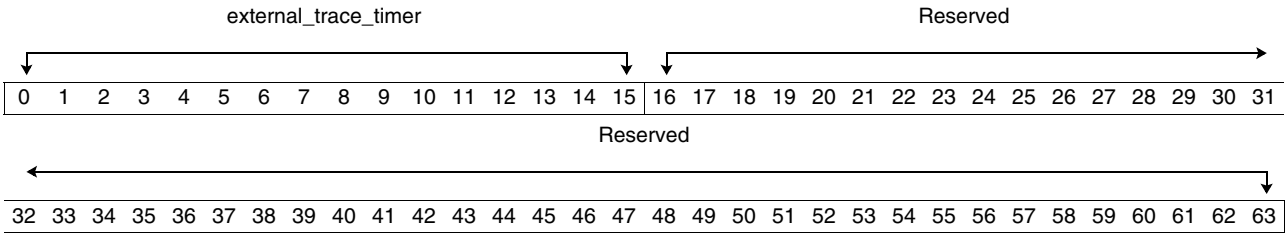


Bits	Field Name	Description
0:9	write_address	Address for trace array writing.
10:19	read_address	Address for trace array reading.
20	full	Trace array full (read only)
21	empty	Trace array empty (read only)
22:31	data_count	Count of trace array addresses containing valid data (read only)
32:63	Reserved	Bits are not implemented; all bits read back zero

11.3.6 External Trace Timer Register (ext_tr_timer)

The external trace timer sets the upper limit for data to be transferred from the trace buffer.

Register Short Name	ext_tr_timer	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509120’	**Memory Map Area**	Pervasive: Trace Logic Analyzer
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



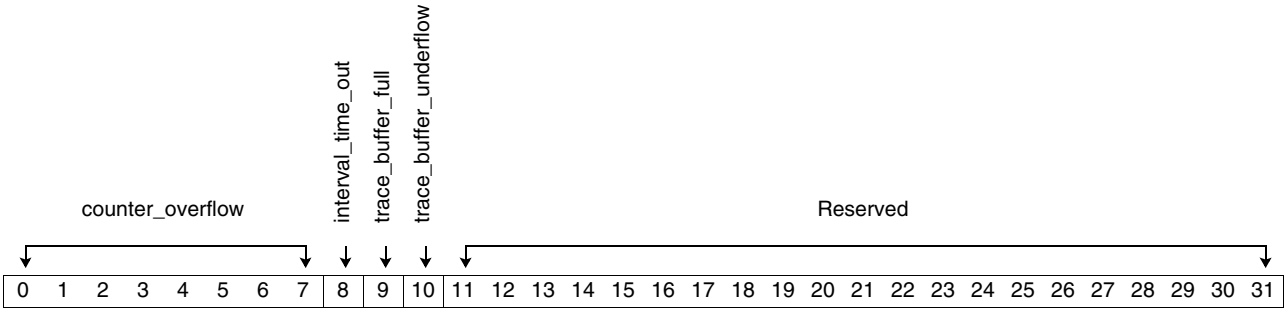
Bits	Field Name	Description
0:15	external_trace_timer	The external trace timer sets the upper limit for data to be transferred from the trace buffer. These 16 bytes are burst over the auxiliary bus one byte per NClk/2 cycles to external memory. The timer is a 16-bit binary up-counter that counts NClk/2 cycles. The external trace timer should be initialized to a value of 2^{16} - 1 - N, where N is the number of NClk/2 cycles between the 16-byte trace array reads. The maximum effective transfer rate in GB/s = 16 bytes/(N × NClk/2). A value of x‘FFEF’ yields the maximum rate of 1/(NClk/2) GB/s.
16:63	Reserved	Bits are not implemented; all bits read back zero.

11.4 Performance Monitor Only Registers

11.4.1 Performance Monitor Status/Interrupt Mask Register (pm_status)

This is a dual-function register. When the register is written, a '1' enables an associated interrupt, and a '0' disables the associated interrupt. Reading this register clears the status bits and resets any pending associated performance-monitor interrupt. (Status read, mask write).

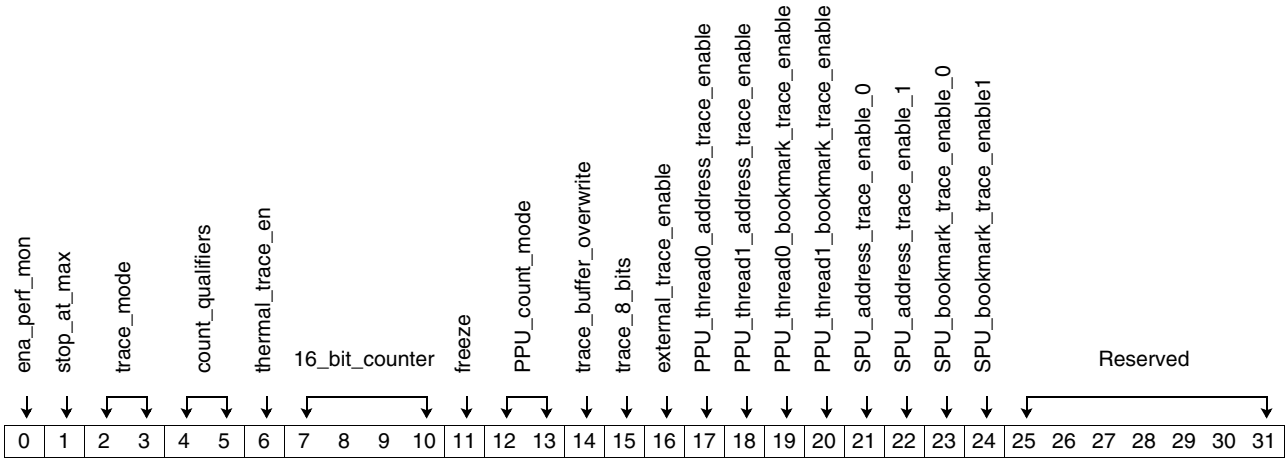
Register Short Name	pm_status	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x'509400'	**Memory Map Area**	Pervasive: Performance Monitor
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:7	counter_overflow	Counter [n] overflowed since the last read of this register. There is one bit for each of the eight performance monitor counters.
8	interval_time_out	The interval timer overflowed since the last read of this register.
9	trace_buffer_full	The trace buffer filled since the last read of this register.
10	trace_buffer_underflow	This bit is set when an MMIO read occurred from an empty local trace buffer since the last read of this register.
11:31	Reserved	Bits are not implemented; all bits read back zero

11.4.2 Performance Monitor Control Register (pm_control)

Register Short Name	pm_control	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509408’	**Memory Map Area**	Pervasive: Performance Monitor
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



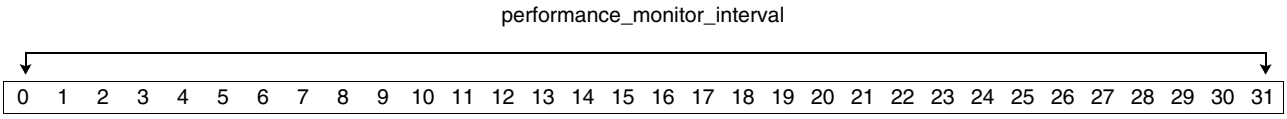
Bits	Field Name	Description
0	ena_perf_mon	Enable performance monitor 0 Disable 1 Enable The performance monitor shares resources with the trace logic analyzer, therefore software should enable only the performance monitor or the trace logic analyzer, not both at the same time.
1	stop_at_max	Stop at maximum 0 Disable. Run the performance monitor counters through the maximum count (wrap). 1 Enable. Perform counter overflow occurrence tracing.
2:3	trace_mode	Trace mode. Pack performance monitor data into 128 bits at each performance monitor interval timeout. Store the 128 bits to the trace buffer. The pm_interval must not be read through the MMIO when in trace modes ‘01’ and ‘11’. 00 No trace. Do not store performance-monitor data to the trace buffer. 01 Count trace. Store performance-monitor counter values to the trace buffer. The stop_at_max bit must be set to ‘1’ and the counter initial value set to zero (x‘FF00’ for counting 8 bits of a 16-bit counter mode). 10 Occurrence trace. Store 64 bits of performance-monitor bus occurrence data to the trace buffer. 11 Threshold trace. Store 8 bits of performance monitor counter overflow data to the trace buffer. The stop_at_max bit must be set to ‘1’. Counter freeze is not supported in this mode (bit[11] must be ‘0’).

Bits	Field Name	Description
4:5	count_qualifiers	Count qualifiers 00 Do not use count qualifiers. 01 Allow the start of counting upon PM0 timeout; PM4 may not be used as a normal counter. 10 Allow the stop of counting upon PM4 timeout; PM0 may not be used as a normal counter. 11 Allow the start of counting upon PM0 timeout and stop upon PM4 timeout. When PM0 or PM4 are enabled as count qualifiers, they must be configured as two 16-bit counters. They stop at the maximum count, regardless of the state of the stop_at_max pm_control bit. PM1, PM2, PM3, PM5, PM6, and PM7 are subject to count qualification. After the stop count qualifier count reaches maximum count, all counters are held. The performance monitor must be disabled and count qualifiers reinitialized to use the count qualifiers again. If in count trace mode, the start and stop of count tracing is subject to the count qualifiers.
6	thermal_trace_en	Thermal trace enable 0 Disable 1 Enable. Enable thermal data to override data going to the trace array on bits[64:123]. The trace bits[0:63] might contain header and SPU address information per the trace data formats. 8-bit count tracing, occurrence tracing, and threshold tracing are not useful in conjunction with thermal tracing.
7:10	16_bit_counter	16-bit counter for pm0_4, pm1_5, pm2_6, pm3_7. For each bit: 0 Disable. Configure the counter to act as a 32-bit counter 1 Enable. Configure the counter to act as two 16-bit counters
11	freeze	Freeze all counters on overflow 0 Disable. Do not freeze. 1 Enable. Freeze all counters (including the interval timer) on any counter overflow, except the interval timer. When counters PM0 and PM4 are configured as count qualifiers, their overflowing does not cause a freeze. Freezing counts is not supported for the occurrence tracing and threshold tracing modes (see trace bits[2,3]).
12:13	PPU_count_mode	PPU count mode. Count PPU performance monitor signals only if the signal occurs while the PPU is in any of the following modes: 00 Supervisor mode 01 Hypervisor mode 10 Problem mode 11 Any of the above modes
14	trace_buffer_overwrite	Trace buffer overwrite 0 Disable. Do not overwrite. Performance monitor data is written until the trace buffer is full. When external trace is disabled, this provides a means to record performance data for the subsequent 1024 time intervals following a start event. Data may be read out of the trace buffer by setting the pm_control trace bits[2:3] to zero to prevent further writing to the buffer and then reading out the performance data from the trace buffer. When external trace is enabled, this allows for the use of the trace buffer as a FIFO for speed matching the trace buffer input and output data rates. 1 Enable. Overwrite trace buffer data when the trace buffer is full. This provides for continuous writing of counts to the buffer until the occurrence of a stop event. The data from the most recent 1024 count intervals can then be read out of the trace buffer. After overwriting, because the read pointer no longer points to the oldest data, the values of the write pointer can be used to determine the sequential order of the data. Do not set this bit if external trace is enabled.
15	trace_8_bits	Trace 8 bits of 16-bit counters 0 Disable 1 Enable. Trace only the lower 8 bits of the counters if the performance monitor is set for 16-bit count tracing. In this case, the counter must be initialized to x‘FF00’. The 64-bit count data is written to the trace arrays (bits[64:127]) with the 8-bit count values in the following order: v0_v2_v4_v6_v1_v3_v5_v7.
16	external_trace_enable	Trace buffer output (enable external trace) 0 Disable. Count read from trace buffer through the MMIO. 1 Enable. Performs external trace through BIC or MIC. MMIO reads from the trace buffer are not allowed when external trace is enabled.

Bits	Field Name	Description
17	PPU_thread0_address_trace_enable	PPU thread 0 address trace enable 0 Disable 1 Enable. Enable PPU thread 0 branch address for recent specific branch and link instructions to be stored to the trace buffer, in addition to the counter data stored at specified intervals. When the PPU address or a bookmark is enabled, then trace bits[0:15] are overwritten with header information.
18	PPU_thread1_address_trace_enable	PPU thread 1 address trace enable 1 Enable. Enable PPU thread 1 branch address for recent specific branch and link instructions to be stored to the trace buffer in addition to the counter data stored at specified intervals. When the PPU address or a bookmark is enabled, then trace bits[0:15] are overwritten with header information.
19	PPU_thread0_bookmark_trace_enable	PPU thread 0 bookmark trace enable 0 Disable 1 Enable. Enable PPU thread 0 bookmark SPR write data to be stored to the trace buffer, in addition to the counter data stored at specified intervals. When the PPU address or a bookmark is enabled, then trace bits[0:15] are overwritten with header information.
20	PPU_thread1_bookmark_trace_enable	PPU thread 1 bookmark trace enable 0 Disable 1 Enable. Enable PPU thread 1 bookmark SPR write data to be stored to the trace buffer, in addition to the counter data stored at specified intervals. When the PPU address or a bookmark is enabled, then trace bits[0:15] are overwritten with header information.
21	SPU_address_trace_enable_0	SPU address trace enable 0 0 Disable 1 Enable. Overwrite trace bits[16:31] with the most recently received SPU program counter (address). Setup in the SPUx must be performed to route that particular SPU's address to the debug_bus event bit[1].
22	SPU_address_trace_enable_1	SPU address trace enable 1 0 Disable 1 Enable. Overwrite trace bits[32:47] with the most recently received SPU program counter (address). Setup in the SPUx must be performed to route that particular SPU's address to the debug_bus event bit[3].
23	SPU_bookmark_trace_enable_0	SPU bookmark trace enable 0 0 Disable 1 Enable. Store an SPU bookmark record to the trace buffer. Trace bits[0:15] are overwritten with header information and bits[16:31] with the most recently received SPU bookmark value. Setup in the SPUx must be performed to route that particular SPU's bookmark to the debug_bus event bit[1].
24	SPU_bookmark_trace_enable1	SPU bookmark trace enable1 0 Disable 1 Enable. Store an SPU bookmark record to the trace buffer. Trace bits[0:15] are overwritten with header information and bits[32:47] with the most recently received SPU bookmark value. Setup in the SPUx must be performed to route that particular SPU's bookmark to the debug_bus event bit[3].
25:31	Reserved	Bits are not implemented; all bits read back zero.

11.4.3 Performance Monitor Interval Register (pm_interval)

Register Short Name	pm_interval	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509410’	**Memory Map Area**	Pervasive: Performance Monitor
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

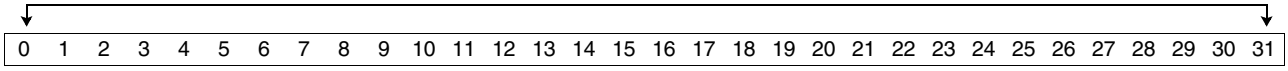


Bits	Field Name	Description
0:31	performance_monitor_interval	Performance-monitor interval in core clock cycles (minimum interval is 10 cycles). This can be used in a trace mode to time the interval for writes to the trace buffer. This is a 32-bit binary up-counter. To program for an interval of N cycles, this counter must be initialized to a value of 2^{32} - N - 1. (For example, x‘FFFF FFF5’ gives an interval of 10 cycles). Write the initial value latch or read the current count through this MMIO address. The performance monitor interval timer must be initialized prior to enabling and reenabling the performance monitor when using the count, threshold, and occurrence trace modes. **Note:** The total performance data rate, including potential address trace, must be lower than the trace buffer output rate (as determined by the external trace timer) to prevent the overflow of the trace buffer. **Note:** The pm_interval is undefined when trace_mode (pm_control[2:3]) is set to ‘01’ or ‘11’.

11.4.4 Performance Monitor Counter Pairs Registers (pmM_N)

Register Short Name	pm0_4 pm1_5 pm2_6 pm3_7	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509418’ x‘509420’ x‘509428’ x‘509430’	**Memory Map Area**	Pervasive: Performance Monitor
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

performance_monitor_counter

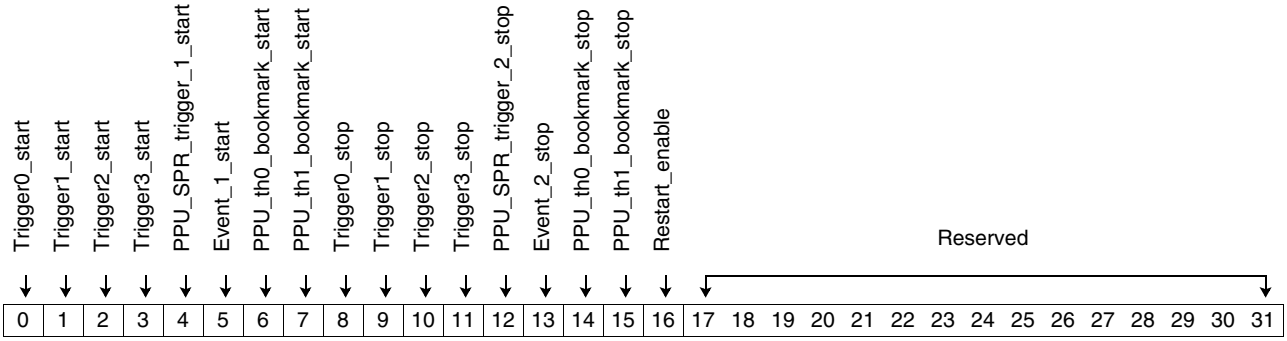


Bits	Field Name	Description
0:31	performance_monitor_counter	32-bit count value or two 16-bit count values Performance-monitor counters 0, 1, 2, and 3 are read or written in either bits[0:15] or bits[0:31]. In 16-bit counter mode, performance-monitor counters 4, 5, 6, and 7 are read or written in bits [16:31]. In the count_trace or occurrence modes of counting, the performance-monitor counters must be written before the performance monitor is enabled, and values must not be read through MMIO. These are binary up-counters, therefore the value read indicates the number of events counted. For 8-bit occurrence counting, where N is the threshold value, the counter should be initialized to 2^{16} - 1 - N where N > 0. For a 32-bit counter, this would be 2^{32} - 1 - N where N > 0. Write the initial value latch or read the current count through this MMIO address.

11.4.5 Performance Monitor Start Stop Register (pm_start_stop)

Start and stop for performance monitor counters. The start condition is an OR function of the start bits. The stop condition is an OR function of the stop bits. If the start qualifier is turned on, then the start signals act as a prequalifier to the start count qualifier. If the stop qualifier is turned on, then the stop signals act as a prequalifier to the stop count qualifier. The restart_enable signal allows a prequalifier start after a prequalifier stop.

Register Short Name	pm_start_stop	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x'509438'	**Memory Map Area**	Pervasive: Performance Monitor
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



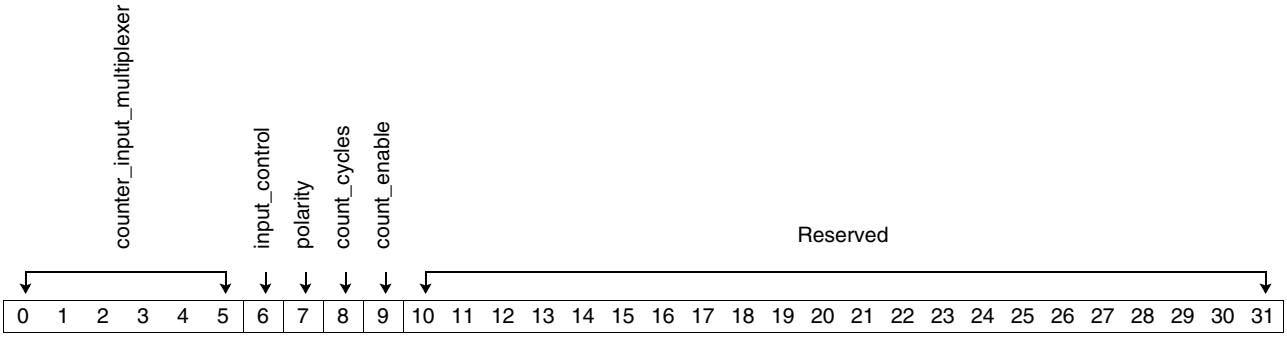
Bits	Field Name	Description
0	Trigger0_start	Start counting upon debug bus trigger 0.
1	Trigger1_start	Start counting upon debug bus trigger 1.
2	Trigger2_start	Start counting upon debug bus trigger 2.
3	Trigger3_start	Start counting upon debug bus trigger 3.
4	PPU_SPR_trigger_1_start	Start counting upon PPU SPR trigger 1.
5	Event_1_start	Start counting upon debug bus event 1.
6	PPU_th0_bookmark_start	Start counting upon PPU thread 0 (th0) bookmark start (req. bookmark enabled in pm_control).
7	PPU_th1_bookmark_start	Start counting upon PPU thread 1 (th1) bookmark start (req. bookmark enabled in pm_control).
8	Trigger0_stop	Stop counting upon debug bus trigger 0.
9	Trigger1_stop	Stop counting upon debug bus trigger 1.
10	Trigger2_stop	Stop counting upon debug bus trigger 2.
11	Trigger3_stop	Stop counting upon debug bus trigger 3.
12	PPU_SPR_trigger_2_stop	Stop counting upon PPU SPR trigger 2.
13	Event_2_stop	Stop counting upon debug bus event 2.

Bits	Field Name	Description
14	PPU_th0_bookmark_stop	Stop counting upon PPU th0 bookmark stop (req. bookmark enabled in pm_control).
15	PPU_th1_bookmark_stop	Stop counting upon PPU th1 bookmark stop (req. bookmark enabled in pm_control).
16	Restart_enable	Allows prequalifier start after a prequalifier stop; requires at least one start and one stop prequalifier to be set and the count qualifier (cq) start and cq stop (pm_control[4:5]) disabled.
17:31	Reserved	Bits are not implemented; all bits read back zero.

11.4.6 Performance Monitor Counter Control Registers (pmN_control)

These are the performance monitor counter control registers that allow control per counter. This is in contrast with the pm_control register, which applies across the entire performance monitor facility.

Register Short Name	pm0_control pm1_control pm2_control pm3_control pm4_control pm5_control pm6_control pm7_control	**Privilege Type**	Privilege 1
Access Type	MMIO Write Only	**Width**	32 bits
Hex Offset From BE_MMIO_Base	x‘509440’ x‘509448’ x‘509450’ x‘509458’ x‘509460’ x‘509468’ x‘509470’ x‘509478’	**Memory Map Area**	Pervasive: Performance Monitor
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

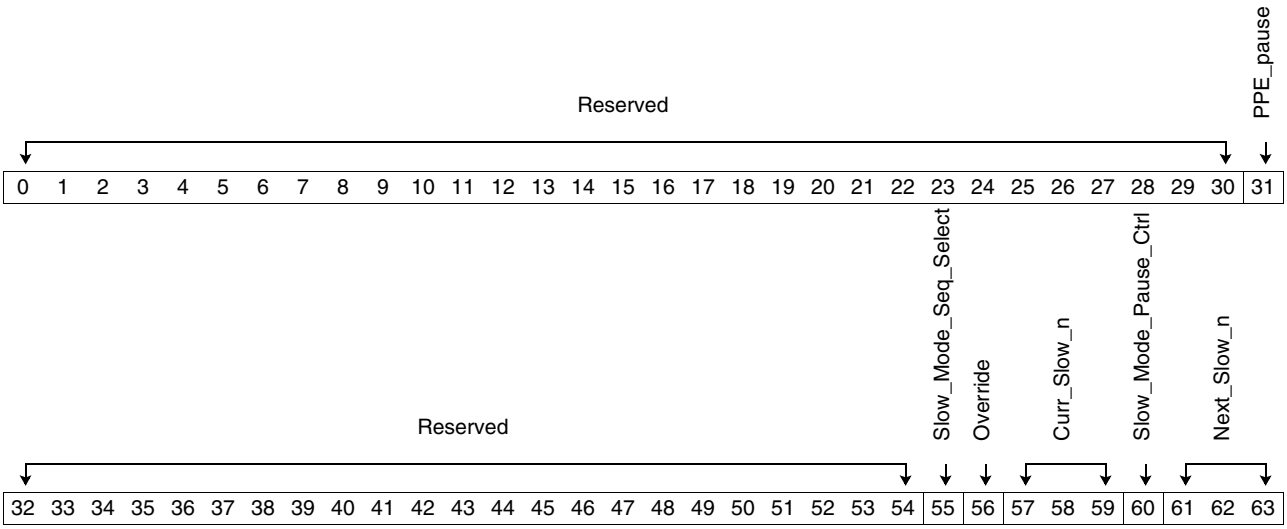


Bits	Field Name	Description
0:5	counter_input_multiplexer	Counter Input Multiplexer 000000 Input is performance monitor bus bit 0 or trigger 0. 000001 Input is performance monitor bus bit 1 or trigger 1. 000010 Input is performance monitor bus bit 2 or trigger 2. 000011 Input is performance monitor bus bit 3 or trigger 3. 000100 Input is performance monitor bus bit 4 or event 0. 000101 Input is performance monitor bus bit 5 or event 1. 000110 Input is performance monitor bus bit 6 or event 2. 000111 Input is performance monitor bus bit 7 or event 3. 001000 Input is performance monitor bus bit 8 or external_trigger_in. 001001 Input is performance monitor bus bit 9 or spr_trigger1. 001010 Input is performance monitor bus bit 10 or spr_trigger2. 001011 Input is performance monitor bus bit 11. . . . The contents of the performance monitor bus are determined by the settings of the group_control and the debug_bus_control registers. In order to count cycles regardless of any input signal, event or trigger, set counter_input_multiplexer (bits[0:5]) to '010000', input_control (bit[6]) to '1', polarity (bit[7]) =0. Also, see bit[6] (input_control).
6	input_control	Input control 0 Input is a performance-monitor bus bit selected by the counter_input_multiplexer bits. 1 Input is the alternative trigger or event selected by the counter_input_multiplexer bits.
7	polarity	Polarity 0 Negative polarity. Count when 0, or, if counting edges, count negative transitions. 1 Positive polarity. Count when 1, or, if counting edges, count positive transitions.
8	count_cycles	Count cycles 0 Count edges 1 Count cycles
9	count_enable	Count enable 0 Disable this counter 1 Enable this counter
10:31	Reserved	Bits are not implemented; all bits read back zero

11.5 Power Management Control Registers

11.5.1 Power Management Control Register (PMCR)

Register Short Name	PMCR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509880’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

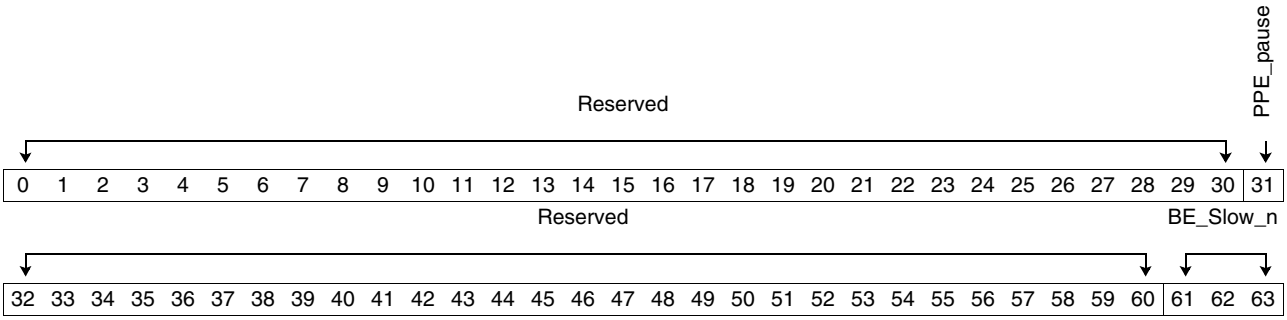


Bits	Field Name	Description
0:30	Reserved	Bits are not implemented; all bits read back zero.
31	PPE_pause	Enables the PPE Pause (0) state. 0 Disable 1 Enable
32:54	Reserved	Bits are not implemented; all bits read back zero.
55	Slow_Mode_Seq_Select	Slow Mode Sequence Select 0 Transition frequencies in steps of 1/8 1 Transition frequencies in steps of 1/4
56	Override	Current Slow Mode Override 0 Slow mode 1 Overrides the current slow mode status stored in the hardware to the value in Curr_Slow(n) bits[57:59]. Hardware resets this bit to ‘0’ after the update is completed. **Note:** Setting this bit might indirectly cause a slow mode transition if the Curr_Slow(n) setting is different than the Next_Slow setting. If this is the case, hardware first updates the Slow_Mode status register to the Curr_Slow(n) value, and then changes to the Next_Slow bit value.

Bits	Field Name	Description
57:59	Curr_Slow_n	Current CBE Slow (n) State Controls the current CBE Slow (n) state. These bits encode the CBE core frequencies. **Value** **n** **CBE Core Frequency** (NClk) 000 0 Max 001 1 Max/2 010 2 Max/3 011 3 Max/4 100 4 Max/5 101 5 Max/6 110 6 Max/8 111 7 Max/10 **Note:** These bit settings cannot change value until the corresponding bits in the PMSR Register match these bit settings. The minimum allowed frequency of the processor is 1.0 GHz whenever power is applied to the core. This is required for the long term reliability of the processor.
60	Slow_Mode_Pause_Ctrl	Slow Mode Pause Control 0 System switches to new slow mode setting without waiting for pause mode. 1 System waits for pause mode before switching to new slow mode setting.
61:63	Next_Slow_n	Next CBE Slow (n) State Controls the next Slow (n) state at the CBE level. This field encodes the CBE core frequencies. **Value** **n** **CBE Core Frequency** (NClk) 000 0 Max 001 1 Max/2 010 2 Max/3 011 3 Max/4 100 4 Max/5 101 5 Max/6 110 6 Max/8 111 7 Max/10 **Note:** These bit settings cannot change value until the corresponding bits in the PMSR Register match these bit settings. The minimum allowed frequency of the processor is 1.0 GHz whenever power is applied to the core. This is required for the long term reliability of the processor.

11.5.2 Power Management Status Register (PMSR)

Register Short Name	PMSR	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509888'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:30	Reserved	Bits are not implemented; all bits read back zero.
31	PPE_pause	Enables the PPE Pause (0) state.
32:60	Reserved	Bits are not implemented; all bits read back zero.
61:63	BE_Slow_n	CBE Slow (n) State Status Status for the CBE Slow (n) state. **Value** **n** **CBE Core Frequency** (NClk) 000 0 Max 001 1 Max/2 010 2 Max/3 011 3 Max/4 100 4 Max/5 101 5 Max/6 110 6 Max/8 111 7 Max/10

11.6 Thermal Management MMIO Registers

Thermal monitor control (TMC) logic is capable of generating a thermal interrupt. The thermal interrupt is defined in the *PowerPC Architecture* and has its dedicated interrupt vector.

The thermal sensor interrupt registers control the generation of a thermal management interrupt to the PPE. This set of registers consists of the thermal sensor interrupt temperature registers (TS_ITR1 and TS_ITR2), the Thermal Sensor Interrupt Status Register (TS_ISR), and the Thermal Sensor Interrupt Mask Register (TS_IMR).

Table 11-2. Encode to Temperature Mapping

Thermal Sensor Temperature Encoding			
6-bit Encode	Temperature Range	6-bit Encode	Temperature Range
0	Temp ≤ 65°C	16	95°C ≤ Temp < 97°C
1	65°C ≤ Temp < 67°C	17	97°C ≤ Temp < 99° C
2	67°C ≤ Temp < 69°C	18	99°C ≤ Temp < 101°C
3	69°C ≤ Temp < 71°C	19	101°C ≤ Temp < 103°C
4	71°C ≤ Temp < 73°C	20	103°C ≤ Temp < 105°C
5	73°C ≤ Temp < 75°C	21	105°C ≤ Temp < 107°C
6	75°C ≤ Temp < 77°C	22	107°C ≤ Temp < 109°C
7	77°C ≤ Temp < 79°C	23	109°C ≤ Temp < 111°C
8	79°C ≤ Temp < 81°C	24	111°C ≤ Temp < 113°C
9	81°C ≤ Temp < 83°C	25	113°C ≤ Temp < 115°C
10	83°C ≤ Temp < 85°C	26	115°C ≤ Temp < 117°C
11	85°C ≤ Temp < 87°C	27	117°C ≤ Temp < 119°C
12	87°C ≤ Temp < 89°C	28	119°C ≤ Temp < 121°C
13	89°C ≤ Temp < 91°C	29	121°C ≤ Temp < 123°C
14	91 °C ≤ Temp < 93°C	30	123°C ≤ Temp < 125°C
15	93°C ≤ Temp < 95°C	31 – 63	125°C ≤ Temp
Note: Refer to the *Cell Broadband Engine Datasheet* for the accuracy of the Digital Thermal Sensor.			

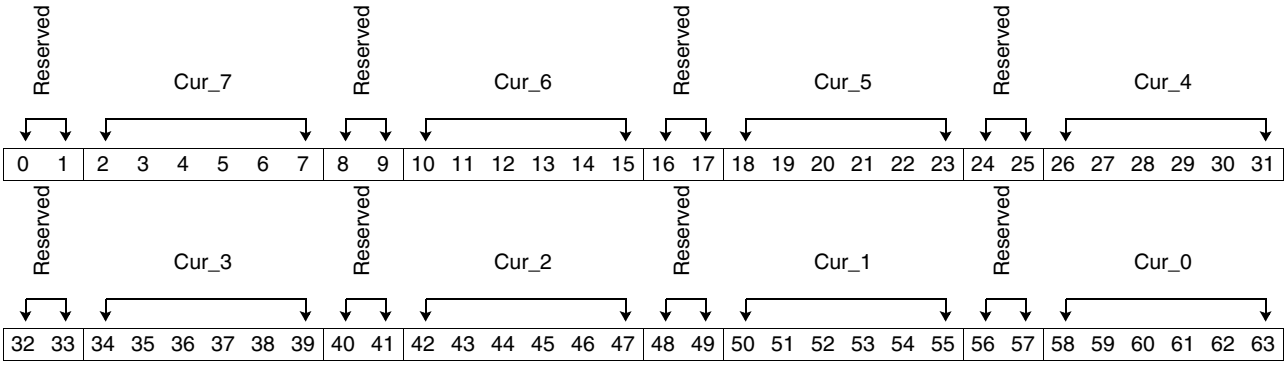
11.6.1 Thermal Sensor Current Temperature Status Register 1 (TS_CTSR1)

The TS_CTSR1 register contains the encoding for the current temperature of the sensors in the SPEs. The thermal sensor current temperature status registers (TS_CTSR1 and TS_CTSR2) contain the encoding for the current temperature of each DTS. Due to latencies in the sensor's temperature detection, latencies in reading these registers, and normal temperature fluctuations, the temperature reported in these registers is that of an earlier point in time and might not reflect the actual temperature when software receives the data.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [2:7]	Cur(7)	bits [34:39]	Cur(3)
bits [10:15]	Cur(6)	bits [42:47]	Cur(2)
bits [18:23]	Cur(5)	bits [50:55]	Cur(1)
bits [26:31]	Cur(4)	bits [58:63]	Cur(0)

Register Short Name	TS_CTSR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509800'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:7	Cur_7	Current temperature level digital thermal sensor 7. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 7.
8:9	Reserved	Bits are not implemented; all bits read back zero.
10:15	Cur_6	Current temperature level digital thermal sensor 6. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 6.
16:17	Reserved	Bits are not implemented; all bits read back zero.
18:23	Cur_5	Current temperature level digital thermal sensor 5. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 5.
24:25	Reserved	Bits are not implemented; all bits read back zero.

Bits	Field Name	Description
26:31	Cur_4	Current temperature level digital thermal sensor 4. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 4.
32:33	Reserved	Bits are not implemented; all bits read back zero.
34:39	Cur_3	Current temperature level digital thermal sensor 3. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 3.
40:41	Reserved	Bits are not implemented; all bits read back zero.
42:47	Cur_2	Current temperature level digital thermal sensor 2. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 2.
48:49	Reserved	Bits are not implemented; all bits read back zero.
50:55	Cur_1	Current temperature level digital thermal sensor 1. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 1.
56:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	Cur_0	Current temperature level digital thermal sensor 0. If the sensor is not tracking, it reports 0. This sensor is located in physical SPE 0.

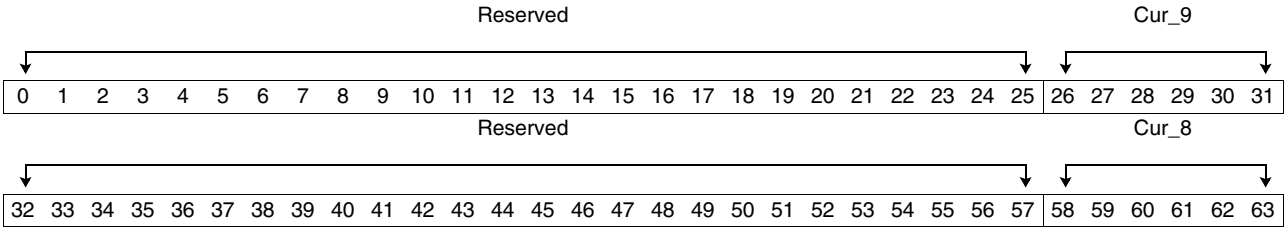
11.6.2 Thermal Sensor Current Temperature Status Register 2 (TS_CTSR2)

This register contains the encoding for the current temperature of the sensor in the PPE and the sensor located adjacent to the linear thermal sensor. The thermal sensor current temperature status registers (TS_CTSR1 and TS_CTSR2) contain the encoding for the current temperature of each DTS. Due to latencies in the sensor's temperature detection, latencies in reading these registers, and normal temperature fluctuations, the temperature reported in these registers is that of an earlier point in time and might not reflect the actual temperature when software receives the data.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [26:31] Cur(7)
bits [58:63] Cur(8)

Register Short Name	TS_CTSR2	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509808'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:25	Reserved	Bits are not implemented; all bits read back zero.
26:31	Cur_9	Current temperature level digital thermal sensor 9. This sensor is located adjacent to the linear thermal sensor. If the sensor is not tracking, it reports 0.
32:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	Cur_8	Current temperature level digital thermal sensor 8. This sensor is located in the PPU. If the sensor is not tracking, it reports 0.

The thermal sensor maximum temperature status registers (TS_MTSR1 and TS_MTSR2) contain the encoding for the maximum temperature reached for each sensor from the time of the last read of these registers. Reading these registers causes the TMCU to copy the current temperature for each sensor into the register. After the read, the TMCU continues to track the maximum temperature starting from this point. Each register is independent; a read of one register does not affect the contents of the other.

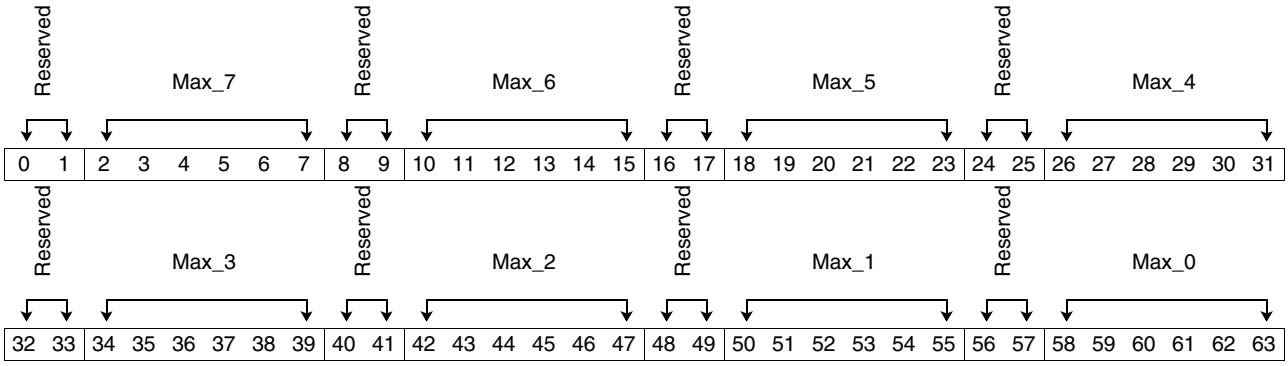
11.6.3 Thermal Sensor Maximum Temperature Status Register 1 (TS_MTSR1)

This register contains the encoding for the maximum temperature of the sensors located in the SPEs.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [2:7] Max(7) bits [34:39] Max(3)
bits [10:15] Max(6) bits [42:47] Max(2)
bits [18:23] Max(5) bits [50:55] Max(1)
bits [26:31] Max(4) bits [58:63] Max(0)

Register Short Name	TS_MTSR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509810'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:7	Max_7	Maximum temperature level reached by digital thermal sensor 7 from the time of the last read of this register. Digital thermal sensor 7 is located in physical SPE 7.
8:9	Reserved	Bits are not implemented; all bits read back zero.
10:15	Max_6	Maximum temperature level reached by digital thermal sensor 6 from the time of the last read of this register. Digital thermal sensor 6 is located in physical SPE 6.
16:17	Reserved	Bits are not implemented; all bits read back zero.
18:23	Max_5	Maximum temperature level reached by digital thermal sensor 5 from the time of the last read of this register. Digital thermal sensor 5 is located in physical SPE 5.
24:25	Reserved	Bits are not implemented; all bits read back zero.
26:31	Max_4	Maximum temperature level reached by digital thermal sensor 4 from the time of the last read of this register. Digital thermal sensor 4 is located in physical SPE 4.
32:33	Reserved	Bits are not implemented; all bits read back zero.
34:39	Max_3	Maximum temperature level reached by digital thermal sensor 3 from the time of the last read of this register. Digital thermal sensor 3 is located in physical SPE 3.

Bits	Field Name	Description
40:41	Reserved	Bits are not implemented; all bits read back zero.
42:47	Max_2	Maximum temperature level reached by digital thermal sensor 2 from the time of the last read of this register. Digital thermal sensor 2 is located in physical SPE 2.
48:49	Reserved	Bits are not implemented; all bits read back zero.
50:55	Max_1	Maximum temperature level reached by digital thermal sensor 1 from the time of the last read of this register. Digital thermal sensor 1 is located in physical SPE 1.
56:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	Max_0	Maximum temperature level reached by digital thermal sensor 0 from the time of the last read of this register. Digital thermal sensor 0 is located in physical SPE 0.

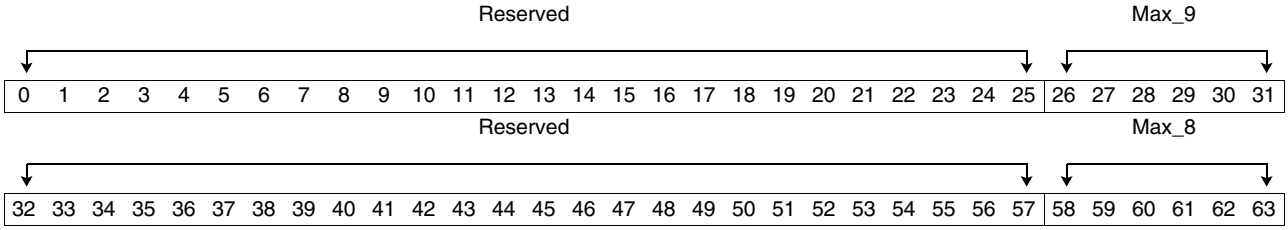
11.6.4 Thermal Sensor Maximum Temperature Status Register 2 (TS_MTSR2)

This register contains the encoding for the maximum temperature of the sensors located in the PPE and adjacent to the linear thermal sensor.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [26:31] Max(9)
bits [58:63] Max(8)

Register Short Name	TS_MTSR2	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509818’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:25	Reserved	Bits are not implemented; all bits read back zero.
26:31	Max_9	Maximum temperature level reached by digital thermal sensor 9 from the time of the last read of this register. Digital thermal sensor 9 is located adjacent to the linear thermal sensor.
32:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	Max_8	Maximum temperature level reached by digital thermal sensor 8 from the time of the last read of this register. Digital thermal sensor 8 is located in the PPU.

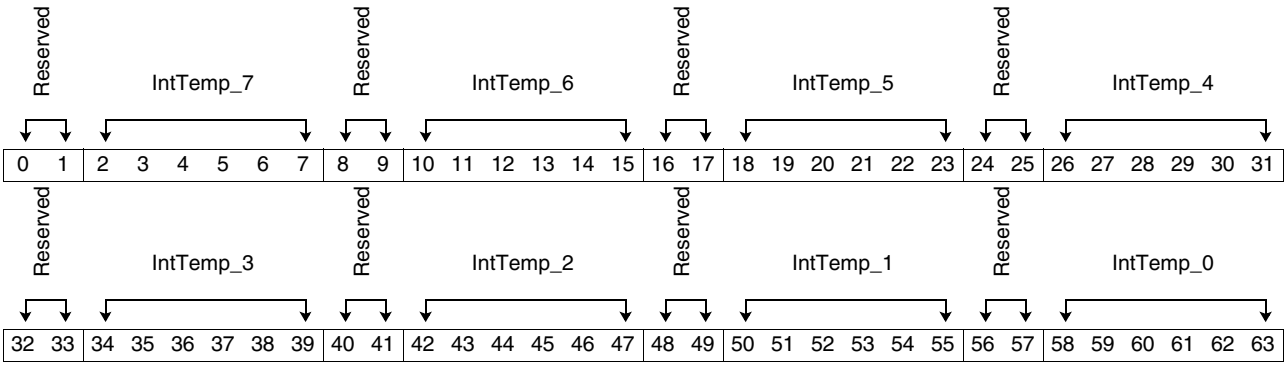
11.6.5 Thermal Sensor Interrupt Temperature Register 1 (TS_ITR1)

The TS_ITR1 register contains the interrupt temperature level for the sensors located in the SPEs. The encoded interrupt temperature levels in this register are compared to the corresponding interrupt temperature encoding in TS_CTSR1. The result of these comparisons is used to generate a thermal management interrupt. Each sensor's interrupt temperature level is independent.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [2:7]	IntTemp(7)	bits [34:39]	IntTemp(3)
bits [10:15]	IntTemp(6)	bits [42:47]	IntTemp(2)
bits [18:23]	IntTemp(5)	bits [50:55]	IntTemp(1)
bits [26:31]	IntTemp(4)	bits [58:63]	IntTemp(0)

Register Short Name	TS_ITR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509820'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:1	Reserved	Bits are not implemented; all bits read back zero.
2:7	IntTemp_7	Temperature level at which digital thermal sensor 7 causes an interrupt. Digital thermal sensor 7 is located in physical SPE 7.
8:9	Reserved	Bits are not implemented; all bits read back zero.
10:15	IntTemp_6	Temperature level at which digital thermal sensor 6 causes an interrupt. Digital thermal sensor 6 is located in physical SPE 6.
16:17	Reserved	Bits are not implemented; all bits read back zero.
18:23	IntTemp_5	Temperature level at which digital thermal sensor 5 causes an interrupt. Digital thermal sensor 5 is located in physical SPE 5.
24:25	Reserved	Bits are not implemented; all bits read back zero.
26:31	IntTemp_4	Temperature level at which digital thermal sensor 4 causes an interrupt. Digital thermal sensor 4 is located in physical SPE 4.

Bits	Field Name	Description
32:33	Reserved	Bits are not implemented; all bits read back zero.
34:39	IntTemp_3	Temperature level at which digital thermal sensor 3 causes an interrupt. Digital thermal sensor 3 is located in physical SPE 3.
40:41	Reserved	Bits are not implemented; all bits read back zero.
42:47	IntTemp_2	Temperature level at which digital thermal sensor 2 causes an interrupt. Digital thermal sensor 2 is located in physical SPE 2.
48:49	Reserved	Bits are not implemented; all bits read back zero.
50:55	IntTemp_1	Temperature level at which digital thermal sensor 1 causes an interrupt. Digital thermal sensor 1 is located in physical SPE 1.
56:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	IntTemp_0	Temperature level at which digital thermal sensor 0 causes an interrupt. Digital thermal sensor 0 is located in physical SPE 0.

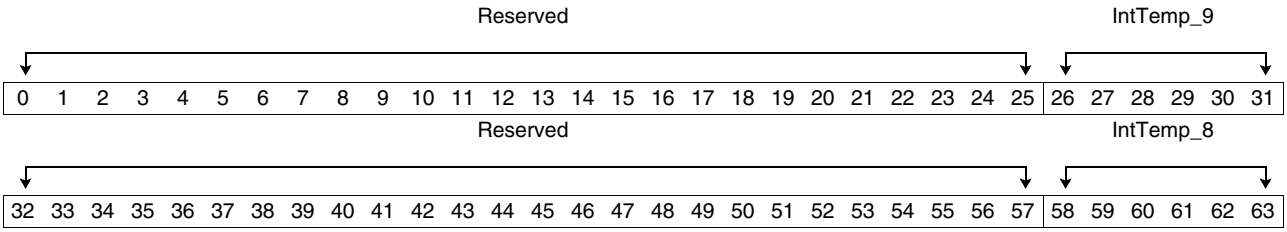
11.6.6 Thermal Sensor Interrupt Temperature Register 2 (TS_ITR2)

The TS_ITR2 register contains the interrupt temperature level for the sensors located in the PPE and adjacent to the linear thermal sensor. The encoded interrupt temperature levels in this register are compared to the corresponding interrupt temperature encoding in TS_CTSR2. The result of these comparisons is used to generate a thermal management interrupt. Each sensor's interrupt temperature level is independent.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [26:31] IntTemp(9)
bits [58:63] IntTemp(8)

Register Short Name	TS_ITR2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509828'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:25	Reserved	Bits are not implemented; all bits read back zero.
26:31	IntTemp_9	Temperature level at which digital thermal sensor 9 causes an interrupt. Digital thermal sensor 9 is located adjacent to the linear thermal sensor.
32:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	IntTemp_8	Temperature level at which digital thermal sensor 8 causes an interrupt. Digital thermal sensor 8 is located in the PPU.

11.6.7 Thermal Sensor Global Interrupt Temperature Register (TS_GITR)

The TS_GITR register contains a second interrupt temperature level in addition to the independent interrupt temperature levels set in TS_ITR1 and TS_ITR2. This level applies to all sensors in the Cell Broadband Engine. The encoded global interrupt temperature level in this register is compared to the current temperature encoding for each sensor. The result of these comparisons is used to generate a thermal management interrupt.

The TS_GITR register provides early indication of a temperature rise in the CBE. The interrupt corresponding to this temperature point can cause an attention to a system controller. Privileged software or the system controller or both can use this information to start actions to control the temperature (such as increasing the fan speed, rebalancing the application running, and so on).

Note: See *Table 11-2* on page 264 for the mapping of the temperature encoding for the following bit range and field name:

bits [58:63] GIntTemp(8)

Register Short Name	TS_GITR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509830’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

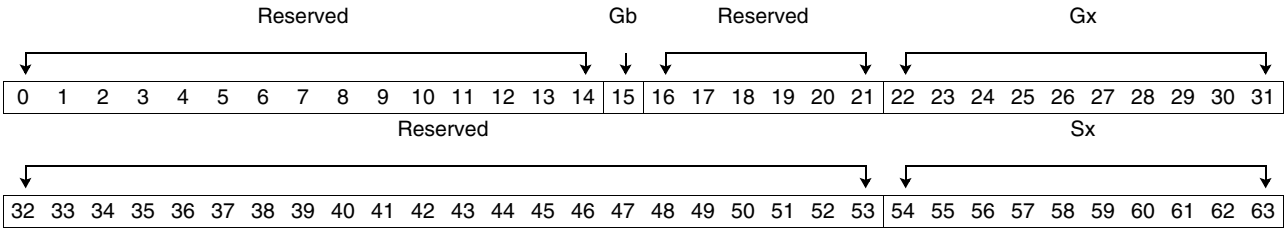
Bits	Field Name	Description
0:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	GIntTemp	Global temperature level at which any enable digital thermal sensor causes an interrupt.

11.6.8 Thermal Sensor Interrupt Status Register (TS_ISR)

The TS_ISR register identifies which sensors met the interrupt condition. This register contains two sets of status bits; the digital sensor global threshold interrupt status bits (TS_ISR[22:31]) and the digital sensor threshold interrupt status bits (TS_ISR[54:63]).

Hardware sets the status bits as described in the following tables. After the status bit is set to '1', the state is maintained until reset to '0' by privileged software. Privileged software resets a status bit to '0' by writing a '1' to the corresponding bit in the TS_ISR.

Register Short Name	TS_ISR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509838'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



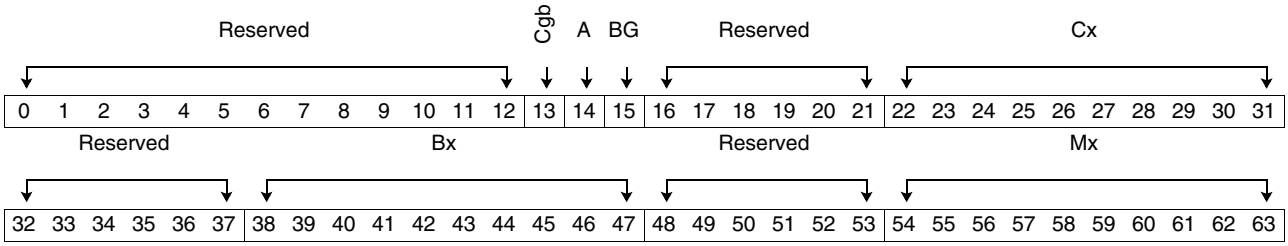
Bits	Field Name	Description
0:14	Reserved	Bits are not implemented. All bits read back zero.
15	Gb	Digital Sensor Global Below Threshold interrupt status. This bit is not set when the interrupt range is configured to above or equal to the global interrupt temperature level. 0 Interrupt not pending for Digital Thermal Sensor. 1 Interrupt pending for Digital Thermal Sensor below the global threshold interrupt level.
16:21	Reserved	Bits are not implemented. All bits read back zero.
22:31	Gx	Digital Sensor Global Threshold interrupt status (where x equals the sensor number). These bits are not set when the interrupt range is configured to below the global interrupt temperature level. The sensor numbers range left to right from 9 to 0, so that bit 22 is associated with sensor number 9 and bit 31 is associated with sensor number 0. See TS_CTSR1 and TS_CTSR2 for sensor number mapping. For each bit: x'000' Interrupt not pending for Digital Thermal Sensor x. x'001' Interrupt pending for Digital Thermal Sensor x.
32:53	Reserved	Bits are not implemented. All bits read back zero.
54:63	Sx	Digital Sensor Threshold interrupt status (where x equals sensor number). The sensor numbers range left to right from 9 to 0, so that bit 54 is associated with sensor number 9 and bit 63 is associated with sensor number 0. See TS_CTSR1 and TS_CTSR2 for sensor number mapping. For each bit: x'000' Interrupt not pending for Digital Thermal Sensor x. x'001' Interrupt pending for Digital Thermal Sensor x.

Programming Note: If the interrupt condition is still met for a status bit that is being reset, the status bit remains set. To avoid an immediate interrupt, privileged software should either mask the interrupting condition using the TS_IMR register or ensure that the interrupting condition is not met before resetting the interrupt status bit.

11.6.9 Thermal Sensor Interrupt Mask Register (TS_IMR)

The TS_IMR register contains mask bits (Mx) to prevent an interrupt status bit from generating a thermal management interrupt to the PPE. It also contains controls (Bx) to select the temperature range for the interrupt conditions. In addition, this register contains controls for (Cx) to select which sensors participate in the global interrupt condition.

Register Short Name	TS_IMR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509840’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



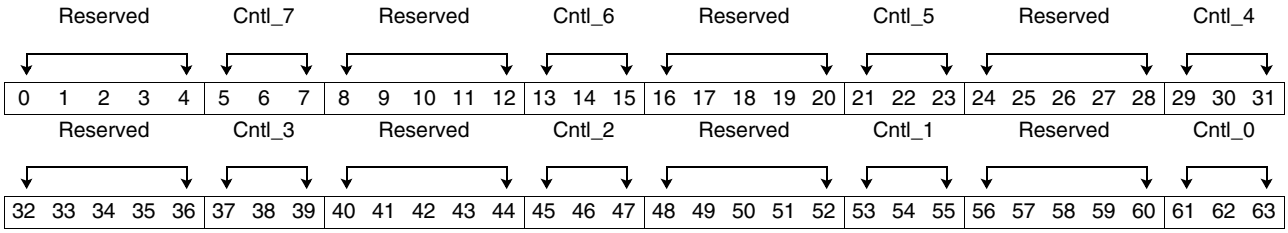
Bits	Field Name	Description
0:12	Reserved	Bits are not implemented; all bits read back zero.
13	Cgb	Mask for Digital Thermal Global Below Threshold interrupt 0 Sensor will not cause a global interrupt (disabled). 1 Sensor may cause a global interrupt (enabled). **Note:** This bit only affects the interrupt generation when the global interrupt is set for interrupting when all sensors are below the threshold.
14	A	Enable the assertion of system controller “Attention” signal 0 Attention will not be asserted (disabled). 1 Attention will be asserted when the Gb or any Gx bit is set (enabled).
15	BG	Direction for Digital Thermal Global Threshold interrupt 0 Interrupt when the temperature of any enabled sensor is above or equal to the global Interrupt temperature level. 1 Interrupt when the temperatures of all enabled sensors are below the global interrupt temperature level.
16:21	Reserved	Bits are not implemented; all bits read back zero.

Bits	Field Name	Description
22:31	Cx	Mask for Digital Thermal Global Threshold interrupt (where x equals sensor number). The sensor numbers range left to right from 9 to 0, so that bit 22 is associated with sensor number 9 and bit 31 is associated with sensor number 0. See TS_CTSR1 and TS_CTSR2 for sensor number mapping. For each bit: 0 Sensor will not cause a global interrupt (disabled). 1 Sensor may cause a global interrupt (enabled).
32:37	Reserved	Bits are not implemented; all bits read back zero.
38:47	Bx	Direction for Digital Thermal Threshold interrupt (where x equals sensor number). The sensor numbers range left to right from 9 to 0, so that bit 38 is associated with sensor number 9 and bit 47 is associated with sensor number 0. See TS_CTSR1 and TS_CTSR2 for sensor number mapping. For each bit: 0 Interrupt when temperature is above or equal to the interrupt temperature level. 1 Interrupt when temperature is below the interrupt temperature level.
48:53	Reserved	Bits are not implemented; all bits read back zero.
54:63	Mx	Mask for Digital Thermal Threshold interrupt (where x equals sensor number). The sensor numbers range left to right from 9 to 0, so that bit 54 is associated with sensor number 9 and bit 63 is associated with sensor number 0. See TS_CTSR1 and TS_CTSR2 for sensor number mapping. For each bit: 0 Interrupt is disabled. 1 Interrupt is enabled.

11.6.10 Thermal Management Control Register 1 (TM_CR1)

This register contains the controls for the sensors located in the SPEs. Each sensor has independent controls.

Register Short Name	TM_CR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509848’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:4	Reserved	Bits are not implemented; all bits read back zero.
5:7	Cntl_7	Thermal management control for digital thermal sensor 7. Digital thermal sensor 7 is located in physical SPE 7. 0*xx* Throttling disabled. The SPE’s execution will not be throttled or stopped when the temperature reaches the throttle point. 100 Normal Operation. Throttling and SPE stop safety is enabled. 101 Always throttled. The SPE’s execution is throttled regardless of temperature. SPE stop safety is enabled. 110 SPE stop safety disabled. Throttling will occur as defined. The SPE is not stopped if the encoded current temperature reaches the FullThrottleSPE point. 111 Always throttled and core stop disabled. The SPE’s execution will be throttled regardless of temperature. The SPE is not stopped if encoded current temperature reaches the FullThrottleSPE point.
8:12	Reserved	Bits are not implemented; all bits read back zero.
13:15	Cntl_6	Thermal management control for digital thermal sensor 6. Digital thermal sensor 6 is located in physical SPE 6. The bit definition for this field is the same as the Cntl_7 field.
16:20	Reserved	Bits are not implemented; all bits read back zero.
21:23	Cntl_5	Thermal management control for digital thermal sensor 5. Digital thermal sensor 5 is located in physical SPE 5. The bit definition for this field is the same as the Cntl_7 field.
24:28	Reserved	Bits are not implemented; all bits read back zero.
29:31	Cntl_4	Thermal management control for digital thermal sensor 4. Digital thermal sensor 4 is located in physical SPE 4. The bit definition for this field is the same as the Cntl_7 field.
32:36	Reserved	Bits are not implemented; all bits read back zero.
37:39	Cntl_3	Thermal management control for digital thermal sensor 3. Digital thermal sensor 3 is located in physical SPE 3. The bit definition for this field is the same as the Cntl_7 field.
40:44	Reserved	Bits are not implemented; all bits read back zero.

Bits	Field Name	Description
45:47	Cntl_2	Thermal management control for digital thermal sensor 2. Digital thermal sensor 2 is located in physical SPE 2. The bit definition for this field is the same as the Cntl_7 field.
48:52	Reserved	Bits are not implemented; all bits read back zero.
53:55	Cntl_1	Thermal management control for digital thermal sensor 1. Digital thermal sensor 1 is located in physical SPE 1. The bit definition for this field is the same as the Cntl_7 field.
56:60	Reserved	Bits are not implemented; all bits read back zero.
61:63	Cntl_0	Thermal management control for digital thermal sensor 0. Digital thermal sensor 0 is located in physical SPE 0. The bit definition for this field is the same as the Cntl_7 field.

11.6.11 Thermal Management Control Register 2 (TM_CR2)

This register contains the controls for the sensors located in the PPE and the sensor located adjacent to the linear thermal sensor.

Register Short Name	TM_CR2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509850’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

Bits	Field Name	Description
0:60	Reserved	Bits are not implemented; all bits read back zero.
61:63	Cntl_8	Thermal management control for digital thermal sensor 8. Digital thermal sensor 8 is located in the PPE. 0*xx* Throttling disabled. The core’s execution is not throttled or stopped when the temperature reaches the throttle point. 100 Normal Operation. Throttling and core stop safety is enabled. 101 Always throttled. The core’s execution is throttled regardless of temperature. Core stop safety is enabled. 110 Core stop safety disabled. Throttling occurs as defined. The core will not be stopped if the encoded current temperature reaches the FullThrottlePPC point. 111 Always throttled and core stop disabled. The core’s execution is throttled regardless of temperature. The core is not stopped if the encoded current temperature reaches the FullThrottlePPC point.

11.6.12 Thermal Management System Interrupt Mask Register (TM_SIMR)

This register controls which PPE interrupts cause the thermal management logic to exit a throttling state on the PPE. Throttling is exited for both PPE threads, regardless of the thread targeted by the interrupt. Throttling of the SPEs is never exited based on a system interrupt condition. The PPE interrupt conditions that can override a throttling condition are listed below:

- External
- Decrementer
- Hypervisor Decrementer
- System Error
- Thermal Management

Register Short Name	TM_SIMR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509858'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

Bits	Field Name	Description
0:58	Reserved	Bits are not implemented; all bits read back zero.
59	T	Mask for Thermal Management Interrupt 0 Interrupt does not exit PPE throttling state (disabled). 1 Interrupt exits the PPE throttling state (enabled).
60	S	Mask for System Error Interrupt 0 Interrupt does not exit PPE throttling state (disabled). 1 Interrupt exits the PPE throttling state (enabled).
61	H	Mask for Hypervisor Decrementer Interrupt 0 Interrupt does not exit PPE throttling state (disabled). 1 Interrupt exits the PPE throttling state (enabled).
62	D	Mask for Decrementer Interrupt 0 Interrupt does not exit PPE throttling state (disabled). 1 Interrupt exits the PPE throttling state (enabled).
63	E	Mask for External Interrupt 0 Interrupt does not exit PPE throttling state (disabled). 1 Interrupt exits the PPE throttling state (enabled).

11.6.13 Thermal Management Throttle Point Register (TM_TPR)

This register contains the encoded temperature points at which throttling of a core's execution begins and ends. This register also contains an encoded temperature point at which a core's execution is fully throttled (in other words, stopped).

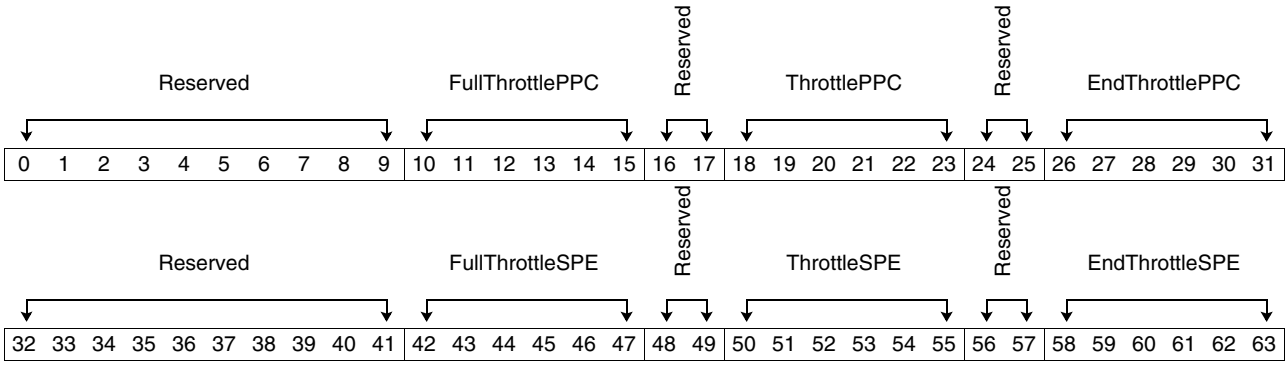
The values in this register are used to set three temperature points for transition between the three thermal management states; 1) Normal Run (N), 2) Core Throttled (T), and 3) Core Stopped (S). Independent temperature points are supported for the PPE and the SPEs.

When the encoded current temperature of a sensor in the TS_CTSR is equal to or greater than the throttle temperature ([18:23] or [50:55]), throttling execution of the corresponding core begins, if enabled. Execution throttling continues until the encoded current temperature of the corresponding sensor is less than the encoded temperature to end throttling ([26:31] or [58:63]). As a safety measure, if the encoded current temperature is equal to or greater than the full throttle point ([10:15] or [42:47]), the corresponding core is stopped.

Note: See *Table 11-2* on page 264 for mappings of the temperature encodings for the following bit ranges and field names:

bits [10:15] FullThrottlePPC
bits [18:23] ThrottlePPC
bits [26:31] EndThrottlePPC
bits [42:47] FullThrottleSPE
bits [50:55] ThrottleSPE
bits [58:63] EndThrottleSPE

Register Short Name	TM_TPR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509860'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



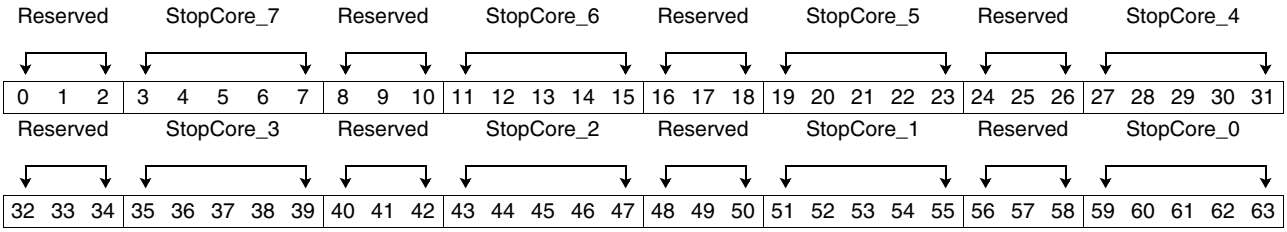
Bits	Field Name	Description
0:9	Reserved	Bits are not implemented; all bits read back zero.
10:15	FullThrottlePPC	Full throttling temperature point. The PPC is stopped when TS_CTSR2[58:63] is greater than or equal to TM_TPR[10:15]. Typically the value of this field should be greater than the value of the ThrottlePPC field. Setting this field to a value less than or equal to the ThrottlePPC field prevents dynamic throttling. Setting the value of this field to x‘1F’ effectively prevents the PPC from ever being stopped. All values for this field are valid.
16:17	Reserved	Bits are not implemented; all bits read back zero.
18:23	ThrottlePPC	Throttle temperature point. Dynamic throttling of the PPC begins when TS_CTSR2[58:63] is greater than or equal to TM_TPR[26:31]. Typically the value of this field should be a value corresponding to a temperature less than the maximum junction temperature supported by an implementation. (Refer to the product specific datasheet for the maximum junction temperature.) Setting the value of this field to x‘1F’ effectively prevents the PPC from ever being dynamically throttled. Setting the value of this field to x‘00’ effectively throttles the PPC regardless of the temperature. All values for this field are valid.
24:25	Reserved	Bits are not implemented; all bits read back zero.
26:31	EndThrottlePPC	End of throttle temperature point. Dynamic throttling of the PPC ceases when TS_CTSR2[58:63] is less than TM_TPR[26:31]. Typically, the value of this field should be less than the value of the ThrottlePPC field. Setting this field to a value greater than the ThrottlePPC field might prevent dynamic throttling. A value equal to the ThrottlePPC field provides very little hysteresis between the starting and stopping throttle point. All values for this field are valid.
32:41	Reserved	Bits are not implemented; all bits read back zero.
42:47	FullThrottleSPE	Full throttling temperature point. The SPE will be stopped when TS_CTSR2[Cur(x)] is greater than or equal to TM_TPR[42:46] (where $0 \le x \le 7$). Typically the value of this field should be greater than the value of the ThrottleSPE field. Setting this field to a value less than or equal to the ThrottleSPE field prevents dynamic throttling. Setting the value of this field to x‘1F’ effectively prevents the SPE from ever being stopped. All values for this field are valid.
48:49	Reserved	Bits are not implemented; all bits read back zero.
50:55	ThrottleSPE	Throttle temperature point. Dynamic throttling of the SPE begins when TS_CTSR2[Cur(x)] $\ge$ TM_TPR[58:63] (where $0 \le x \le 7$). Typically the value of this field should be a value corresponding to a temperature less than the maximum junction temperature supported by an implementation. Refer to the product specific datasheet for the maximum junction temperature. Setting the value of this field to x‘1F’ effectively prevents the SPE from ever being dynamically throttled. Setting the value of this field to x‘00’ effectively throttles the SPE regardless of the temperature. All values for this field are valid.
56:57	Reserved	Bits are not implemented; all bits read back zero.
58:63	EndThrottleSPE	End of throttle temperature point. Dynamic throttling of the SPE ceases when TS_CTSR2[Cur(x)] is less than TM_TPR[58:63] (where $0 \le x \le 7$). Typically the value of this field should be less than the value of the ThrottleSPE field. Setting this field to a value greater than the ThrottleSPE field might prevent dynamic throttling. A value equal to the ThrottleSPE field provides very little hysteresis between the starting and stopping throttle point. All values for this field are valid.

11.6.14 Thermal Management Stop Time Register 1 (TM_STR1)

The thermal management stop time registers control the amount of throttling applied to a specific core in the thermal management throttled state. The values in this register are expressed in a percentage of time a core is stopped versus the time a core is run (CoreStop(x)/32). The actual number of clocks that a core is stopped and run is controlled by the Thermal Management Scale Register.

The Thermal Management Stop Time Register 1 contains the throttling stop times for the sensors located in the SPEs. Each sensor has independent stop time.

Register Short Name	TM_STR1	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x'509868'	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:2	Reserved	Bits are not implemented; all bits read back zero.
3:7	StopCore_7	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
8:10	Reserved	Bits are not implemented; all bits read back zero.
11:15	StopCore_6	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
16:18	Reserved	Bits are not implemented; all bits read back zero.
19:23	StopCore_5	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
24:26	Reserved	Bits are not implemented; all bits read back zero.
27:31	StopCore_4	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
32:34	Reserved	Bits are not implemented; all bits read back zero.
35:39	StopCore_3	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
40:42	Reserved	Bits are not implemented; all bits read back zero.
43:47	StopCore_2	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
48:50	Reserved	Bits are not implemented; all bits read back zero.

Bits	Field Name	Description
51:55	StopCore_1	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.
56:58	Reserved	Bits are not implemented; all bits read back zero.
59:63	StopCore_0	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 results in the dynamic throttling of the core to be disabled. A setting of 0 does not disable the core stop safety.

11.6.15 Thermal Management Stop Time Register 2 (TM_STR2)

The thermal management stop time registers control the amount of throttling applied to a specific core in the thermal management throttled state. The values in this register are expressed as a percentage of time that a core is stopped versus the time that a core is run (CoreStop(x)/32). The actual number of clocks that a core is stopped and run is controlled by the Thermal Management Scale Register.

The Thermal Management Control Register 2 contains the throttling stop times for the sensor located in the PPE.

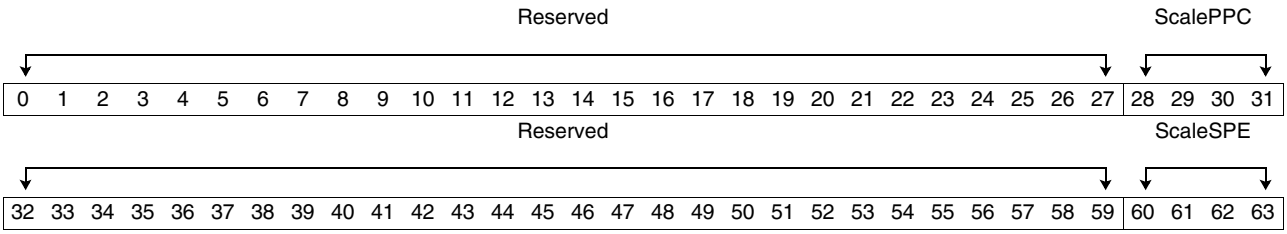
Register Short Name	TM_STR2	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509870’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV

Bits	Field Name	Description
0:58	Reserved	Bits are not implemented; all bits read back zero.
59:63	StopCore_8	Throttling Stop Time when throttle temperature high reached. Setting this register to 0 disables the dynamic throttling of the core. A setting of 0 does not disable the core stop safety.

11.6.16 Thermal Management Throttle Scale Register (TM_TSR)

The Thermal Management Throttle Scale Register controls the actual number of cycles that a core is stopped and run during the thermal management throttle state. The Thermal Management Throttle Scale Register contains the scale factors for the sensors located in the SPEs and PPC.

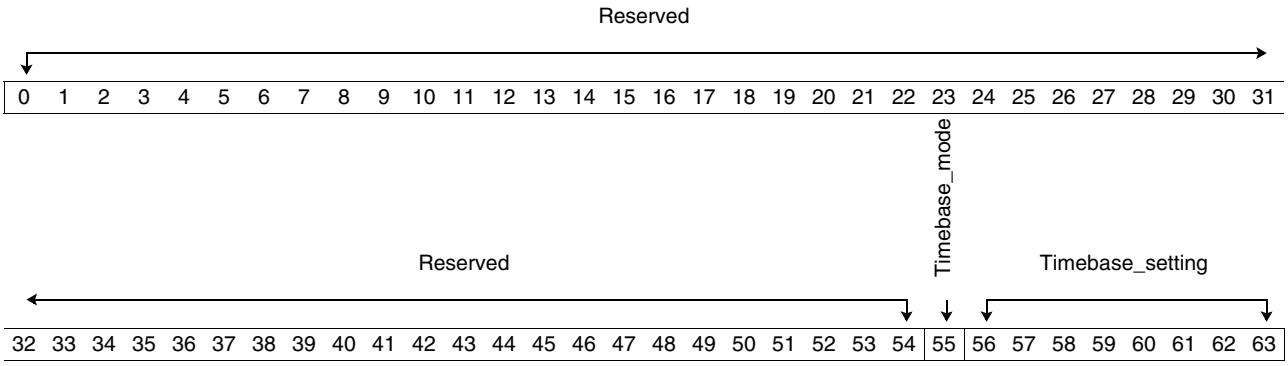
Register Short Name	TM_TSR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509878’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:27	Reserved	Bits are not implemented; all bits read back zero.
28:31	ScalePPC	ScalePPC 0 Disables ScalePPC 1 Enables throttling of scale time when the throttle temperature high is reached Setting both StopCore and Scale to 0 also disables the throttling feature.
32:59	Reserved	Bits are not implemented; all bits read back zero.
60:63	ScaleSPE	ScaleSPE 0 Disables ScaleSPE 1 Enables throttling of scale time when the throttle temperature high is reached Setting both StopCore and Scale to 0 also disables the throttling feature.

11.6.17 Time Base Register (TBR)

Register Short Name	TBR	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘509890’	**Memory Map Area**	Pervasive: Thermal and Power Management
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	PRV



Bits	Field Name	Description
0:54	Reserved	Bits are not implemented; all bits read back zero.
55	Timebase_mode	Time Base Mode 0 Internal time base sync mode 1 External time base sync mode. When this bit equals 1, the value of bits [55:63] has no effect.
56:63	Timebase_setting	Time Base Setting Internal reference clock divider setting that is based on an LFSR. A setting of x‘00’ (the POR default value) indicates the LFSR is not counting.

12. PowerPC Processor Element Special Purpose Registers

This section describes special purpose registers (SPRs) used by the PowerPC Processor Element (PPE). These registers are read from using the **mfspr** PowerPC instruction or written to using the **mtspr** PowerPC instruction. *Table 12-1* on page 288 shows the PPE SPR memory map and lists the PPE SPRs.

Table 12-1 uses the following conventions:

- Architected SPRs that do not contain implementation-specific information are included in *Table 12-1* but are not described in this manual. *Table 12-1* provides cross references to additional information for each architected SPR.
- Architected SPRs that contain implementation-specific information are included in *Table 12-1* and are described in this section. When applicable, each register description provides specific cross references to additional information. The SPRs in the *Decimal* column are highlighted to indicate that an SPR is implementation specific.

  Programming Note: Programmers should be aware that code using implementation-specific information is not guaranteed to be portable across different implementations of the architecture, although a significant effort is made to minimize incompatibilities between different designs.
- The notes provided at the end of *Table 12-1* describe unique and implementation-specific information for certain SPRs.

Table 12-1. PPE Special Purpose Registers (Page 1 of 5)

SPR		Duplicated for Multithreading[3]	Register Name (Short Name) • Cross Reference to Additional Information	Unit[4]	Read/Write	Synchronization Requirements[5]				Hypervisor/ Privileged[6]		Size (Bits)	Power-On Reset (POR) Value (All bits set to '0' unless otherwise noted)
						For Data		For Instructions					
Decimal[1]	spr[5:9] spr[0:4][2]					Before Writes	After Writes	Before Writes	After Writes	Read (mf)	Write (mt)		
01	00000 00001	Yes	*Fixed-Point Exception Register (XER)* • *PowerPC User Instruction Set Architecture, Book I*	XU	R/W	N/A				—	—	64	
08	00000 01000	Yes	*Link Register (LR)* • *PowerPC User Instruction Set Architecture, Book I*	IU	R/W	N/A				—	—	64	
09	00000 01001	Yes	*Count Register (CTR)* • *PowerPC User Instruction Set Architecture, Book I*	IU	R/W	N/A				—	—	64	
18	00000 10010	Yes	*Data Storage Interrupt Status Register (DSISR)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				Priv		32	
19	00000 10011	Yes	*Data Address Register (DAR)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				Priv		64	
22	00000 10110	Yes	*Decrementer Register (DEC)* • *PowerPC Operating Environment Architecture, Book III*	MMU	R/W	None				Priv		32	x'7FFF_FFFF'
25	00000 11001	No	*Storage Description Register 1 (SDR1)* • *PowerPC Operating Environment Architecture, Book III*	MMU	R/W	PTE-sync	CSI	PTE-sync	CSI	HV		64	
26	00000 11010	Yes	*Machine Status Save/Restore Register 0 (SRR0)* • *PowerPC Operating Environment Architecture, Book III*	IU	R/W	N/A				Priv		64	
27	00000 11011	Yes	*Machine Status Save/Restore Register 1 (SRR1)* • *PowerPC Operating Environment Architecture, Book III*	IU	R/W	N/A				Priv		64	
29	00000 11101	Yes	*Address Compare Control Register (ACCR)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	CSI		None		Priv		64	
136	00100 01000	No	*Control Register (CTRL)* • *Section 12.1.1 Control Register (CTRL)* on page 293	IU	R	N/A				—	N/A	32	
152	00100 11000				W	None				N/A	Priv[7]		N/A
256	01000 00000	Yes	• *VXU Register Save (VRSAVE)* • See the *PowerPC Microprocessor Family: Vector/SIMD Multimedia Extension Technology Programming Environments Manual*	XU	R/W	N/A				—	—	32	
259	01000 00011	Yes	*Software Use Special Purpose Register 3 (SPRG3) - Read Only* • *PowerPC Operating Environment Architecture, Book III*	XU	R	N/A				—	N/A	64	
268	01000 01100	No	*Time Base Register (TB)* • *PowerPC Operating Environment Architecture, Book III* The physical implementation of this register includes the following registers: • Time Base Register - Read Only (TB), SPR 268 • Time Base Upper Register - Read Only (TBU), SPR 269 • Time Base Lower Register - Write Only (TBL), SPR 284 • Time Base Upper Register - Write Only (TBU), SPR 285	MMU	R	N/A				—	N/A	64	
269	01000 01101											32	

Table 12-1. PPE Special Purpose Registers (Page 2 of 5)

SPR		Duplicated for Multithreading[3]	Register Name (Short Name) • Cross Reference to Additional Information	Unit[4]	Read/Write	Synchronization Requirements[5]				Hypervisor/ Privileged[6]		Size (Bits)	Power-On Reset (POR) Value (All bits set to '0' unless otherwise noted)
						For Data		For Instructions					
Decimal[1]	spr[5:9] spr[0:4][2]					Before Writes	After Writes	Before Writes	After Writes	Read (mf)	Write (mt)		
272	01000 10000	Yes	*Software Use Special Purpose Register 0 (SPRG0)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				Priv		64	
273	01000 10001	Yes	*Software Use Special Purpose Register 1 (SPRG1)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				Priv		64	
274	01000 10010	Yes	*Software Use Special Purpose Register 2 (SPRG2)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				Priv		64	
275	01000 10011	Yes	*Software Use Special Purpose Register 3 (SPRG3)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				Priv		64	
284	01000 11100	No	*Time Base Register (TB)* • *PowerPC Operating Environment Architecture, Book III* The physical implementation of this register includes the following registers: • Time Base Register - Read Only (TB), SPR 268 • Time Base Upper Register - Read Only (TBU), SPR 269 • Time Base Lower Register - Write Only (TBL), SPR 284 • Time Base Upper Register - Write Only (TBU), SPR 285	MMU	W	None				N/A	HV	32	N/A
285	01000 11101											32	
287	01000 11111	No	*PPE Processor Version Register (PVR)* • *PowerPC Operating Environment Architecture, Book III* **Note:** The PVR is updated if the latch-to-latch path of the PPU is not logically and functionally equivalent to the previous version (that is, if the unit is not Verity clean with the previous version).	XU	R	N/A				Priv	N/A	32	x'0070_1000' (65 nm DD 1.1) x'0070_0501' (10KE DD 3.2) x'0070_0501' (10KE DD 3.1) x'0070_0500' (10KE DD 3.0) x'0070_0400' (10KE DD 2.0) x'0070_0100' (10KE DD 1.0)
304	01001 10000	Yes	*Hypervisor Software Use Special Purpose Register 0 (HSPRG0)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				HV		64	
305	01001 10001	Yes	*Hypervisor Software Use Special Purpose Register 1 (HSPRG1)*[8] • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	N/A				HV		64	
310	01001 10110	No	*Hypervisor Decrementer Register (HDEC)*[8] • *PowerPC Operating Environment Architecture, Book III*	MMU	R/W	None				HV		32	x'7FFF_FFFF'
312	01001 11000	No	*Real Mode Offset Register (RMOR)* • *Section 12.1.2 Real Mode Offset Register (RMOR)* on page 295	MMU	R/W	CSI		None	CSI	HV		64	
313	01001 11001	No	*Hypervisor Real Mode Offset Register (HRMOR)*[8] • *Section 12.1.3 Hypervisor Real Mode Offset Register (HRMOR)* on page 296	MMU	R/W	CSI		None	CSI	HV		64	
314	01001 11010	Yes	*Hypervisor Machine Status Save/Restore Register 0 (HSRR0)*[8] • *PowerPC Operating Environment Architecture, Book III*	IU	R/W	N/A				HV		64	

Table 12-1. PPE Special Purpose Registers (Page 3 of 5)

SPR		Duplicated for Multithreading[3]	Register Name (Short Name) • Cross Reference to Additional Information	Unit[4]	Read/Write	Synchronization Requirements[5]				Hypervisor/ Privileged[6]		Size (Bits)	Power-On Reset (POR) Value (All bits set to '0' unless otherwise noted)
						For Data		For Instructions					
Decimal[1]	spr[5:9] spr[0:4][2]					Before Writes	After Writes	Before Writes	After Writes	Read (mf)	Write (mt)		
315	01001 11011	Yes	*Hypervisor Machine Status Save/Restore Register 1 (HSRR1)*[8] • *PowerPC Operating Environment Architecture, Book III*	IU	R/W	N/A				HV		64	
318	01001 11110	Par-tial	*Logical Partition Control Register (LPCR)*[8] • *Section 12.1.4 Logical Partition Control Register (LPCR)* on page 297	MMU	R/W	CSI		None	CSI	HV		64	
319	01001 11111	No	*Logical Partition Identity Register (LPIDR)*[8] • *Section 12.1.5 Logical Partition Identity Register (LPIDR)* on page 298	MMU	R/W	CSI		CSI		HV		32	
896	11100 00000	Yes	*Thread Status Register Local (TSRL)* • *Section 12.1.6 Thread Status Register Local (TSRL)* on page 299	IU	R/W	None			CSI	—	—	64	
897	11100 00001	Yes	*Thread Status Register Remote (TSRR)* • *Section 12.1.7 Thread Status Register Remote (TSRR)* on page 301	IU	R	N/A				—	N/A	64	
921	11100 11001	No	*Thread Switch Control Register (TSCR)* • *Section 12.1.8 Thread Switch Control Register (TSCR)* on page 302	IU	R/W	None			CSI	HV		64	
922	11100 11010	No	*Thread Switch Timeout Register (TTR)* • *Section 12.1.9 Thread Switch Timeout Register (TTR)* on page 304	IU	R/W	None			CSI	HV		64	
946	11101 10010	Yes	*PPE Translation Lookaside Buffer Index Hint Register (PPE_TLB_Index_Hint)* • *Section 12.1.10.1 PPE Translation Lookaside Buffer Index Hint Register (PPE_TLB_Index_Hint)* on page 305	MMU	R	N/A				Priv	N/A	64	
947	11101 10011	No	*PPE Translation Lookaside Buffer Index Register (PPE_TLB_Index)* • *Section 12.1.11 PPE Translation Lookaside Buffer Index Register (PPE_TLB_Index)* on page 307	MMU	R/W[9]	None				HV		64	
948	11101 10100	No	*PPE Translation Lookaside Buffer Virtual-Page Number Register (PPE_TLB_VPN)* • *Section 12.1.11.1 PPE Translation Lookaside Buffer Virtual-Page Number Register (PPE_TLB_VPN)* on page 309	MMU	R/W	CSI	CSI	None	CSI	HV		64	
949	11101 10101	No	*PPE Translation Lookaside Buffer Real-Page Number Register (PPE_TLB_RPN)* • *Section 12.1.11.2 PPE Translation Lookaside Buffer Real-Page Number Register (PPE_TLB_RPN)* on page 310	MMU	R/W	None				HV		64	
951	11101 10111	No	*PPE Translation Lookaside Buffer RMT Register (PPE_TLB_RMT)* • *Section 12.1.12 PPE Translation Lookaside Buffer RMT Register (PPE_TLB_RMT)* on page 312	MMU	R/W	CSI	CSI	None	CSI	HV		64	
952	11101 11000	Yes	*Data Range Start Register 0 (DRSR0)* • Range Start Register (RSR) in the *Cell Broadband Engine Architecture*	XU	R/W	Sync	Sync, CSI[10]	None	None	HV		64	
953	11101 11001	Yes	*Data Range Mask Register 0 (DRMR0)* • Range Mask Register (RMR) in the *Cell Broadband Engine Architecture*	XU	R/W	Sync	Sync, CSI[10]	None	None	HV		64	

Table 12-1. PPE Special Purpose Registers (Page 4 of 5)

SPR		Duplicated for Multithreading[3]	Register Name (Short Name) • Cross Reference to Additional Information	Unit[4]	Read/Write	Synchronization Requirements[5]				Hypervisor/ Privileged[6]		Size (Bits)	Power-On Reset (POR) Value (All bits set to '0' unless otherwise noted)
						For Data		For Instructions					
Decimal[1]	spr[5:9] spr[0:4][2]					Before Writes	After Writes	Before Writes	After Writes	Read (mf)	Write (mt)		
954	11101 11010	Yes	*Data Class ID Register 0 (DCIDR0)* • *Cell Broadband Engine Architecture* • *Section 12.1.13.1 Data Class ID Register 0 (DCIDR0)* on page 313	XU	R/W	Sync	Sync, CSI[10]	None	None	HV		32	
955	11101 11011	Yes	*Data Range Start Register 1 (DRSR1)* • *Range Start Register (RSR)* in the *Cell Broadband Engine Architecture*	XU	R/W	Sync	Sync, CSI[10]	None	None	HV		64	
956	11101 11100	Yes	*Data Range Mask Register 1 (DRMR1)* • Range Mask Register (RMR) in the *Cell Broadband Engine Architecture*	XU	R/W	Sync	Sync, CSI[10]	None	None	HV		64	
957	11101 11101	Yes	*Data Class ID Register 1 (DCIDR1)* • *Section 12.1.13.2 Data Class ID Register 1 (DCIDR1)* on page 314	XU	R/W	Sync	Sync, CSI[10]	None	None	HV		32	
976	11110 10000	Yes	*Instruction Range Start Register 0 (IRSR0)* • Range Start Register (RSR) in the *Cell Broadband Engine Architecture*	IU	R/W	None	None	Sync	Sync, CSI[10]	HV		64	
977	11110 10001	Yes	*Instruction Range Mask Register 0 (IRMR0)* Range Mask Register (RMR) in the *Cell Broadband Engine Architecture*	IU	R/W	None	None	Sync	Sync, CSI[10]	HV		64	
978	11110 10010	Yes	*Instruction Class ID Register 0 (ICIDR0)* • *Section 12.1.14.1 Instruction Class ID Register 0 (ICIDR0)* on page 315	IU	R/W	None	None	Sync	Sync, CSI[10]	HV		32	
979	11110 10011	Yes	*Instruction Range Start Register 1 (IRSR1)* • Range Start Register (RSR) in the *Cell Broadband Engine Architecture*	IU	R/W	None	None	Sync	Sync, CSI[10]	HV		64	
980	11110 10100	Yes	*Instruction Range Mask Register 1 (IRMR1)* • Range Mask Register (RMR) in the *Cell Broadband Engine Architecture*	IU	R/W	None	None	Sync	Sync, CSI[10]	HV		64	
981	11110 10101	Yes	*Instruction Class ID Register 1 (ICIDR1)* • *Section 12.1.14.2 Instruction Class ID Register 1 (ICIDR1)* on page 316	IU	R/W	None	None	Sync	Sync, CSI[10]	HV		32	
1008	11111 10000	No	*Hardware Implementation Register 0 (HID0)* • *Section 12.1.15 Hardware Implementation Register 0 (HID0)* on page 317	IU	R/W	Sync	Sync, CSI[10]	Sync	Sync, CSI[10]	HV		64	
1009	11111 10001	No	*Hardware Implementation Register 1 (HID1)* • *Section 12.1.16 Hardware Implementation Register 1 (HID1)* on page 320	IU	R/W	Sync	Sync, CSI[10]	Sync	Sync, CSI[10]	HV		64	
1012	11111 10100	No	*Hardware Implementation Register 4 (HID4)* • *Section 12.1.17 Hardware Implementation Register 4 (HID4)* on page 324	XU	R/W	Sync	Sync, CSI[10]	Sync	Sync, CSI[10]	HV		64	
1013	11111 10101	Yes	*Data Address Breakpoint Register (DABR)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	Sync	CSI	None		HV		64	
1015	11111 10111	Yes	*Data Address Breakpoint Register Extension (DABRX)* • *PowerPC Operating Environment Architecture, Book III*	XU	R/W	Sync	CSI	None		HV		64	
1017	11111 11001	No	*Hardware Implementation Register 6 (HID6)* • *Section 12.1.18 Hardware Implementation Register 6 (HID6)* on page 327	MMU	R/W	Sync	Sync, CSI[10]	Sync	Sync, CSI[10]	HV		64	

Table 12-1. PPE Special Purpose Registers (Page 5 of 5)

SPR		Duplicated for Multithreading[3]	Register Name (Short Name) • Cross Reference to Additional Information	Unit[4]	Read/Write	Synchronization Requirements[5]				Hypervisor/ Privileged[6]		Size (Bits)	Power-On Reset (POR) Value (All bits set to '0' unless otherwise noted)
						For Data		For Instructions					
Decimal[1]	spr[5:9] spr[0:4][2]					Before Writes	After Writes	Before Writes	After Writes	Read (mf)	Write (mt)		
N/A	N/A	Yes	*Machine State Register (MSR)* • *Section 12.2.1 Machine State Register (MSR)* on page 331	IU	R/W	N/A				Priv	—		
1022	11111 11110	No	*CBEA-Compliant Processor Version Register (BP_VR)* • *Section 12.1.19 CBEA-Compliant Processor Version Register (BP_VR)* on page 329 • *Cell Broadband Engine Architecture*	XU	R	N/A				Priv	N/A	64	
1023	11111 11111	Yes	*Processor Identification Register (PIR)*[8] • *Section 12.1.20 Processor Identification Register (PIR)* on page 330	XU	R	N/A				Priv	N/A	32	See *Section 12.1.20*

1. Implementation-specific SPRs are highlighted.
2. The order of the two 5-bit halves of the SPR number is reversed (to follow the convention of the architecture documents).
3. Any register that is not duplicated per thread requires special care by the hypervisor when written. The hypervisor must not cause an implicit branch or undefined behavior for the thread that is not writing the register.
4. The unit notation is execution unit (XU), instruction unit (IU), memory management unit (MMU).
5. For more information, see the synchronization requirements for context alterations section of the *PowerPC Operating Environment Architecture, Book III.*
   - "N/A" indicates that the operation does not apply to this register.
   - "None" indicates that no synchronization is required.
   - "Sync" refers to the lightweight **sync L = 1** instruction unless otherwise indicated.
   - "CSI" stands for context-synchronizing instruction.
6. Explanation of the *Hypervisor/Privileged* column:
   - HV: Indicates that the register is a hypervisor resource. The hypervisor state must be enabled (MSR[HV] = '1') to write this register. Attempts to modify the contents of a hypervisor resource (such as using the move-to SPR instruction) in privileged but nonhypervisor state (MSR[HV, PR] = '00') cause a privileged-instruction program interrupt.
   - Priv: Indicates that the register is privileged. The privileged state must be enabled (MSR[PR] = '0') to read or write to the register. Attempts to access the contents of a privileged resource (such as using the move-to SPR or move-from SPR instruction) in nonprivileged state (MSR[PR] = '1') cause a privileged-instruction program interrupt.
   - —: Indicates that the register is neither privileged nor a hypervisor resource.
   - N/A: Indicates that the register is either read-only or write-only.
     – Writes using move-to instructions to unimplemented SPRs are treated as **nop** instructions. Architected registers are not changed.
     – Writes using move-to instructions to read-only SPRs are treated like unimplemented SPRs.
     – Reads using move-from instructions from unimplemented SPRs cause zeros to be written back to the general purpose register (GPR).
     – Reads using move-from instructions from write-only SPRs are treated like unimplemented SPRs.
7. The Thread Enable Bits field of CTRL can be modified only in hypervisor mode. Attempts to modify the Thread Enable Bits field of the CTRL Register (using the move-to SPR instruction) while not in hypervisor mode (MSR[HV] = '0') are ignored.
8. These registers are for logical partitioning (LPAR) support.
9. Reading of these registers is allowed for diagnostic purposes.
10. Indicates a **sync** instruction followed by any context-synchronizing instruction.

12.1 SPR Definitions

This section describes the implementation-specific special purpose registers (SPRs) used in this implementation. For a complete listing of SPRs, see *Table 12-1 PPE Special Purpose Registers* on page 288.

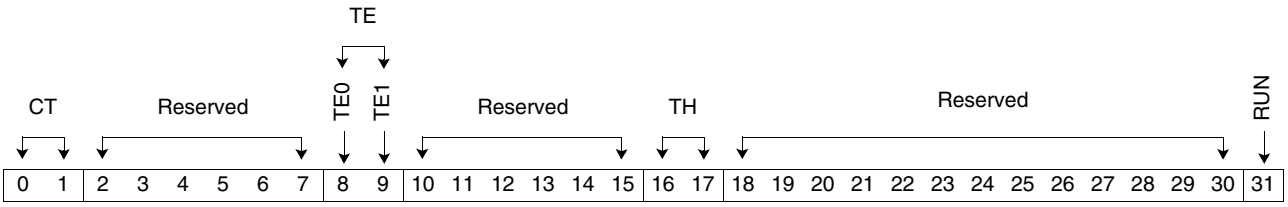
Notes:

The following information applies to the tables used at the beginning of each register description in this section.

1. In this section, power-on reset (POR) is defined as the sequence that starts when power is first applied to the chip and ends when the load function is complete.
2. *Value at Initial POR* is the value that was initialized during the scan initialization or configuration ring part of the POR sequence.
3. Some fields within the architected registers are not physically implemented. Writing to these fields has no effect, and reading from these fields returns '0'. These fields are marked Reserved. The Rsvd_I bits are reserved and implemented. Writes to Rsvd_I bits are preserved on a read.

12.1.1 Control Register (CTRL)

Register Short Name	CTRL	**Privilege Type**	Read: Not privileged Write: Privileged
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	136 (Read) 152 (Write)	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan Initialization
Specification Type	PowerPC architected register	**Unit**	IU



Bits	Field Name	Description
0:1	CT	Current thread active (Read Only) These read-only bits contain the current thread bits for threads 0 and 1. Software can read these bits to determine on which thread they are operating. Only one current thread bit is set at a time. 00 Reserved 01 Thread 1 is reading CTRL. 10 Thread 0 is reading CTRL. 11 Reserved
2:7	Reserved	Bits are not implemented; all bits read back zero.

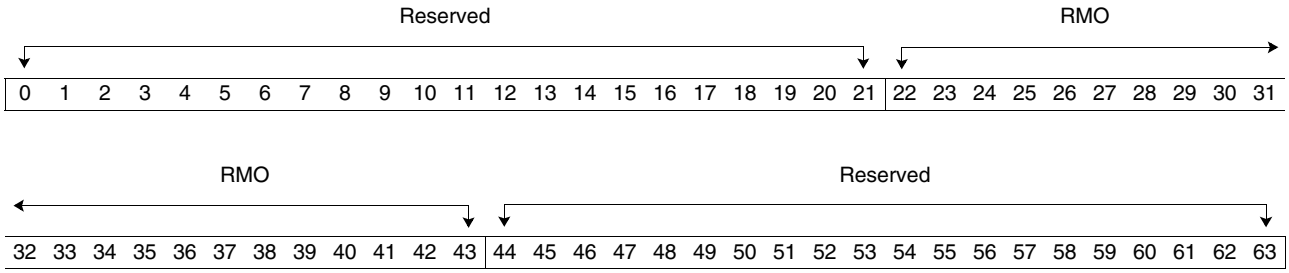
Bits	Field Name	Description
8:9	TE	Thread enable bits (Read/Write) The hypervisor state can suspend its own thread by setting the TE bit for its thread to '0'. The hypervisor state can resume the opposite thread by setting the TE bit for the opposite thread to '1'. The hypervisor state cannot suspend the opposite thread by setting the TE bit for the opposite thread to '0'. This setting is ignored and does not cause an error. TE0 is the Thread Enable bit for thread 0. TE1 is the Thread Enable bit for thread 1. If thread 0 executes the **mtctrl** instruction, these are the bit values: [TE0, TE1] Description 00 Disable or suspend thread 0; thread 1 unchanged. 01 Disable or suspend thread 0; enable or resume thread 1 if it was disabled. 10 Unchanged. 11 Enable or resume thread 1 if it was disabled. If thread 1 executes the **mtctrl** instruction, these are the bit values: [TE0, TE1] Description 00 Thread 0 unchanged; disable or suspend thread 1. 01 Unchanged. 10 Enable or resume thread 0 if it was disabled; disable or suspend thread 1. 11 Enable or resume thread 0 if it was disabled. **Note:** Software should not disable a thread when in trace mode (MSR[SE] or MSR[BE] set to '1'). Doing so causes SRR0 to be undefined and can cause a system livelock hang condition.
10:15	Reserved	Bits are not implemented; all bits read back zero.
16:17	TH	Thread history If thread A writes CTRL[RUN], then CTRL[16] is set; otherwise, if thread B writes CTRL[31], then CTRL[17] is set. These bits cannot be set directly by writing bits [16] or [17] with an **mtctrl** instruction. They are only set when a thread writes CTRL[RUN].
18:30	Reserved	Bits are not implemented; all bits read back zero.
31	RUN	Run state bit.

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III*.

12.1.2 Real Mode Offset Register (RMOR)

Register Short Name	RMOR	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	312	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	PowerPC architected register	**Unit**	MMU



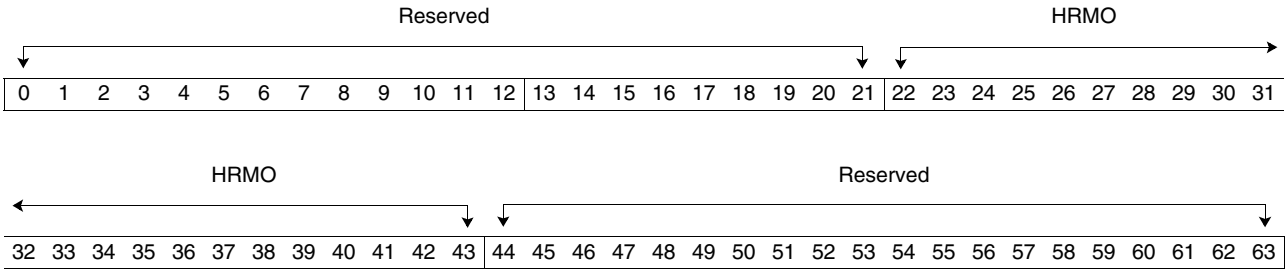
Bits	Field Name	Description
0:21	Reserved	Bits are not implemented; all bits read back zero.
22:43	RMO	Real-mode offset The offset from x‘0’ at which real-mode memory begins.
44:63	Reserved	Bits are not implemented; all bits read back zero.

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III*.

12.1.3 Hypervisor Real Mode Offset Register (HRMOR)

Register Short Name	HRMOR	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	313	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	PowerPC architected register	**Unit**	MMU



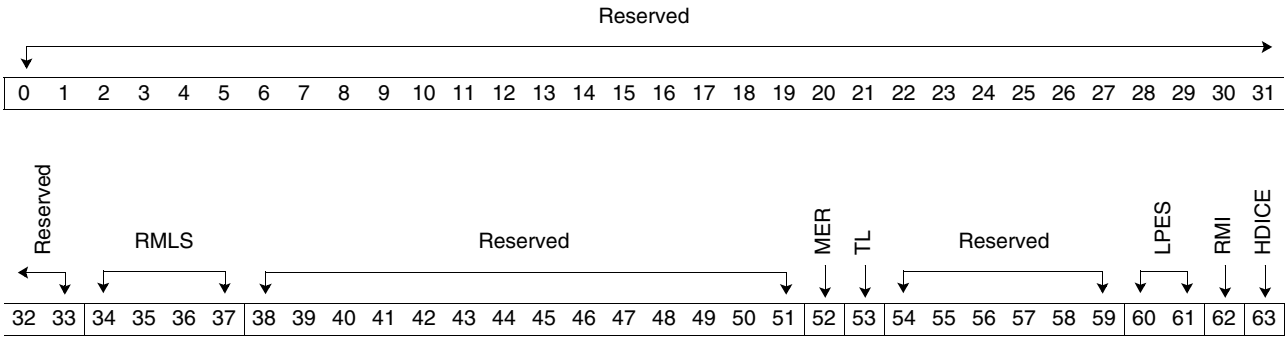
Bits	Field Name	Description
0:21	Reserved	Bits are not implemented; all bits read back zero.
22:43	HRMO	Hypervisor real-mode offset The offset from x'0' at which hypervisor real-mode memory begins.
44:63	Reserved	Bits are not implemented; all bits read back zero.

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III*.

12.1.4 Logical Partition Control Register (LPCR)

Register Short Name	LPCR	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	318	**Register Duplicated for Multithreading?**	Partially (See notes in the bit definitions below.)
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	PowerPC architected register	**Unit**	MMU



Bits	Field Name	Description
0:33	Reserved	Bits are not implemented; all bits read back zero.
34:37	RMLS	Real-mode limit selector Both threads share this field.
38:51	Reserved	Bits are not implemented; all bits read back zero.
52	MER	Mediate external exception request (interrupt enable) This field is duplicated per thread.
53	TL	Translation lookaside buffer (TLB) load Both threads share this field. 0 TLB is loaded by hardware. 1 TLB is loaded by software.
54:59	Reserved	Bits are not implemented; all bits read back zero.
60:61	LPES	Logical partitioning (environment selector) Both threads share this field.
62	RMI	Real-mode caching (caching inhibited) This field is duplicated per thread.
63	HDICE	Hypervisor decrementer interrupt control enable This field is duplicated per thread.

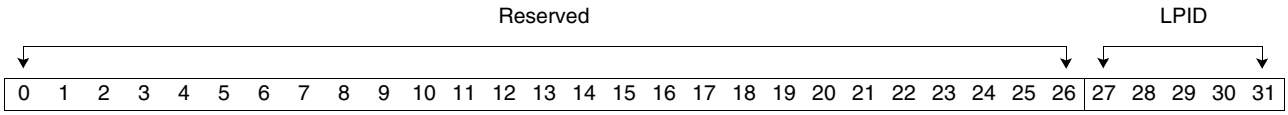
Programming Note: Mediated external interrupts are not yet defined in *PowerPC Operating Environment Architecture, Book III*. This facility is a proposed extension that is expected to be made part of the architecture.

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III*.

12.1.5 Logical Partition Identity Register (LPIDR)

Register Short Name	LPIDR	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	319	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	PowerPC architected register	**Unit**	MMU



Bits	Field Name	Description
0:26	Reserved	Bits are not implemented; all bits read back zero.
27:31	LPID	Logical partition ID

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III*.

Programming Note:

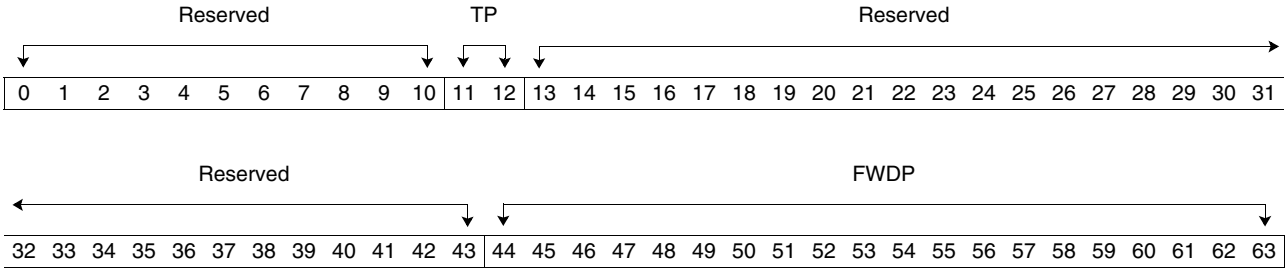
- TLB entries are tagged with the LPID when they are created. Therefore, the TLB does not need to be invalidated on a partition context switch.

12.1.6 Thread Status Register Local (TSRL)

This register allows a thread to read its own status.

Each thread has a Thread Status Register (TSR). When a thread reads its own TSR, this register is called the Thread Status Register Local (TSRL). When a thread reads the TSR for the other thread, this register is called the Thread Status Register Remote (TSRR).

Register Short Name	TSRL	**Privilege Type**	Not Privileged
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	896	**Register Duplicated for Multithreading?**	Yes
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



Bits	Field Name	Description
0:10	Reserved	Bits are not implemented; all bits read back zero.
11:12	TP	Thread priority (read/write) 00 Disabled 01 Low priority 10 Medium priority 11 High priority (If a system reset interrupt is taken, this field is set to ‘11’.) A thread cannot disable itself by attempting to directly set the TP field to ‘00’ (an attempt to do so is ignored). The thread must be disabled by setting the CTRL Register appropriately. The Thread Status Control Register (TSCR) controls which thread priorities can be selected. When in problem state (MSR[PR] = ‘1’), the following thread priorities are available: • If TSCR[UCP] is set to ‘0’, the priority cannot be changed. • If TSCR[UCP] is set to ‘1’, the priority can be set to low or medium. When in privileged but nonhypervisor state (MSR[HV,PR] = ‘00’), the following thread priorities are available: • If TSCR[UCP, PSCTP] is set to ‘00’, the priority cannot be changed. • If TSCR[UCP, PSCTP] is set to ‘10’, the priority can be set to low or medium. • If TSCR[PSCTP] is set to ‘1’, the priority can be set to low, medium, or high. When in hypervisor state (MSR[HV, PR] = ‘10’), the priority can be set to low, medium, or high. **Note:** Attempts to set the TP field illegally are ignored, and the priority does not change.
13:43	Reserved	Bits are not implemented; all bits read back zero.

Bits	Field Name	Description
44:63	FWDP	Forward progress timer (Read Only) This field is loaded from TTR[TTIM] each time the current thread completes a PowerPC Architecture instruction. This resets the timer to the maximum count. If the current thread is not disabled (TSRL[TP] = ‘00’), this field is decremented by one each time an instruction completes on the opposite thread. If TSCR[FPCF] = ‘1’ and the timer reaches x‘00001’, then after the next instruction completes, instructions for the opposite thread are flushed and no dispatch slots are given to the opposite thread until one instruction completes on the current thread. This field stops decrementing at x‘00001’ (the minimum count). This field is Initialized at POR to x‘00000’ (the maximum count).

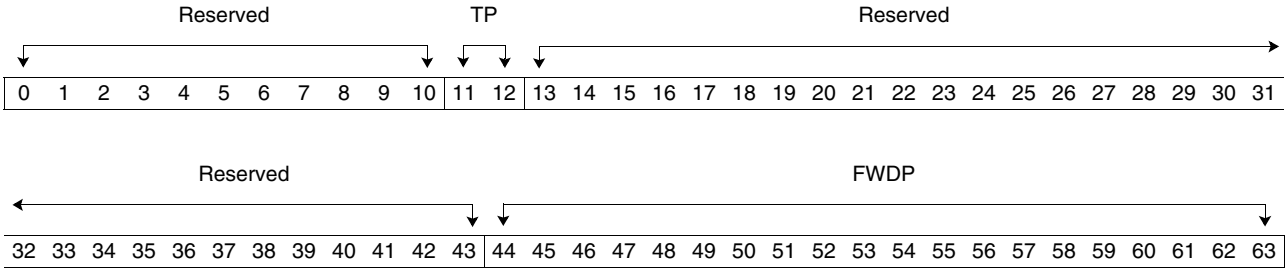
Related Registers: *Section 12.1.7 Thread Status Register Remote (TSRR)* on page 301

12.1.7 Thread Status Register Remote (TSRR)

This register allows a thread to read the status of the other thread.

Each thread has a TSR. When a thread reads its own TSR, this register is called the Thread Status Register Local (TSRL). When a thread reads the TSR of the other thread, this register is called the Thread Status Register Remote (TSRR).

Register Short Name	TSRR	**Privilege Type**	Not Privileged
Access Type	SPR Read Only	**Width**	64 bits
Decimal SPR Number	897	**Register Duplicated for Multithreading?**	Yes
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU

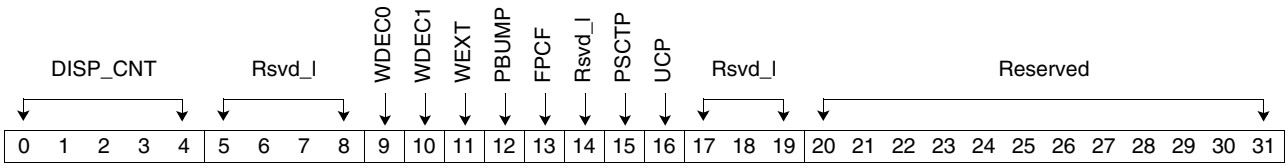


Bits	Field Name	Description
0:10	Reserved	Bits are not implemented; all bits read back zero.
11:12	TP	Thread priority. Shows the thread priority of the opposite thread. 00 Disabled 01 Low priority 10 Medium priority 11 High priority
13:43	Reserved	Bits are not implemented; all bits read back zero.
44:63	FWDP	Forward progress timer Shows the counter value of the forward progress timer for the opposite thread. See the TSRL[FWDP] field description.

Related Registers: *Section 12.1.6 Thread Status Register Local (TSRL)* on page 299

12.1.8 Thread Switch Control Register (TSCR)

Register Short Name	TSCR	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	921	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



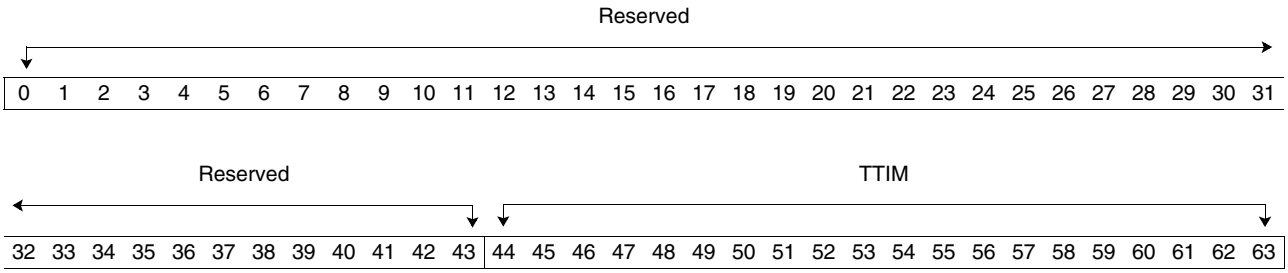
Bits	Field Name	Description
0:4	DISP_CNT	Thread dispatch count Used to control the number of dispatch cycles each thread is given, based on the priority of the thread. A DISP_CNT of '00000' equals 32, which is the maximum dispatch count. During normal operation, this field should be set to 4.
5:8	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
9	WDEC0	Decrementer wakeup enable for thread 0 0 Disabled 1 If a decrementer exception exists and the corresponding thread is suspended, the thread is activated.
10	WDEC1	Decrementer wakeup enable for thread 1 0 Disabled 1 If a decrementer exception exists and the corresponding thread is suspended, the thread is activated.
11	WEXT	External interrupt wakeup enable 0 Disabled 1 If an external interrupt exception exists and the corresponding thread is suspended, the thread is activated.
12	PBUMP	Thread priority boost enable 0 Disabled 1 If a system-caused interrupt exception is presented, the corresponding interrupt is not masked, and the priority of the corresponding thread is less than medium, sets the priority of the thread to medium. The hardware internally boosts the priority level to medium when the interrupt is pending. This does not change the value in the TSRL[TP] bits for the affected thread. The internal priority remains boosted to medium until an **mttsrl** or a priority-changing **nop** instruction occurs.
13	FPCF	Forward progress count flush enable **Note:** This bit only enables or disables the flush from occurring. The forward progress timer does not stop decrementing when set to zero. During normal operation, this bit should be set to '1'.
14	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.

Bits	Field Name	Description
15	PSCTP	Privileged but not hypervisor state change thread priority enable Enables the privileged but not the hypervisor state (MSR[HV,PR] = '00') to change priority with "**or** Rx, Rx, Rx" **nop** instructions or with writes to TSRL[TP]. 0 The ability of the privileged state to change thread priority is determined by TSCR[UCP]. 1 The privileged state can change thread priority to low, medium, or high.
16	UCP	Problem state change thread priority enable Enables the problem state to change priority with "**or** Rx, Rx, Rx" **nop** instructions or writes to TSRL[TP]. 0 The problem state cannot change thread priority. 1 The problem state can change thread priority to low or medium only.
17:19	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
20:31	Reserved	Bits are not implemented; all bits read back zero.

12.1.9 Thread Switch Timeout Register (TTR)

This register is used to ensure forward progress of the instruction dispatch.

Register Short Name	TTR	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	922	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



Bits	Field Name	Description
0:43	Reserved	Bits are not implemented; all bits read back zero.
44:63	TTIM	Thread timeout flush value A value of x‘00000’ generates the maximum count. See the description of TSRL[FWDP]. The recommended value for this field is x‘04000’. This setting, along with the TSCR[FPCF] set to ‘1’, is necessary to guarantee that a forward progress timeout does not occur because one thread prevents the other from completing instructions. If one thread executes 16 K instructions without the other thread completing one instruction, then the first thread is blocked even if it has higher priority. The second thread gets full resources to execute an instruction.

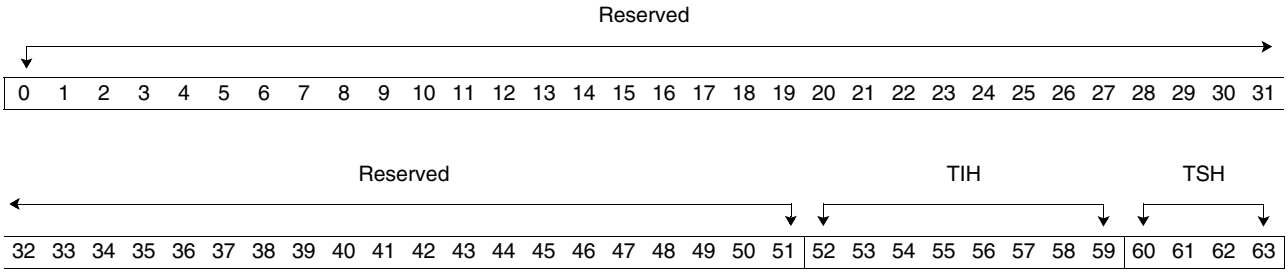
12.1.10 Translation Lookaside Buffer Special Purpose Registers

This section describes the TLB special purpose registers (SPRs).

12.1.10.1 PPE Translation Lookaside Buffer Index Hint Register (PPE_TLB_Index_Hint)

Hardware updates this register when a TLB miss occurs with LPCR[TL] set to '1' (software tablewalk mode).

Register Short Name	PPE_TLB_Index_Hint	**Privilege Type**	Read: Privileged
Access Type	SPR Read Only	**Width**	64 bits
Decimal SPR Number	946	**Register Duplicated for Multithreading?**	Yes
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MMU



Bits	Field Name	Description
0:51	Reserved	Bits are not implemented; all bits read back zero.
52:59	TIH	PPE TLB index hint The index (congruence class) of the TLB that the hardware least recently used (LRU) facility would have chosen to replace if hardware TLB updates were enabled (LPCR[TL] = '0').
60:63	TSH	TLB set hint The recommended set for replacement (software replaces entries in the correct order to maintain the LRU). Valid values are: 1000 Set 0 0100 Set 1 0010 Set 2 0001 Set 3

Programming Note:

- The PPE_TLB_Index_Hint Register is separate from the PPE_TLB_Index to avoid the possibility of hardware changing the index due to a fault while software is updating a TLB entry.
- This implementation supports a 256 × 4 TLB array. Hence, 8 fully encoded bits [52:59] are used to choose among the 256 rows (or congruence classes) of the TLB. Four fully decoded bits [60:63] are used to choose among the four columns (or sets).
- The effective address that caused the TLB miss that led to this register being set can be determined from the Data Address Register (DAR).

Related Registers: *Section 12.1.11 PPE Translation Lookaside Buffer Index Register (PPE_TLB_Index)* on page 307

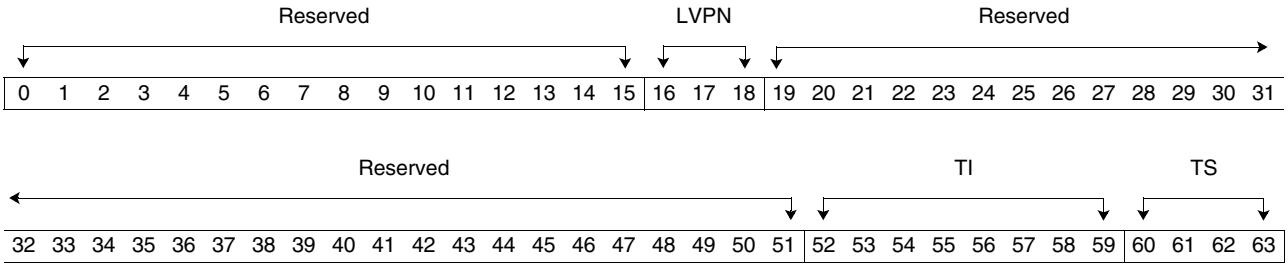
Additional Information:

- For additional information, see the *Cell Broadband Engine Architecture* document.

12.1.11 PPE Translation Lookaside Buffer Index Register (PPE_TLB_Index)

This register is readable for diagnostic purposes only.

Register Short Name	PPE_TLB_Index	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	947	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MMU



Bits	Field Name	Description
0:15	Reserved	Bits are not implemented; all bits read back zero.
16:18	LVPN	Lower virtual page number LVPN[0:2] corresponds to VPN[57:59]. **Note:** The abbreviated virtual page number (AVPN)[0:56] corresponds to VPN[0:56]. The VPN corresponds to AVPN concatenated with LVPN. The PowerPC processor unit (PPU) only implements LVPN[0:2], because the TI field of this register implies LVPN[3:12 – *p*].
19:51	Reserved	Bits are not implemented; all bits read back zero.
52:59	TI	PPE TLB index The fully encoded index of the TLB chosen for replacement.
60:63	TS	TLB set The set chosen for replacement. The following are valid set combinations: 1000 Set 0 0100 Set 1 0010 Set 2 0001 Set 3 Setting multiple bits causes multiple sets in the TLB to be written with identical data. This is not recommended because it makes inefficient use of the TLB.

Programming Note:

- This implementation supports a 256 × 4 TLB array. Hence, 8 fully encoded bits [52:59] are used to choose among the 256 rows (or congruence classes) of the TLB. Four fully decoded bits [60:63] are used to choose among the four columns (or sets).
- This register is read by the memory management unit (MMU) hardware when an **mtspr** or **mfspr** instruction is executed with a target address of either the PPE_TLB_VPN or PPE_TLB_RPN. This tells the MMU which entry of the TLB software it should update. Software writes to this register to indicate which entry it wants to replace. The register is also readable for debug purposes. Subsequent writes to the PPE_TLB_VPN and PPE_TLB_RPN are based on the last value written to this register.

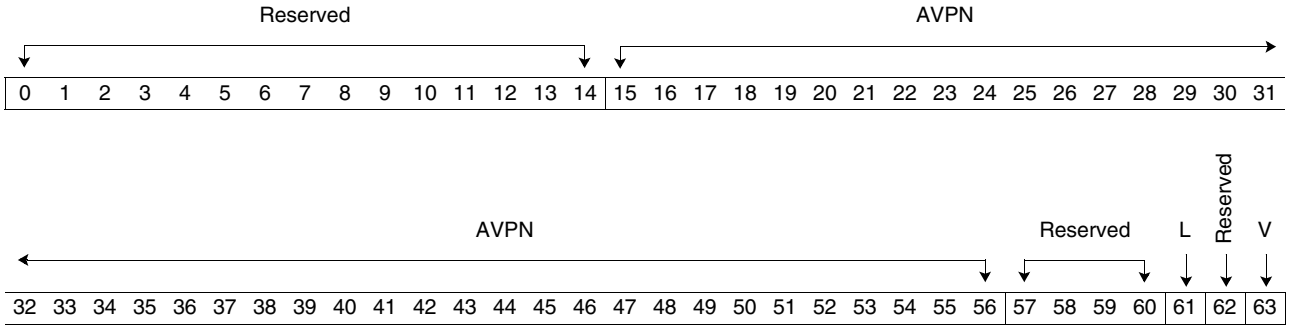
Related Registers: *Section 12.1.10.1 PPE Translation Lookaside Buffer Index Hint Register (PPE_TLB_Index_Hint)* on page 305

Additional Information:

- For additional information, see the *Cell Broadband Engine Architecture* document.

12.1.11.1 PPE Translation Lookaside Buffer Virtual-Page Number Register (PPE_TLB_VPN)

Register Short Name	PPE_TLB_VPN	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	948	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MMU



Bits	Field Name	Description
0:14	Reserved	Bits are not implemented; all bits read back zero.
15:56	AVPN	Abbreviated virtual page number The AVPN corresponds to VPN[15:56]. For a description of the AVPN, see *PowerPC Operating Environment Architecture, Book III.* **Note:** When reading a 16 MB TLB entry, bit [56] of the AVPN is undefined. Software should ignore it.
57:60	Reserved	Bits are not implemented; all bits read back zero.
61	L	Large-page mode 0 4 KB page 1 Large page
62	Reserved	Bits are not implemented; all bits read back zero.
63	V	Valid bit 0 Invalid 1 Valid

Programming Note:

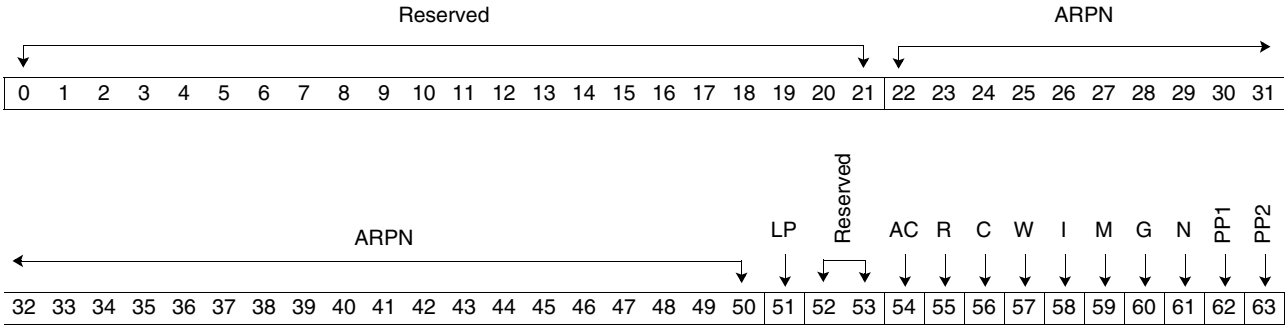
- If the VPN is being invalidated to change the protection attributes of a page or to steal the page, a TLB invalidate entry command must be issued to invalidate any cache of the effective-to-real-address translation that might be associated with the TLB entry being invalidated.
- This register acts as a placeholder for the TLB entry pointed to by PPE_TLB_Index. Changing the value of PPE_TLB_Index implicitly changes the value of this register to match the contents of the corresponding TLB array entry pointed to by PPE_TLB_Index.
- This register is meant to be written as part of a sequence of instructions.

Additional Information:

- For additional information, see the *Cell Broadband Engine Architecture* document.

12.1.11.2 PPE Translation Lookaside Buffer Real-Page Number Register (PPE_TLB_RPN)

Register Short Name	PPE_TLB_RPN	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	949	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MMU



Bits	Field Name	Description
0:21	Reserved	Bits are not implemented; all bits read back zero.
22:50	ARPN	Abbreviated real page number The ARPN corresponds to RPN[22:50]. To obtain the full 30-bit RPN, the ARPN is combined with the LP field.
51	LP	Large page size selector If PPE_TLB_VPN[L] is set to ‘1’, then bit [51] is used as the page size selector (see HID6[LB] in *Section 12.1.18 Hardware Implementation Register 6 (HID6)* on page 327). Otherwise, bit [51] corresponds to RPN[51].
52:53	Reserved	Bits are not implemented; all bits read back zero.
54	AC	Address compare
55	R	Reference The implementation always treats this bit as ‘1’. Any attempt to set it to ‘0’ is ignored.
56	C	Change
57	W	Write-through This bit is forced to ‘0’ in this implementation.
58	I	Caching inhibited
59	M	Memory coherency bit Memory is always coherent on this processor. Therefore, this value is forced to ‘1’. **Note:** Reference and Change bit updates are done with M = ‘1’. Software should not set the page table entry (PTE) M bit to ‘0’ because it might be implicitly overwritten by a Reference or Change bit update.
60	G	Guarded
61	N	No execute 0 Execute page 1 No-execute page
62	PP1	Page-Protection bit 1 for tags inactive mode

Bits	Field Name	Description
63	PP2	Page-Protection bit 2 for tags inactive mode

Programming Note:

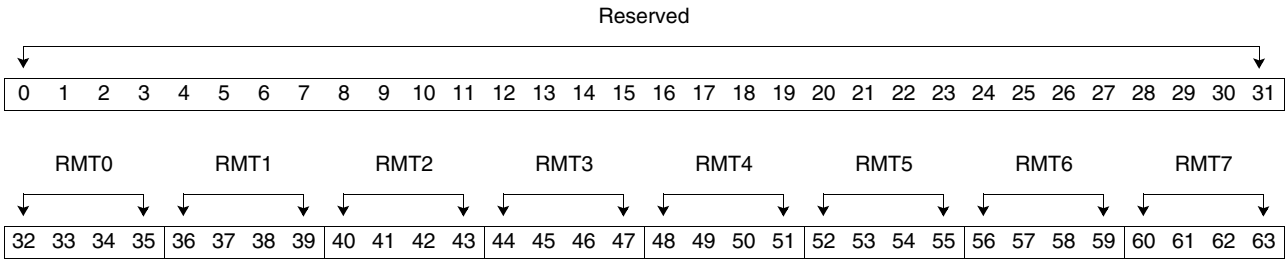
- This implementation supports two concurrent large page sizes. The LP field controls the selection between the two large page sizes when the L bit is set in the PPE_TLB_VPN Register. The address of the real page (that is, the real page number or RPN) is formed by concatenating the ARPN field with a zero when the L bit is set. Because the RPN must be on a page size boundary, software must set some low-order bits to zero when L is set or the results are undefined. For 64 KB, 1 MB, and 16 MB pages, the low-order 4, 8, and 12 bits of the RPN respectively are zero. If the L bit is not set in the PPE_TLB_VPN, the RPN is formed by concatenating the ARPN with the LP field, and the page size is 4 KB.
- This register acts as a placeholder for the TLB entry pointed to by PPE_TLB_Index. Changing the value of PPE_TLB_Index implicitly changes the value of this register to match the contents of the corresponding TLB array entry pointed to by PPE_TLB_Index.
- This register is meant to be written as part of a sequence of instructions.

Additional Information:

- For additional information, see *Cell Broadband Engine Architecture* document.

12.1.12 PPE Translation Lookaside Buffer RMT Register (PPE_TLB_RMT)

Register Short Name	PPE_TLB_RMT	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	951	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	MMU



Bits	Field Name	Description
0:31	Reserved	Bits are not implemented; all bits read back zero.
32:35	RMT0	Entry 0 of the replacement management table (RMT)
36:39	RMT1	Entry 1 of the RMT
40:43	RMT2	Entry 2 of the RMT
44:47	RMT3	Entry 3 of the RMT
48:51	RMT4	Entry 4 of the RMT
52:55	RMT5	Entry 5 of the RMT
56:59	RMT6	Entry 6 of the RMT
60:63	RMT7	Entry 7 of the RMT

Programming Note:

- Each RMT entry consists of 4 bits, which are fully decoded and correspond to a set in the TLB. If an effective address matches a range register, then the TLB considers the corresponding RMT entry for this range when replacing entries in the TLB. Each bit of the RMT entry can be thought of as a set enabler. When a bit is set to '1', it indicates that the corresponding set of the TLB is a valid entry to replace if the translation requires a TLB update to occur (for example, when the translation page table entry does not currently reside in the TLB). If multiple sets are indicated, they are replaced in priority order, beginning with invalid entries first, and then proceeding with valid entries from left to right.

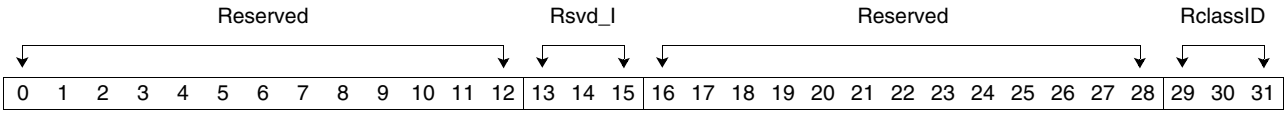
Additional Information:

- For additional information, see the *Cell Broadband Engine Architecture* document.

12.1.13 Data Range SPRs

12.1.13.1 Data Class ID Register 0 (DCIDR0)

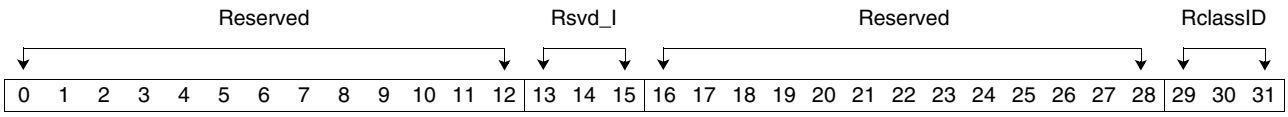
Register Short Name	DCIDR0	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	954	**Register Duplicated for Multithreading?**	Yes Each thread has a DCIDR0 and a DCIDR1.
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	XU



Bits	Field Name	Description
0:12	Reserved	Bits are not implemented; all bits read back zero.
13:15	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
16:28	Reserved	Bits are not implemented; all bits read back zero.
29:31	RclassID	Replacement class identifier

12.1.13.2 Data Class ID Register 1 (DCIDR1)

Register Short Name	DCIDR1	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	957	**Register Duplicated for Multithreading?**	Yes Each thread has a DCIDR0 and a DCIDR1.
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	XU

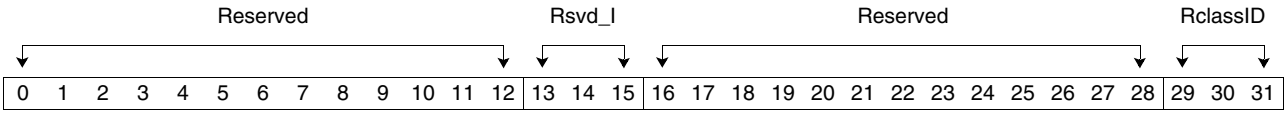


Bits	Field Name	Description
0:12	Reserved	Bits are not implemented; all bits read back zero.
13:15	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
16:28	Reserved	Bits are not implemented; all bits read back zero.
29:31	RclassID	Replacement class identifier

12.1.14 Instruction Range SPRs

12.1.14.1 Instruction Class ID Register 0 (ICIDR0)

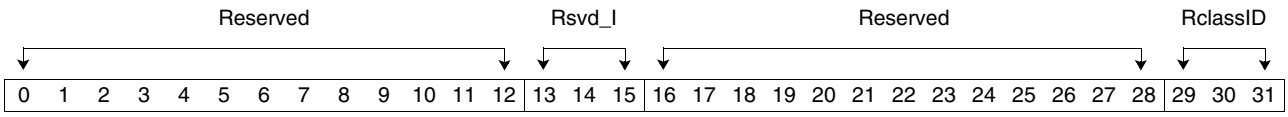
Register Short Name	ICIDR0	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	978	**Register Duplicated for Multithreading?**	Yes Each thread has an ICIDR0 and an ICIDR1.
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



Bits	Field Name	Description
0:12	Reserved	Bits are not implemented; all bits read back zero.
13:15	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
16:28	Reserved	Bits are not implemented; all bits read back zero.
29:31	RclassID	Replacement class identifier

12.1.14.2 Instruction Class ID Register 1 (ICIDR1)

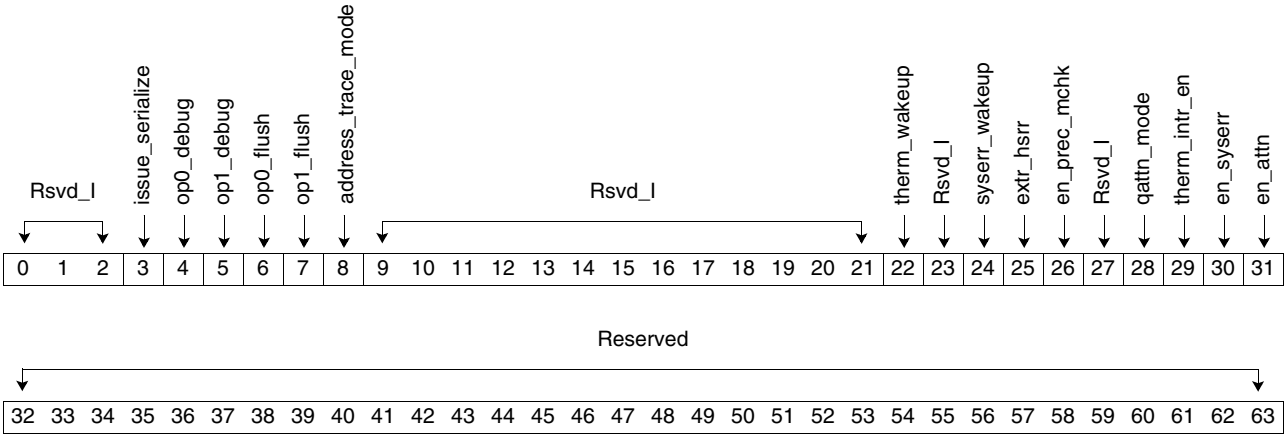
Register Short Name	ICIDR1	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	32 bits
Decimal SPR Number	981	**Register Duplicated for Multithreading?**	Yes Each thread has an ICIDR0 and an ICIDR1.
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



Bits	Field Name	Description
0:12	Reserved	Bits are not implemented; all bits read back zero.
13:15	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
16:28	Reserved	Bits are not implemented; all bits read back zero.
29:31	RclassID	Replacement class identifier

12.1.15 Hardware Implementation Register 0 (HID0)

Register Short Name	HID0	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	1008	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



Bits	Field Name	Description
0:2	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
3	issue_serialize	Issue serialize mode 0 Normal operation 1 Next instruction is not issued until all previous instructions have completed (no dual-issue either).
4	op0_debug	Opcode compare 0 takes a maintenance interrupt on an opcode compare match. This is a hardware debug facility that is not visible to software. The IU supports two opcode compare facilities that are 32-bit compares under mask control. A hit is detected when all bits are to be compared, as indicated when the bit-wise mask bits are equal. The compare is not thread based. Action is taken at the commit point of a matching instruction. 0 Opcode compare 0 does not take a maintenance interrupt. 1 Opcode compare 0 takes a maintenance interrupt on an opcode compare match.
5	op1_debug	Opcode compare 1 takes a maintenance interrupt on an opcode compare match. This is a hardware debug facility that is not visible to software. 0 Opcode compare 1 does not take a maintenance interrupt. 1 Opcode compare 1 takes a maintenance interrupt on an opcode compare match.
6	op0_flush	Opcode compare 0 causes an internal flush to that thread. 0 Opcode compare 0 does not cause an internal flush. 1 Opcode compare 0 causes an internal flush to that thread.
7	op1_flush	Opcode compare 1 causes an internal flush to that thread. 0 Opcode compare 1 does not cause an internal flush. 1 Opcode compare 1 causes an internal flush to that thread.

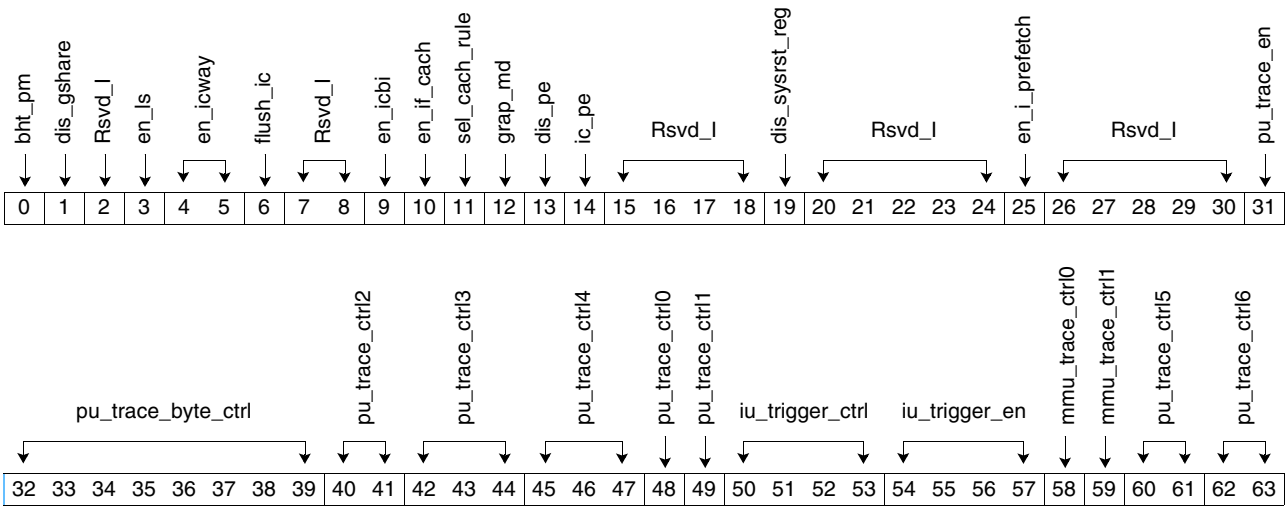
Bits	Field Name	Description
8	address_trace_mode	Address trace mode 0 Every time an address trace event occurs, the address for the new event is sent to the trace array. The partial address of the older event is recorded. 1 The whole 64-bit address is sent to the trace array every time. Events occurring while writing a 64-bit address are ignored.
9:21	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
22	therm_wakeup	Enable thermal management interrupt to wakeup suspended thread **Note:** Wakeup occurs even if HID0[therm_intr_en] = '0'.
23	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
24	syserr_wakeup	Enable system error interrupt to wakeup suspended thread Allows the system error interrupt to wake up either thread that is dormant. When a system error interrupt is received, if this bit is enabled the interrupt wakes up the thread that the interrupt was intended for (or possibly both threads). For example, if thread 0 is dormant, thread 1 is active, syserr_wakeup is set, and the interrupt is for thread 0, then thread 0 is awakened. (The active thread [thread 1] is unaffected because it is already awake.) If both threads are dormant and if syserr_wakeup is set, the interrupt awakens both threads. **Note:** Wakeup occurs even if HID0[en_syserr] = '0'.
25	extr_hsrr	Enable extended external interrupt The PowerPC Architecture specifies that external interrupts use HSRR0 and HSRR1 to save and restore external interrupts when LPCR[LPES0] = '0'. This bit also enables the mediated external interrupt feature. 0 Normal operation (direct external interrupts) 1 Enabled (mediated external interrupt enabled) **Note:** In the case of mediated external interrupts, setting LPCR[MER] before disabling a thread with an **mtctrl** instruction awakens the thread if TSCR[WEXT] is set, even if HID0[25] = '0', which normally disables mediated external interrupts.
26	en_prec_mchk	Enable precise machine check **Note:** All loads to caching-inhibited (I = '1') space cause the thread to be blocked at dispatch until data is returned for the load.
27	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
28	qattn_mode	Service processor control 0 ATTN only inactivates the thread issuing the instruction. 1 ATTN on one instruction also inactivates the other thread and causes it to go into maintenance.
29	therm_intr_en	Master thermal management interrupt enable Clearing this bit disables all thermal management interrupts regardless of the MSR state. 0 Disables all thermal management interrupts regardless of the MSR state. 1 Enabled
30	en_syserr	Enable system errors System errors generated from outside the PPE. 0 Disabled 1 Enabled
31	en_attn	Enable attention instruction (enable support processor **attn** instruction) This is a hardware debug facility that is not visible to software. It enables the **attn** instruction to quiesce the processor. If this bit is disabled, the **attn** instruction is treated as an illegal instruction.
32:63	Reserved	Bits are not implemented; all bits read back zero.

Programming Note:

- After POR, this register is set to x‘0000_0000_0000_0000’. In normal operation, the hypervisor should set this register to its preferred value of x‘0000_0047_0000_0000’ before booting the system.
- After POR, no changes to this register are necessary in typical operating mode. The only reason to change any field of this register is for diagnostic purposes in a lab environment.

12.1.16 Hardware Implementation Register 1 (HID1)

Register Short Name	HID1	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	1009	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	IU



Bits	Field Name	Description
0	bht_pm	Branch history table prediction mode 0 Static prediction. A conditional branch instruction is always predicted as not taken. 1 The branch history table (BHT) is used for branch prediction.
1	dis_gshare	Disable global branch history branch prediction 0 The global branch history is active. 1 Forces global history bits to zero (which disables the global branch history).
2	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
3	en_ls	Enable link stack 0 Disable link stack. The predicted branch target address is always x‘0’. 1 Enable link stack. All four entries of the link stack are used.
4:5	en_icway	Enable instruction cache way (one bit per way) 00 Cache disabled 10 Way A enabled 01 Way B enabled 11 Cache enabled **Note:** The L1 instruction cache is thread independent; that is, all instructions access the same cache. A and B refer to the *way* or *set* (a product of 2-way associativity in the cache).

Bits	Field Name	Description
6	flush_ic	Flush instruction cache This bit can be used to flush the instruction cache. When this bit is changed from '0' to '1', hardware detects this change and flushes the entire instruction cache. **Note:** • Software has to reset this bit after it is set (this is called a *sticky* bit). This bit has to be set to '0' before setting it to '1' to trigger a flush, which is positive-edge triggered. • The change state only occurs in a set that is enabled. If the set is disabled, then there is nothing to flush (it is already marked as invalid or already flushed).
7:8	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
9	en_icbi	Enable forced **icbi** match mode. In this mode, whenever an **icbi** is presented to the processor, the eight entries in the instruction cache that correspond to the real address (least significant 12 bits) of the line are invalidated. Eight entries are invalidated because there are four congruence classes per way where an entry can be stored because bits 50:51 of the effective address (EA) (not RA) are used to index the instruction cache. 0 Disable 1 Enable
10	en_if_cach	Enable instruction fetch cacheability control 0 All instruction fetch accesses are treated as caching inhibited (regardless of the state of the page table I bit). 1 The state of the page table I bit controls instruction fetch cacheability.
11	sel_cach_rule	Select which cacheability control rule to use 0 Configuration ring 1 Hardware implementation dependent (HID) register
12	grap_md	Graphics rounding mode The vector/single instruction, multiple data (SIMD) multimedia extension unit (VXU) operates in graphics rounding mode.
13	dis_pe	Disable parity error reporting and recovery 0 Normal operation 1 Disable
14	ic_pe	Force instruction cache parity error 0 Normal operation 1 Inject a parity error into the instruction cache. **Note:** Software has to reset this bit after it is set (this is called a *sticky* bit). This bit has to be set to '0' before setting it to '1' to trigger a parity error, which is positive-edge triggered.
15:18	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
19	dis_sysrst_reg	Disable configuration ring system reset interrupt address register 0 Enable (jump to configuration ring register value) 1 Disable (jump to x'100') This only applies to thread 0. For thread 1, always jump to x'100' on a system reset interrupt.
20:24	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
25	en_i_prefetch	Enable instruction prefetch 0 Normal operation 1 Enable
26:30	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
31	pu_trace_en	Enable PPU performance monitor/debug bus 0 PPU bus is disabled. Debug bus latches do not clock functionality. 1 PPU bus is enabled. Debug bus latches clock functionality.

Bits	Field Name	Description
32:39	pu_trace_byte_ctrl	Byte enables for PPU performance monitor bus/global debug bus 0 PPU bus is disabled for byte *x*. 1 PPU bus is enabled for byte *x*.
40:41	pu_trace_ctrl2	PPU trace bus [0:63] output control 00 Disable trace bus [0:63] (place zeros on bus). 01 Select IU [0:63] trace bus. 10 Select XU [0:63] trace bus. 11 Select VSU [0:63] trace bus.
42:44	pu_trace_ctrl3	PPU trace bus [64:95] output control 000 Disable trace bus [64:95] (place zeros on bus). 001 Select XU [0:31]. 010 Select XU [64:95]. 011 Select VSU [0:31]. 100 Select VSU [64:95]. 101 Select IU [64:95].
45:47	pu_trace_ctrl4	PPU trace bus [96:127] output control 000 Disable trace bus [96:127] (place zeros on bus). 001 Select VSU [32:63]. 010 Select VSU [96:127]. 011 Select XU [32:63]. 100 Select XU [96:127]. 101 Select IU [96:127].
48	pu_trace_ctrl0	PPU trace bus [0:63] output control 0 Place PPU trace bus [0:63] on [0:63] output. 1 Place PPU trace bus [64:127] on [0:63] output.
49	pu_trace_ctrl1	PPU trace bus [64:127] output control 0 Place PPU trace bus [64:127] on [64:127] output. 1 Place PPU trace bus [0:63] on [64:127] output.
50:53	iu_trigger_ctrl	IU trigger bus control 0 IABR match thread 0 should be placed on IU trigger bus bit *x*. 1 IABR match thread 1 should be placed on IU trigger bus bit *x*. Bit [50] controls trigger bus bit [0]. Bit [51] controls trigger bus bit [1]. Bit [52] controls trigger bus bit [2]. Bit [53] controls trigger bus bit [3].
54:57	iu_trigger_en	IU trigger bus enable 0 Pass XU trigger bus onto PPU trigger bus for bit *x*. 1 Pass IU trigger bus onto PPU trigger bus for bit *x*. Bit [54] controls trigger bus bit [0]. Bit [55] controls trigger bus bit [1]. Bit [56] controls trigger bus bit [2]. Bit [57] controls trigger bus bit [3].
58	mmu_trace_ctrl0	MMU ramp controls for PPU trace bus [0:31] 0 Do not place MMU trace bus on final PPU trace bus output bits [0:31]. 1 Place MMU trace bus on final PPU trace bus output bits [0:31].
59	mmu_trace_ctrl1	MMU ramp controls for PPU trace bus [64:95] 0 Do not place MMU trace bus on final PPU trace bus output bits [64:95]. 1 Place MMU trace bus on final PPU trace bus output bits [64:95].

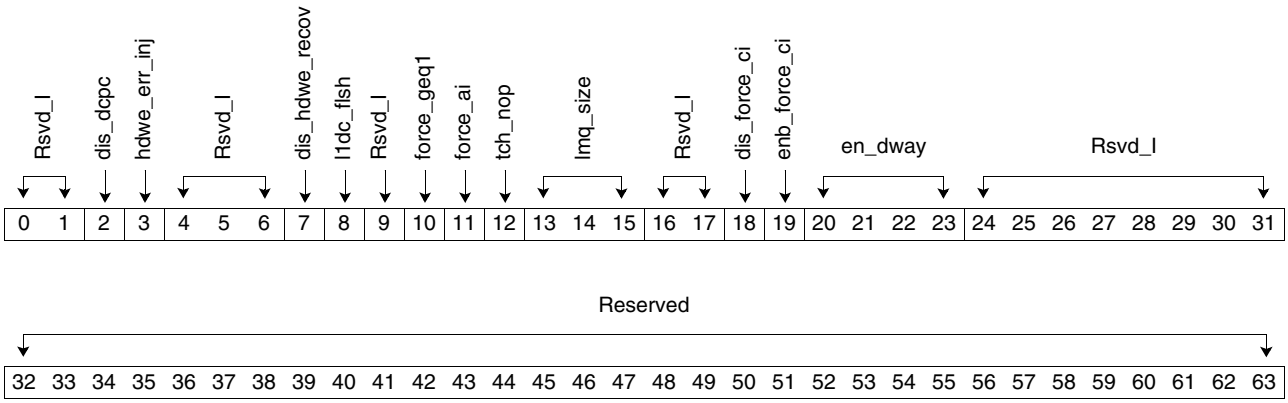
Bits	Field Name	Description
60:61	pu_trace_ctrl5	PPU trace bus [0:31] output control 00 Place PPU trace bus [0:31] on [0:31] output. 01 Place PPU trace bus [32:63] on [0:31] output. 10 Place PPU trace bus [64:95] on [0:31] output.
62:63	pu_trace_ctrl6	PPU trace bus [32:63] output control 00 Place PPU trace bus [32:63] on [32:63] output. 01 Place PPU trace bus [0:31] on [32:63] output. 10 Place PPU trace bus [96:127] on [32:63] output.

Programming Note:

- After POR, this register is set to x‘0000_0000_0000_0000’. In normal operation, the hypervisor should set this register to its preferred value of x‘9C30_1040_0000_0000’ before booting the system.
- After POR, no changes to this register are necessary in typical operating mode. The only reason to change any field of this register is for diagnostic purposes in a lab environment.

12.1.17 Hardware Implementation Register 4 (HID4)

Register Short Name	HID4	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	1012	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	XU



Bits	Field Name	Description
0:1	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
2	dis_dcpc	Disable data cache parity checking Prevents Fault Isolation Register (FIR) updates and recovery. 0 Normal operation 1 Disable data cache parity checking.
3	hdwe_err_inj	Inject parity error into cache Forces a 1-bit error somewhere in both of the 32-byte blocks that are being reloaded to the data cache. Used for hardware initialization. 0 Normal operation 1 Inject a parity error into the cache.
4:6	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
7	dis_hdwe_recov	Disable data cache parity error hardware recovery FIR bits are still reported to the MMU to cause a machine check or checkstop. 0 Normal operation 1 Disable data cache parity error hardware recovery.
8	l1dc_flsh	L1 data cache flash invalidate 0 Normal operation 1 All sectors set to invalid and held invalid. (Implemented as an edge detect) **Note:** • Software has to reset this bit after it is set (this is called a *sticky* bit). This bit has to be set to '0' before setting it to '1' to trigger a flush, which is positive-edge triggered. • The change state only occurs in a set that is enabled. If the set is disabled, then there is nothing to flush (it is already marked as invalid or already flushed). • Implemented as an edge detect.

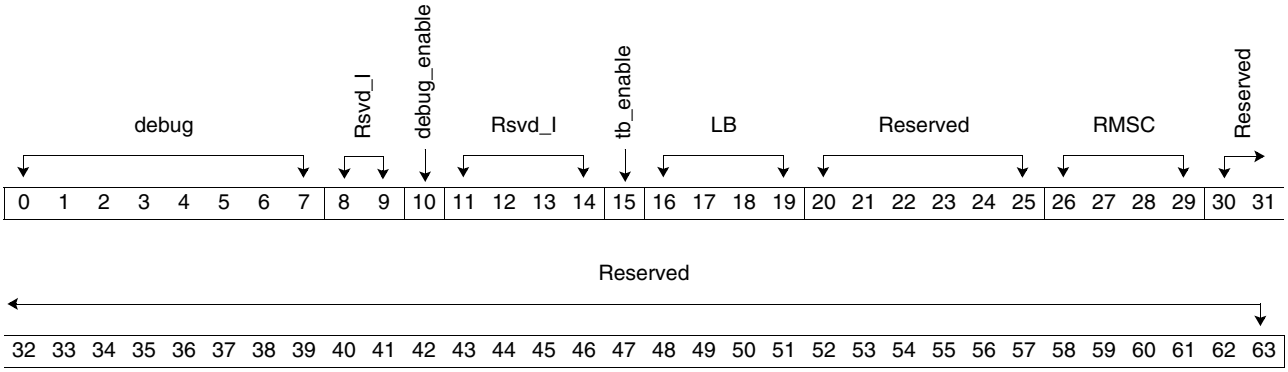
Bits	Field Name	Description
9	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
10	force_geq1	Force all load instructions to be treated as if being loaded to guarded storage 0 Normal operation 1 Force all load instructions to be treated as if being loaded to guarded storage. In the PPU, this causes touch instructions (such as **dcbt**) to be treated as **nop** instructions. This might also have an effect on the ordering of stores in the core interface unit (CIU).
11	force_ai	Force an alignment interrupt instead of microcode on unaligned operations Used for debugging. Prevents misaligned flushes. Any load or store that spans a 32-byte boundary (or an 8-byte boundary in DABR mode) takes an alignment interrupt. 0 Normal operation 1 Force alignment interrupt.
12	tch_nop	Force data cache block touch x-form (**dcbt**) and data cache block touch for store (**dcbtst**) instructions to function like **nop** instructions. Only performs the translation. 0 Normal operation 1 Force **dcbt** and **dcbtst** to function like **nop** instructions.
13:15	lmq_size	Maximum number of outstanding demand requests to the memory subsystem (only applies to loads) 000 Eight outstanding requests to the PowerPC processor storage subsystem (PPSS) 111 Seven outstanding requests to the PPSS 110 Six outstanding requests to the PPSS 101 Five outstanding requests to the PPSS 100 Four outstanding requests to the PPSS 011 Three outstanding requests to the PPSS 010 Two outstanding requests to the PPSS 001 One outstanding requests to the PPSS
16:17	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
18	dis_force_ci	0 Force data cache inhibit, unless HID4[19] disables. 1 Use effective-address-to-real-address translation (ERAT) cache inhibit bit (normal translation).
19	enb_force_ci	Enable force_data_cache_inhibit (only valid if dis_force_ci = 0) 0 Use configuration ring (PPU/XU bit 174 [cfg_force_ci]). 1 Use HID4[18]
20:23	en_dway	Enable L1 data cache way (one bit per way) Bit 20 enables L1 data cache way A when set to ‘1’. Bit 21 enables L1 data cache way B when set to ‘1’. Bit 22 enables L1 data cache way C when set to ‘1’. Bit 23 enables L1 data cache way D when set to ‘1’. **Notes:** • At POR, the cache is disabled. • When all bits are zero, no writes to the L1 data cache occur. • When the L1 data cache is completely disabled, microcoded loads and stores do not work. • Changing the value of these bits at any state other than immediately after a POR is not recommended (undefined behavior might result). If any tag in the cache has been allocated (that is, a valid bit is on), then a flash invalidate of the tag must occur. This can be achieved by setting and then resetting HID4(8).
24:31	Rsvd_I	Reserved. Latch bit is implemented; the value read is the value written.
32:63	Reserved	Bits are not implemented; all bits read back zero.

Programming Note:

- After POR, this register is set to x‘0000_0000_0000_0000’. In normal operation, the hypervisor should set this register to its preferred value of x‘0000_3F00_0000_0000’ before booting the system.
- After POR, no changes to this register are necessary in typical operating mode. The only reason to change any field of this register is for diagnostic purposes in a lab environment.

12.1.18 Hardware Implementation Register 6 (HID6)

Register Short Name	HID6	**Privilege Type**	Hypervisor
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	1017	**Register Duplicated for Multithreading?**	No
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Hardware implementation register	**Unit**	MMU



Bits	Field Name	Description
0:7	debug	Debug and performance select x‘80’ Selects performance bus (default value) x‘40’ MMU control and SPR read debug x‘20’ Snoop interface and SLB control debug x‘10’ PPU interface debug x‘08’ LRU and SPR control debug x‘04’ Timebase debug x‘02’ TLB control and snoop control debug x‘01’ Tablewalk and PPE TLB index debug No other combinations are meaningful. This field is for hardware debug purposes only.
8:9	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
10	debug_enable	Turn on the performance or debug bus for the MMU (see bits [0:7] for performance and debug select controls). This field is for hardware debug purposes only.
11:14	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
15	tb_enable	Time-base and decrementer facility enable 0 TBU, TBL, DEC, HDEC, and the hang-detection logic do not update (all update signals from the pervasive logic are ignored). 1 TBU, TBL, DEC, HDEC, and the hang-detection logic are enabled to update.

Bits	Field Name	Description
16:19	LB	Large page bit table If L = '1' and LP = '0', then Large Page Size 1 is used. If L = '1' and LP = '1', then Large Page Size 2 is used. Large Page Size: Size 1 Size 2 0000 16 MB 16 MB 0001 16 MB 1 MB 0010 16 MB 64 KB 0100 1 MB 16 MB 0101 1 MB 1 MB 0110 1 MB 64 KB 1000 64 KB 16 MB 1001 64 KB 1 MB 1010 64 KB 64 KB All other combinations are reserved.
20:25	Reserved	Bits are not implemented; all bits read back zero.
26:29	RMSC	PPE real-mode storage control facility
30:63	Reserved	Bits are not implemented; all bits read back zero.

Programming Note:

- After POR, this register is set to x'0000_0000_0000_0000'. In normal operation, the hypervisor sets HID6[tb_enable] to '1'.
- Programming details for HID6[tb_enable], HID6[LB], and HID6[RMSC] can be found in the sections listed under "Additional Information" for each field description.

12.1.19 CBEA-Compliant Processor Version Register (BP_VR)

The Cell Broadband Engine Architecture (CBEA)-Compliant Processor Version Register is a 32-bit read-only register that contains values that identify the version and revision level of the CBEA-compliant processor. The contents of the CBEA-Compliant Processor Version Register are only accessible using a PPE move-from special-purpose register (**mfspr**) instruction. Read access to the CBEA-Compliant Processor Version Register is privileged; write access is not provided. There is only one CBEA-Compliant Processor Version Register per CBEA-compliant processor.

Version numbers are assigned by the CBEA process. Revision numbers are assigned by an implementation-defined process. The values currently assigned to these fields are as follows:

Design Level	Version	Revision
10KE DD 1.0	x‘0000’	x‘0000’
10KE DD 1.1	x‘0000’	x‘0001’
10KE DD 2.0	x‘0000’	x‘0100’
10KE DD 3.0	x‘0000’	x‘0200’
10KE DD 3.1	x‘0000’	x‘0201’
10KE DD 3.2	x‘0000’	x‘0202’
65 nm DD 1.1	x‘0000’	x‘1000’

Note: The BP_VR is updated if the latch-to-latch paths of every unit in the CBE processor are not logically and functionally equivalent to the previous version (that is, if any unit is not Verity clean with the previous version).

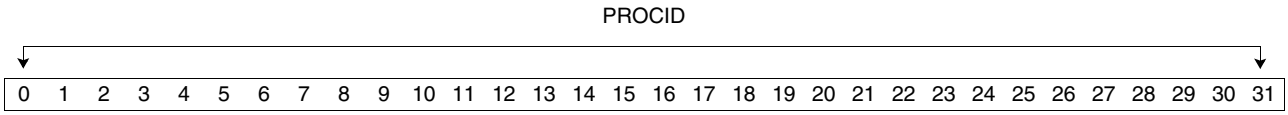
Register Short Name	BP_VR	**Privilege Type**	Read: Privileged
Access Type	SPR Read only	**Width**	32
Decimal SPR Number	1022	**Register Duplicated for Multithreading?**	No
Value at Initial POR (for DD 3.2)	x‘00000000_00000202’	**Value During POR Set By**	Hardwired
Value at Initial POR	x‘00000000_00001000’	**Value During POR Set By**	Hardwired
Specification Type	CBEA architected register	**Unit**	XU

Additional Information:

- For additional information, see *Cell Broadband Engine Architecture.*

12.1.20 Processor Identification Register (PIR)

Register Short Name	PIR	**Privilege Type**	Read: Privileged
Access Type	SPR Read only	**Width**	32 bits
Decimal SPR Number	1023	**Register Duplicated for Multithreading?**	Yes
Value at Initial POR	[0:22] 0 [23:30] Set by configuration chain. [31] 0 for thread 0 1 for thread 1	**Value During POR Set By**	Scan initialization during POR Configuration ring
Specification Type	PowerPC architected register	**Unit**	XU



Bits	Field Name	Description
0:31	PROCID	Processor ID

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III*.

12.2 Special Architected Registers

This section describes registers that have been architected to meet special requirements. These registers have unique characteristics, uses, and applications.

12.2.1 Machine State Register (MSR)

Register Short Name	MSR	**Privilege Type**	Privileged
Access Type	SPR Read/Write	**Width**	64 bits
Decimal SPR Number	N/A	**Register Duplicated for Multithreading?**	Yes
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	PowerPC architected register	**Unit**	IU

Bits	Field Name	Description
0	SF	64-bit mode 0 The processor is in 32-bit mode. 1 The processor is in 64-bit mode.
1	TA	Tags-active mode This mode is not supported. This bit is reserved and is forced to zero.
2	Reserved	Bits are not implemented; all bits read back zero.
3	HV	Hypervisor state 0 The processor is not in hypervisor state. 1 If MSR[PR] = ‘0’, then the processor is in hypervisor state; otherwise, the processor is in problem state.
4:37	Reserved	Bits are not implemented; all bits read back zero.
38	VXU	VXU 0 The processor cannot execute VXU instructions. If the processor attempts to execute a VXU instruction, this causes a VXU unavailable interrupt. 1 The processor can execute VXU instructions.
39:46	Reserved	Bits are not implemented; all bits read back zero.
47	ILE	Interrupt little-endian mode This mode is not supported. This bit is reserved and forced to ‘0’.

Bits	Field Name	Description
48	EE	External interrupt enable 0 External and decrementer interrupts are disabled. 1 External and decrementer interrupts are enabled.
49	PR	Problem state 0 The processor is in privileged state. 1 The processor is in problem state.
50	FP	Floating-point available 0 The processor cannot execute any floating-point instructions. 1 The processor can execute floating-point instructions.
51	ME	Machine check enable 0 Machine check interrupts are disabled. 1 Machine check interrupts are enabled. **Note:** This bit is a hypervisor resource, and can only be modified by **rfid** and **hrfid** instructions while in hypervisor mode. (See *PowerPC Operating Environment Architecture, Book III.*)
52	FE0	Floating-point exception 0 The PPE does not support imprecise-unrecoverable mode or imprecise-recoverable mode. The Floating-Point Exception Mode bits, FE0 and FE1, are interpreted as shown below: [FE0] [FE1] Mode 0 0 Ignore exceptions mode 0 1 Precise mode 1 0 Precise mode 1 1 Precise mode
53	SE	Single-step trace enable 0 The processor executes instructions normally. 1 The processor generates a single-step type of trace interrupt after successfully completing the execution of the next instruction (unless that instruction is **rfid**, **hrfid**, **attn**, or **sc**, which are never traced). Successful completion signifies that the instruction caused no other interrupt.
54	BE	Branch trace enable 0 The processor executes branch instructions normally. 1 The processor generates a branch type of trace interrupt after completing the execution of a branch instruction, whether or not the branch is taken.
55	FE1	Floating-point exception 1 **Note:** See the description of the FE0 bit in this register
56	Rsvd_l	Reserved. Latch bit is implemented; the value read is the value written.
57	Reserved	Bits are not implemented; all bits read back zero.
58	IR	Instruction relocate 0 Instruction address translation is off. 1 Instruction address translation is on.
59	DR	Data relocate 0 Data address translation is off. 1 Data address translation is on.
60	Reserved	Bits are not implemented; all bits read back zero.
61	PMM	Performance monitor mark This bit is reserved and forced to zero.
62	RI	Recoverable interrupt 0 Interrupt is not recoverable. 1 Interrupt is recoverable.

Bits	Field Name	Description
63	LE	Little-endian mode This mode is not supported. This bit is reserved and forced to ‘0’, which means the mode is always big-endian.

Programming Note:

- A system reset sets the POR value of this register to the value indicated by the system reset interrupt. This value is x‘9000_0000_0000_0000’ (where the 9 means MSR[0] is set to ‘1’ and MSR[3] is set to ‘1’).

Additional Information:

- For additional information, see *PowerPC Operating Environment Architecture, Book III.*

Appendix A. Registers Defined in the CBEA

The tables below list the Cell Broadband Engine (CBE) registers whose implementation is identical to the architecture.

A.1 SPE Privilege 1 Registers

Table A-1. SPE Privilege 1 Registers

Register Name and (Short Name)	Additional Information
Class 1 Interrupt Mask Register (INT_Mask_class1)	See *Cell Broadband Engine Architecture*
Class 1 Interrupt Status Register (INT_Stat_class1)	See *Cell Broadband Engine Architecture*
MFC Address Compare Control Register (MFC_ACCR)	*Section A.5.1* on page 336
MFC Atomic Flush Register (MFC_Atomic_Flush)	*Section A.5.2* on page 336
MFC Data Address Register (MFC_DAR)	*Section A.5.3* on page 337
MFC Data-Storage Interrupt Status Register (MFC_DSISR)	*Section A.5.4* on page 337
MFC Real Mode Address Boundary Register (MFC_RMAB)	*Section A.5.5* on page 337
MFC State Register 1 (MFC_SR1)	*Section A.5.6* on page 337
MFC TLB Invalidate All Register (MFC_TLB_Invalidate_All)	Not implemented
MFC TLB Replacement Management Table Index Register (MFC_TLB_RMT_Index)	Not implemented. See *Section A.5.7* on page 337.
MFC Version Register (MFC_VR)	*Section A.5.8* on page 338
SPU Version Register (SPU_VR)	*Section A.5.9* on page 339

A.2 SPE Privilege 2 Registers

Table A-2. SPE Privilege 2 Registers

Register Name and (Short Name)	Additional Information
MFC Control Register (MFC_CNTL)	*Section A.6.1* on page 340
SLB Effective Segment ID Register (SLB_ESID)	*Section A.6.2* on page 340
SLB Invalidate All Register (SLB_Invalidate_All)	*Section A.6.3* on page 340
SPU Channel Data Register (SPU_ChnlData)	*Section A.6.4* on page 340
SPU Channel Index Register (SPU_ChnlIndex)	*Section A.6.5* on page 340
SPU Configuration Register (SPU_Cfg)	*Section A.6.6* on page 341
SPU Outbound Interrupt Mailbox Register (SPU_OutIntrMbox)	*Section A.7* on page 341
SPU Privileged Control Register (SPU_PrivCntl)	*Section A.7.1* on page 341

A.3 SPE Problem State Registers

Table A-3. SPE Problem State Registers

Register Name and (Short Name)	Additional Information
MFC Command Tag Register (MFC_Tag)	*Section A.8.1* on page 341
MFC Effective Address Low Register (MFC_EAL)	*Section A.8.3* on page 342
MFC Transfer Size Register (MFC_Size)	*Section A.8.4* on page 342
Proxy Tag-Group Query Mask Register (Prxy_QueryMask)	*Section A.8.5* on page 342
Proxy Tag-Group Query Type Register (Prxy_QueryType)	*Section A.8.6* on page 342
Proxy Tag-Group Status Register (Prxy_TagStatus)	*Section A.8.7* on page 342
SPU Signal Notification Register 1 (SPU_Sig_Notify_1)	*Section A.8.8* on page 342
SPU Signal Notification Register 2 (SPU_Sig_Notify_2)	*Section A.8.9* on page 342
SPU Outbound Mailbox Register (SPU_Out_Mbox)	*Section A.9.1* on page 343
SPU Inbound Mailbox Register (SPU_In_Mbox)	*Section A.9.2* on page 343

A.4 BIC 0 and BIC 1 MMIO Registers

Table A-4. BIC 0 and BIC 1 MMIO Memory Map (BClk Domain)

Register Name and (Short Name)	Additional Information
Interface n Initialization Register (IFnINIT [n = 0,1])	*Section A.10.1* on page 344.

A.5 SPE Privilege 1 Registers

A.5.1 MFC Address Compare Control Register (MFC_ACCR)

The MFC_ACCR allows the detection of direct memory access (DMA) to a virtual page marked with the address compare (AC) bit in the page-table entry (PTE) set and a range within the local storage.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.5.2 MFC Atomic Flush Register (MFC_Atomic_Flush)

The MFC_Atomic_Flush Register is implementation dependent, and access is privileged. Privileged software uses this register to clear the contents of the cache used for atomic DMA commands and TLB-update operations. Data in the cache that is considered modified is pushed to memory, and the line is invalidated. Valid lines in the cache that are not considered modified are invalidated. The reservation is reset. For this operation to work properly, privileged software must suspend all memory flow controller (MFC) operations.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.5.3 MFC Data Address Register (MFC_DAR)

The MFC_DAR contains the 64-bit effective address (EA) from the DMA requests.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.5.4 MFC Data-Storage Interrupt Status Register (MFC_DSISR)

The MFC_DSISR Register contains status bits relating to the data-storage interrupts (DSI) generated by the SMM.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.5.5 MFC Real Mode Address Boundary Register (MFC_RMAB)

The MFC supports a 4-bit, real-mode boundary (RMB) field. See the *Cell Broadband Engine Architecture* document for more information about this register.

A.5.6 MFC State Register 1 (MFC_SR1)

The MFC_SR1 Register contains configuration information controlled by a hypervisor. Access to this register is privileged. This register corresponds to the PowerPC Processor Element (PPE) Machine State Register (MSR).

In this implementation, MFC_SR1[58], the SPU Master Run Control (S bit), functions as an SPU-enable control. This bit allows privileged code the capability to temporarily suspend synergistic processor unit (SPU) execution and resume it again without affecting the SPU status. If the SPU is stopped and MFC_SR1[58] is disabled (set to ‘0’), then the SPU stays stopped. That is, the status is unchanged from the last stopped reason code in the SPU Status Register (SPU_Status). If MFC_SR1[58] is written with a start command, the SPU does not start and the SPU status remains unchanged; the last stopped reason code remains the same from the last stopped condition. However, reading the SPU Run Control Register (SPU_RunCntl) shows that the run command was accepted. If MFC_SR1[58] is then enabled (set to ‘1’), the SPU remains stopped. When the SPU_RunCntl Register is next written with a run command, the SPU starts execution and the status is updated accordingly.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.5.7 MFC TLB Replacement Management Table Index Register (MFC_TLB_RMT_Index)

This register is not implemented in this version of the CBE.

Implementation Note: Only one translation fault may be outstanding. The implementation can either stop all command queue processing on the first translation error or continue processing. If processing is continued, all ordering rules must be followed (a command must not be processed if it is dependent on a command that is waiting for a translation fault to be resolved). The state of the MFC must appear as if the command (or partial command) were never issued. This is also the case if a second translation fault occurs.

A.5.8 MFC Version Register (MFC_VR)

The MFC_VR Register contains a 32-bit value that identifies the specific version (model) and revision level of the MFC portion of the Cell Broadband Engine Architecture. The contents of this register are accessible from the PPE by using a load doubleword (**ld**) instruction. Read access to the MFC_VR Register from the PPE is privileged, and write access is not provided. Access to this register from the MFC is not provided. There is one MFC_VR Register for each MFC in a Cell Broadband Engine.

Version numbers are assigned by the MFC architecture process. Revision numbers are assigned by an implementation-defined process.

The MFC_VR Register distinguishes between processors that differ in attributes that may affect software. It contains two fields: Version and Revision. The values currently assigned to these fields are as follows:

Design Level	Version	Revision
10KE DD 1.0	x‘0000’	x’0000’
10KE DD 1.1	x‘0000’	x’0001’
10KE DD 2.0	x‘0000’	x’0100’
10KE DD 3.0	x‘0000’	x’0200’
10KE DD 3.1	x‘0000’	x’0201’
10KE DD 3.2	x‘0000’	x’0202’
65 nm DD 1.1	x‘0000’	x’1000’

Note: The MFC_VR is updated if the latch-to-latch path of the MFC is not logically and functionally equivalent to the previous version (that is, if the unit is not Verity clean with the previous version).

See the *Cell Broadband Engine Architecture* document for more information about this register. The ‘Value at Initial POR’ listed below is for CMOS SOI 65 nm DD 1.1.

Register Short Name	MFC_VR	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400018’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_00001000’	**Value During POR Set By**	Hardwired
Specification Type	CBEA architected register	**Unit**	MFC

Related Registers: *Section 12.1.19 CBEA-Compliant Processor Version Register (BP_VR)* and *Appendix A.5.9 SPU Version Register (SPU_VR)*

A.5.9 SPU Version Register (SPU_VR)

The SPU_VR Register contains a 32-bit value that identifies the specific version (model) and revision level of the SPU portion of the Cell Broadband Engine Architecture. The contents of this register are accessible from the PPE using a load doubleword (**ld**) instruction. Read access to the SPU_VR Register from the PPE is privileged, and write access is not provided. Access to this register from the SPU is not provided. There is one SPU_VR Register for each SPU in a Cell Broadband Engine.

Version numbers are assigned by the SPU architecture process; revision numbers are assigned by an implementation-defined process.

The SPU_VR Register distinguishes between processors that differ in attributes that may affect software. It contains two fields: Version and Revision. The values currently assigned to these fields are as follows:

Design Level	Version	Revision
10KE DD 1.0	x‘0000’	x‘0000’
10KE DD 1.1	x‘0000’	x‘0000’
10KE DD 2.0	x‘0000’	x‘0100’
10KE DD 3.0	x‘0000’	x‘0200’
10KE DD 3.1	x‘0000’	x‘0201’
10KE DD 3.2	x‘0000’	x‘0202’
65 nm DD 1.1	x‘0000’	x‘1000’

Note: The SPU_VR is updated if the latch-to-latch path of the SPU is not logically and functionally equivalent to the previous version (that is, if the unit is not Verity clean with the previous version). The SPU_VR is also updated if the BP_VR is updated, even if the SPU itself is Verity clean.

See the *Cell Broadband Engine Architecture* document for more information about this register. The ‘Value at Initial POR’ listed below is for CMOS SOI 65 nm DD 1.1.

Register Short Name	SPU_VR	**Privilege Type**	Privilege 1
Access Type	MMIO Read Only	**Width**	64 bits
Hex Offset From BE_MMIO_Base	SPE*n*: x‘400020’ + (x‘02000’ x *n*)	**Memory Map Area**	SPE Privilege 1
Value at Initial POR	x‘00000000_00001000’	**Value During POR Set By**	Hardwired
Specification Type	CBEA architected register	**Unit**	MFC

Related Registers: *Section 12.1.19 CBEA-Compliant Processor Version Register (BP_VR)* and *Appendix A.5.8 MFC Version Register (MFC_VR)*

A.6 SPE Privilege 2 Registers

A.6.1 MFC Control Register (MFC_CNTL)

The restart DMA operation is only effective if the DMA unit is in normal DMA queue operational status.

The purge bit can be used in the context save/restore for clearing out the queue after an unrecoverable error condition or when cleaning the SPE to prepare it for a new context.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.6.2 SLB Effective Segment ID Register (SLB_ESID)

To properly load the array with data, the segment lookaside buffer (SLB) requires a write sequence similar to the one for the translation lookaside buffer (TLB). First, the index must be written to specify the entry for loading. The virtual segment ID (VSID) and effective segment ID (ESID) fields are written independently, unlike the TLB writes, but the SLB_VSID write should follow the index. The SLB_ESID data is written last because it contains the Valid bit, and the entry should not be valid until all data is loaded.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.6.3 SLB Invalidate All Register (SLB_Invalidate_All)

A write to SLB_Invalidate_All causes the Valid bit in all SLB entries to be set to ‘0’, making the entries invalid. The remaining fields of each entry are undefined.

Writes to this register can be either 64-bit or 32-bit operations. Any write to the least significant word causes an entry in the SLB to be invalidated.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.6.4 SPU Channel Data Register (SPU_ChnlData)

If the data fields in the register are less than 32 bits (such as the SPU_RdEventStat channel, the SPU_WrEventMask channel, and the MFC_RdListStallStat channel), then only those bits defined in the register are updated, and the remaining bits of write data are ignored.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.6.5 SPU Channel Index Register (SPU_ChnlIndex)

The SPU Channel Index Register selects which SPU channel is accessed using the SPU Channel Count Register or the SPU Channel Data Register. Access to this register is privileged. Setting this register to channels that are not accessible by the channel count or channel data has no effect. Reads of the nonaccessible channels cause the channel count or channel data to return zeros. Writes are ignored. See the *Cell Broadband Engine Architecture* document for bit definitions.

Note: In isolation mode, this register is forced to the SPU_WrDec channel, and all writes are ignored. The SPU_WrDec channel cannot be accessed through the channel count or channel data registers, therefore writes to this register have no effect, and reads return zeros.

A.6.6 SPU Configuration Register (SPU_Cfg)

The SPU Configuration Register is used to read or set the configuration of the SPU Signal Notification Registers (SPU_Sig_Notify_1 and SPU_Sig_Notify_2) in the SPUs.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.7 SPU Outbound Interrupt Mailbox Register (SPU_OutIntrMbox)

The PPU reads the mailbox data from the SPU in the SPU_OutIntrMbox Register. The SPU stores to this mailbox by writing to the SPU_WrOutIntrMbox channel (see the *Cell Broadband Engine Architecture* document for more information about the SPU Write Outbound Interrupt Mailbox Channel). When the SPU writes to the SPU_WrOutIntrMbox channel, the SPU_WrOutIntrMbox channel counter decrements from 1 to 0. Reading this register causes the SPU_WrOutIntrMbox channel counter to increment to 1. While the count is 1, further reads to this register do not affect the count, and the read data is the last stored value in this register. The queue depth for this implementation is 1.

Note: For the SPU_WrOutIntrMbox channel, the count is 1 at POR. Software reinitializes the count to 1.

This register has the same behavior as the SPU_Out_Mbox except that the SPU_OutIntrMbox register has a PPU Mailbox Interrupt that allows the SPU to notify the PPU when the SPU has written data to the PPU Mailbox. The PPU Mailbox interrupt is asserted when the SPU_WrOutIntrMbox channel count transitions from 1 to 0, and deasserted when the SPU_WrOutIntrMbox channel counter increments from 0 to 1. The interrupt is taken when it is asserted and enabled. See the *Class 2 Interrupt Mask Register (INT_Mask_class2)* for information about how to enable this interrupt.

See the *Cell Broadband Engine Architecture* document for more information about this register.

Related Register: See the *SPU Outbound Mailbox Register (SPU_Out_Mbox)* on page 343.

A.7.1 SPU Privileged Control Register (SPU_PrivCntl)

The SPU Privileged Control Register provides privileged software with the ability to control the execution environment of the SPU. Access to this register is privileged.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8 SPE Problem State Registers

A.8.1 MFC Command Tag Register (MFC_Tag)

MFC_Size is the upper half of the memory-mapped I/O (MMIO) word and MFC_Tag is the lower half of the same word. This word is written using a single, 32-bit store instruction.

See the *Cell Broadband Engine Architecture* document for more information about this register.

Programming Note: The architecture allows for a future increase in the MFC_Tag register, along with the Prxy_QueryMask and Prxy_TagStatus registers, to 7 bits.

A.8.2 MFC Effective Address High Register (MFC_EAH)

The validity of this parameter is checked asynchronously to the instruction stream. If a segment fault, mapping fault, or protection violation occurs, an MFC data segment exception is generated. If the address is not aligned, an MFC DMA alignment exception is generated.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8.3 MFC Effective Address Low Register (MFC_EAL)

The validity of this parameter is checked asynchronously to the instruction stream. If a segment fault, mapping fault, or protection violation occurs, MFC data segment exception is generated. If the address is not aligned, MFC DMA alignment exception is generated.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8.4 MFC Transfer Size Register (MFC_Size)

MFC_Size is the upper half of the MMIO word and MFC_Tag is the lower half of the same word. This word is written using a single, 32-bit store instruction.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8.5 Proxy Tag-Group Query Mask Register (Prxy_QueryMask)

The Proxy Tag-Group Query Mask Register selects the tag groups to be included in the query operation.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8.6 Proxy Tag-Group Query Type Register (Prxy_QueryType)

The Proxy Tag-Group Query Type Register is used by software to request that the MFC detect a tag-group completion condition.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8.7 Proxy Tag-Group Status Register (Prxy_TagStatus)

The Proxy Tag-Group Status Register contains the current status of the tag groups enabled in the Proxy Tag-Group Query-Mask Register.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.8.8 SPU Signal Notification Register 1 (SPU_Sig_Notify_1)

See the *Cell Broadband Engine Architecture* document for more information.

A.8.9 SPU Signal Notification Register 2 (SPU_Sig_Notify_2)

See the *Cell Broadband Engine Architecture* document for more information.

A.9 SPU Control Registers

A.9.1 SPU Outbound Mailbox Register (SPU_Out_Mbox)

Other processors or devices read the mailbox data from the SPU in the SPU_Out_Mbox Register. The SPU sends data to this mailbox by writing to the SPU_WrOutMbox channel (see the *SPU Write Outbound Mailbox Channel (SPU_WrOutMbox)* in the *Cell Broadband Engine Architecture* for more information). When the SPU writes to the SPU_WrOutMbox channel, the channel counter decrements from 1 to 0. Reading this register causes the SPU_WrOutMbox channel counter to increment to 1. While the channel count is 1, further reads to this register do not affect the count, and the read data is the last stored value in this register. The queue depth for this implementation is 1.

Note: For the SPU Write Outbound Mailbox channel (SPU_WrOutMbox), the count is 1 at POR, and software reinitializes the count to 1.

The SPU Outbound Interrupt Mailbox Register has the same behavior as this register, except that SPU_OutIntrMbox has a Mailbox Interrupt that allows the SPU to notify the PPU when the SPU has written data to the SPU_Out_Mbox.

See the *Cell Broadband Engine Architecture* document for more information about this register.

A.9.2 SPU Inbound Mailbox Register (SPU_In_Mbox)

The PPE writes mailbox data to the SPU in the SPU_In_Mbox Register. This register corresponds to the SPU_RdInMbox channel. If this register is full, then additional writes overwrite the last entry written. The channel count remains at 4.

The SPU Inbound Mailbox Threshold interrupt is used to notify the PPE when the SPU has read all of the SPU_In_Mbox data. The SPU Inbound Mailbox Threshold interrupt is asserted when the SPU_RdInMbox channel count transitions from 1 to 0 (SPU_In_Mbox empty), and deasserted when the SPU_RdInMbox channel counter increments from 0 to 1. This interrupt is almost always asserted. The interrupt is taken when it is asserted and enabled.

See the *Class 2 Interrupt Mask Register (INT_Mask_class2)* for information about how to enable this interrupt. See the *Cell Broadband Engine Architecture* document for more information about the SPU Inbound Mailbox Threshold interrupt.

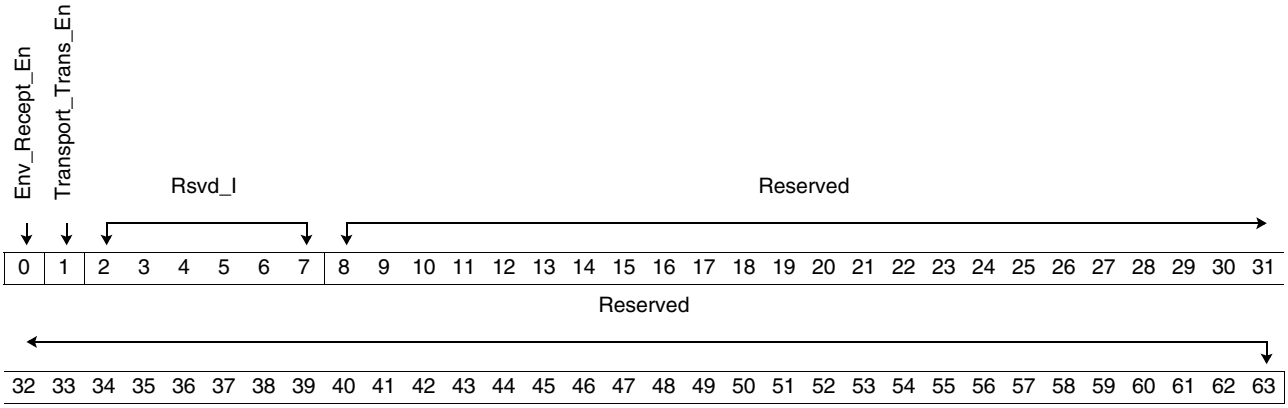
Programming Note: The SPU_NPC Register content is not valid for an illegal instruction.

A.10 BIC 0 and BIC 1 MMIO Registers (BClk Domain)

The BIC 0 (BClk) shared memory space starts at x‘512 000’ and ends at x‘512 3FF’. The BIC 1 (BClk) shared memory space starts at x‘513 000’ and ends at x‘513 3FF’.

A.10.1 Interface n Initialization Register (IFnINIT [n = 0,1])

Register Short Name	IF0INIT IF1INIT	**Privilege Type**	Privilege 1
Access Type	MMIO Read/Write	**Width**	64 bits
Hex Offset From BE_MMIO_Base	x‘512200’ x‘513200’	**Memory Map Area**	BIC 0 BClk BIC 1 BClk
Value at Initial POR	All bits set to zero	**Value During POR Set By**	Scan initialization during POR
Specification Type	Implementation-specific register	**Unit**	BIC



Bits	Field Name	Description
0	Env_Recept_En	Envelope Reception Enable Ignore data and valid from BED until this bit is ‘1’. 0 Envelope receive logic is disabled. 1 Envelope receive logic is enabled.
1	Transport_Trans_En	Transport Layer Transmission Enable Envelopes are not formed until this bit is ‘1’. 0 No envelopes are formed. 1 Envelopes are formed and passed to the link-send logic.
2:7	Rsvd_I	Reserved. Latch bits are implemented; value read is the value written.
8:63	Reserved	Bits are not implemented; all bits read back zero.

A.11 Additional Fault Isolation Registers

The Cell Broadband Engine (CBE) processor contains several fault isolation registers (FIRs) for collecting and reporting information about errors. FIR registers are organized in two categories: global FIRs and local FIRs. The lowest level of error-collection is in the local FIRs, that are implemented in various units on the chip. The errors are then ORed together and reported in the global FIRs, which are implemented in the test control unit within the CBE processor pervasive logic.

The following registers are part of a debug mode for the CBE processor and beyond the scope of this document.

- BIU Fault Isolation, Error Mask, and Checkstop Enable Registers
- IOC Fault Isolation, Error Mask, and Checkstop Enable Registers
  IOC_FIR includes error bits for the CBE interface unit and element interconnect bus (EIB). The IOC_FIR is unique and includes one additional register (IOC_SysErrEn), that is not part of other FIR registers on the chip. The IOC_SysErrEn register is used to generate an enable for the System Error interrupt.
- spec_att_mchk_fir Register
  The spec_att_mchk_fir Register is a status register that collects either the quiesce state from the PowerPC processor unit (PPU), or any machine check or system error interrupt conditions.
- fir_mode_reg Register
  The fir_mode_reg Register is a global RAS logic control register.
- fir_enable_mask Register
  The fir_enable_mask Register is a mask register for the global_fir, checkstop_fir, recoverable_fir, and spec_att_mchk_fir registers.

Glossary

AC0	Address concentrator 0.
ACK	See acknowledgment.
acknowledgment	A transmission control character that is sent as an affirmative response to a data transmission.
architecture	A detailed specification of requirements for a processor or computer system. It does not specify details of how the processor or computer system must be implemented; instead it provides a template for a family of compatible implementations.
ATO	Atomic unit. Part of an SPE's MFC. It is used to synchronize with other processor units.
BED	Cell Broadband Engine distribution bus.
BEI	Cell Broadband Engine Interface Unit. The BEI can be configured to attach to a BIF, or it can be configured to attach to an IOIF.
BHT	Branch history table.
BIC	Bus interface controller. Part of the Cell Broadband Engine interface (BEI) to I/O.
BIF	Cell Broadband Engine interface. The EIB's internal communication protocol. It supports coherent interconnection for to other Cell Broadband Engines and BIF-compliant I/O devices, such as memory subsystems, switches, and bridge chips. See IOIF.
big-endian	A byte-ordering method in memory where the address n of a word corresponds to the most significant byte. In an addressed memory word, the bytes are ordered (left to right) 0, 1, 2, 3, with 0 being the most significant byte. See little endian.
BIU	Bus interface unit. Part of the PPE's interface to the EIB.
cache	High-speed memory close to a processor. A cache usually contains recently-accessed data or instructions, but certain cache-control instructions can lock, evict, or otherwise modify the caching of data or instructions.
caching inhibited	A memory update policy in which the cache is bypassed, and the load or store is performed to or from main memory. A page of storage is considered caching inhibited when the "I" bit has a value of '1' in the page table. Data located in caching inhibited pages cannot be cached at any memory hierarchy that is not visible to all processors and devices in the system. Stores must update the memory hierarchy to a level that is visible to all processors and devices in the system.
castouts	Cache blocks that must be written to memory when a cache miss causes a cache block to be replaced.
CBE	Cell Broadband Engine.

CBEA	Cell Broadband Engine Architecture. The Cell Broadband Engine is one implementation of the Cell Broadband Engine Architecture.
CEC	Central electronics complex.
channel	Channels are unidirectional, function-specific registers or queues. They are the primary means of communication between an SPE's SPU and its MFC, which in turn mediates communication with the PPE, other SPEs, and other devices. These other devices use MMIO registers in the destination SPE to transfer information on the channel interface of that destination SPE. Specific channels have read or write properties, and blocking or nonblocking properties. Software on the SPU uses channel commands to enqueue DMA commands, query DMA and processor status, perform MFC synchronization, access auxiliary resources such as the decrementer (timer), and perform interprocessor communication through mailboxes and signal notification. See also memory channel.
CIU	Core interface unit.
CO	See castouts.
coherence	Refers to memory and cache coherence. The correct ordering of stores to a memory address, and the enforcement of any required cache write-backs during accesses to that memory address. Cache coherence is implemented by a hardware snoop (or inquire) method, which compares the memory addresses of a load request with all cached copies of the data at that address. If a cache contains a modified copy of the requested data, the modified data is written back to memory before the pending load request is serviced.
CRC	Cyclic redundancy check.
cresp	combined response.
CTR	Counter register.
DAR	Data address register.
data storage interrupt	An interrupt posted when a fault is encountered accessing storage or I/O space. A typical data storage interrupt is a page fault or protection violation.
dcbt	Data cache block touch x form instruction.
dcbtst	Data cache block touch store instruction.
decrementer	A register that counts down each time an event occurs. Each SPU contains dedicated 32-bit decrementers for scheduling or performance monitoring, by the program or by the SPU itself.
DERR	Data error.
direct-mapped cache	A cache in which each main memory address can appear in only one location within the cache, operating more quickly when the memory request is a cache hit.

DMA	Direct memory access. A technique for using a special-purpose controller to generate the source and destination addresses for a memory or I/O transfer.
DMA command	A type of MFC command that transfers or controls the transfer of a memory location containing data or instructions.
DMA queue	A set of two queues for holding DMA-transfer commands. The SPE's queue has 16 entries. The PPE's queue has four entries (two plus an additional two for L2 cache) for SPE-requested DMA commands, and eight entries for PPE-requested DMA commands.
DMAC	Direct memory access controller. A controller that performs DMA transfers.
double precision	The specification that causes a floating-point value to be stored (internally) in the long format (two computer words).
DP	See double-precision.
DR	Data relocate.
DSI	Data storage interrupt.
DSISR	Data storage interrupt status register.
dual-issue	Issuing two instructions at once, under certain conditions.
EA	Effective address.
ECC	See error correction code.
effective address	An address generated or used by a program to reference memory. A memory-management unit translates an effective address (EA) to a virtual address (VA), which it then translates to a real address (RA) that accesses real (physical) memory. The maximum size of the effective-address space is 2^{64} bytes.
EIB	Element interconnect bus. The on-chip coherent bus that handles communication between the PPE, SPEs, memory, and I/O devices (or a second Cell Broadband Engine). The EIB is organized as four unidirectional data rings (two clockwise and two counterclockwise).
eieio	Enforce in-order execution of I/O transaction.
ERAT	Effective-to-real address translation, or a buffer or table that contains such translations, or a table entry that contains such a translation.
error correction code	A code appended to a data block that can detect and correct multiple bit errors within the block.
ESID	Effective segment ID.
exception	An error, unusual condition, or external signal that may alter a status bit and will cause a corresponding interrupt, if the interrupt is enabled. See interrupt.

fence	An option for a barrier command that causes the processor to wait for completion of all MFC commands before starting any commands queued after the **fence** command. It does not apply to these immediate commands: **getllar**, **putllc** and **putlluc**.
fetch	Retrieving instructions from either the cache or main memory and placing them into the instruction queue.
FIFO	First in first out. Refers to one way elements in a queue are processed. It is analogous to “people standing in line.”
FIR	Fault isolation register.
FlexIO	Rambus processor bus interface.
floating point	A way of representing real numbers (that is, values with fractions or decimals) in 32 bits or 64 bits. Floating-point representation is useful to describe very small or very large numbers.
FP	Floating point.
general purpose register	An explicitly addressable register that can be used for a variety of purposes (for example, as an accumulator or an index register).
getllar	Get lock line and reserve command.
GPR	See general-purpose register.
GRF	Growable array file.
guarded	Prevented from responding to speculative loads and instruction fetches. The operating system typically implements guarding, for example, on all I/O devices.
HDEC	Hypervisor decrementer.
HID	Hardware-implementation dependent.
hypervisor	A control (or virtualization) layer between hardware and the operating system. It allocates resources, reserves resources, and protects resources among (for example) sets of SPEs that may be running under different operating systems. The Cell Broadband Engine has three operating modes: user, supervisor, and hypervisor. The hypervisor performs a meta-supervisor role that allows multiple independent supervisors’ software to run on the same hardware platform. For example, the hypervisor allows both a real-time operating system and a traditional operating system to run on a single PPE. The PPE can then operate a subset of the SPEs in the Cell Broadband Engine with the real-time operating system, while the other SPEs run under the traditional operating system.
IABR	Instruction address breakpoint register.
ICBI	Instruction cache block invalidate.
ICBIQ	**icbi** queue.

ID	Instruction dispatch.
IIC	Internal interrupt controller.
implementation	A particular processor that conforms to the architecture, but may differ from other architecture-compliant implementations for example in design, feature set, and implementation of optional features.
instruction cache	A cache for providing program instructions to the processor faster than they can be obtained from RAM.
INT	Interrupt.
interrupt	A change in machine state in response to an exception. See exception.
IOC	I/O controller.
IOIF	Cell Broadband Engine I/O Interface. The EIB's noncoherent protocol for interconnection to I/O devices. See BIF.
IR	Instruction relocate.
IS	Instruction issue.
IU	Instruction unit.
JTAG	Joint Test Action Group.
KB	1024 bytes of memory.
L1	Level 1 cache memory. The closest cache to a processor, measured in access time.
L2	Level 2 cache memory. The second-closest cache to a processor, measured in access time. An L2 cache is typically larger than an L1 cache.
ld	Load doubleword instruction.
least recently used	A policy for a caching algorithm that removes from the cache the item that has the longest elapsed time since its last access., An algorithm used to identify and make available the cache space that contains the data that was least recently used.
least significant bit	The bit of least value in an address, register, data element, or instruction encoding.
least significant byte	The byte of least value in an address, register, data element, or instruction encoding.
little-endian	A byte-ordering method in memory where the address *n* of a word corresponds to the least significant byte. In an addressed memory word, the bytes are ordered (left to right) 3, 2, 1, 0, with 3 being the most-significant byte. See big-endian.
livelock	An endless loop in program execution.
lmw	Load multiple word instruction.

local store	The 256-KB local store (LS) associated with each SPE. It holds both instructions and data.
logical partitioning	A function of an operating system that enables the creation of logical partitions.
LPAR	See logical partitioning.
LPID	Logical-partition identity.
LR	Link register.
LRU	See least recently used.
LS	See local store.
LSA	Local Store Address. An address in the LS of an SPU, by which programs running in the SPU and DMA transfers managed by the MFC access the LS.
LSb	See least significant bit.
LSB	See least significant byte.
mailbox	A queue in an SPE's MFC for exchanging 32-bit messages between the SPE and the PPE or other devices. Two mailboxes (the SPU Write Outbound Mailbox and SPU Write Outbound Interrupt Mailbox) are provided for sending messages from the SPE. One mailbox (the SPU Read Inbound Mailbox) is provided for sending messages to the SPE.
main storage	The effective-address (EA) space. It consists physically of real memory (whatever is external to the memory-interface controller, including both volatile and nonvolatile memory), SPU LSs, memory-mapped registers and arrays, memory-mapped I/O devices (all I/O is memory-mapped), and pages of virtual memory that reside on disk. It does not include caches or execution-unit register files. See local store.
mask	A pattern of bits used to accept or reject bit patterns in another set of data. Hardware interrupts are enabled and disabled by setting or clearing a string of bits, with each interrupt assigned a bit position in a mask register
MB	2^{20} bytes of memory.
MBL	MIC bus logic.
memory channel	An interface to external memory chips. The Cell Broadband Engine supports two Rambus extreme data rate (XDR) memory channels.
memory coherency	An aspect of caching in which it is ensured that an accurate view of memory is provided to all devices that share system memory.
memory-mapped	Mapped into the Cell Broadband Engine's addressable-memory space. Registers, SPE local stores (LSs), I/O devices, and other readable or writable storage can be memory-mapped. Privileged software does the mapping.

MERSI	Cache coherency protocol: Modified, Exclusive, Recent, Shared, Invalid, plus Mu (Unsolicited Modified), and Tagged (T).
MFC	Memory flow controller. It is part of an SPE and provides two main functions: moves data via DMA between the SPE's local store (LS) and main storage, and synchronizes the SPU with the rest of the processing units in the system.
mfceieio	MFC enforce in-order execution of I/O command.
mfcsync	MFC synchronize command.
mfspr	Move from special-purpose register instruction.
MIC	Memory interface controller. The Cell Broadband Engine's MIC supports two memory channels.
MMIO	Memory-mapped input/output. See memory-mapped.
MMU	Memory management unit. A functional unit that translates between effective addresses (EAs) used by programs and real addresses (RAs) used by physical memory. The MMU also provides protection mechanisms and other functions.
most significant bit	The highest-order bit in an address, registers, data element, or instruction encoding.
most significant byte	The highest-order byte in an address, registers, data element, or instruction encoding.
MSb	See most significant bit.
MSB	See most significant byte.
MSR	Machine state register.
MT	See multithreading.
mtspr	Move to special-purpose register instruction.
multithreading	Simultaneous execution of more than one program thread. It is implemented by sharing one software process and set of execution resources but duplicating the architectural state (registers, program counter, flags, and so forth) of each thread.
NCU	Noncacheable unit.
no-op	No-operation. A single-cycle operation that does not affect registers or generate bus activity.
page	A region in memory. The PowerPC Architecture defines a page as a 4 KB area of memory, aligned on a 4 KB boundary or a large page size which is implementation dependent.
page fault	A page fault is a condition that occurs when the processor attempts to access a memory location that resides within a page not currently resident in physical memory.

page table	A table that maps virtual addresses (VAs) to real addresses (RAs) and contains related protection parameters and other information about memory locations.
PLG	Physical layer group.
POR	Power-on reset
PowerPC	Of or relating to the PowerPC Architecture or the microprocessors that implement this architecture.
PowerPC Architecture	A computer architecture that is based on the third generation of RISC processors. The PowerPC architecture was developed jointly by Apple, Motorola, and IBM.
PPE	PowerPC Processor Element. The general-purpose processor in the Cell Broadband Engine. Consists of the PPU and the PPSS.
PPSS	PowerPC Processor Storage Subsystem. Part of the PPE. It operates at half the frequency of the PPU and includes an L2 cache and Bus Interface Unit (BIU).
PPU	PowerPC Processor Unit. The part of the PPE that includes execution units, memory-management unit, and L1 cache.
privileged mode	Also known as supervisor mode. The permission level of operating system instructions. The instructions are described in *PowerPC Architecture, Book III* and are required of software the accesses system-critical resources.
problem state	The permission level of user instructions. The instructions are described in *PowerPC Architecture, Books I and II* and are required of software that implements application programs.
PTE	Page table entry. See page table.
PTEG	Page table entry group.
putllc	Put lock line conditional on a reservation command.
putlluc	Put lock line unconditional command.
PVR	Processor version register.
quadword	A group of 16 contiguous locations starting at an address divisible by 16.
RA	Real address.
RAG	Resource allocation group.
RC	Read-and-claim state machine.
real address	An address for physical storage, which includes physical memory, the PPE's L1 and L2 caches, and the SPE's local stores (LSs) if the operating system has mapped the LSs to the real-address space. The maximum size of the real-address space is 2^{42} bytes.
RESN	Returned envelope sequence number. In the BIC. Upon successful reception of an envelope, a positive acknowledgment is generated back to the transmitting chip.

RLM	Random logic macro.
RMT	Replacement management table.
RPN	Real page number.
RRAC	Redwood Rambus Access Cell (RRAC) physical link (PHY). This is an early term for FlexIO, now obsolete.
signal	Information sent on a signal-notification channel. These channels are inbound (to an SPE) registers. They can be used by the PPE or other processor to send information to an SPE. Each SPE has two 32-bit signal-notification registers, each of which has a corresponding memory-mapped I/O (MMIO) register into which the signal-notification data is written by the sending processor. Unlike mailboxes, they can be configured for either one-to-one or many-to-one signalling. These signals are unrelated to UNIX signals. See channel and mailbox.
SIMD	Single instruction, multiple data. Processing in which a single instruction operates on multiple data elements that make up a vector data-type. Also known as vector processing. This style of programming implements data-level parallelism.
SL1	A first-level cache for DMA transfers between local storage and main storage.
SLB	Segment lookaside buffer. It is used to map an effective address (EA) to a virtual address (VA).
SMM	Synergistic memory management unit. It translates EAs to RAs in an SPU.
sndsig	Send signal command.
snoop	To compare an address on a bus with a tag in a cache, in order to detect operations that violate memory coherency.
SPE	Synergistic processor element. Consists of a synergistic processor unit (SPU), a memory flow controller (MFC), and local store (LS).
SPR	Special-purpose register.
SPU	Synergistic processor unit. The part of an SPE that executes instructions from its local store (LS).
supervisor mode	The privileged operation state of a processor. In supervisor mode, software, typically the operating system, can access all control registers and can access the supervisor memory space, among other privileged operations.
sync	Synchronize command.
synchronization	The order in which storage accesses are performed.
TAG	MFC command tag.

tag group	A group of DMA commands. Each DMA command is tagged with a 5-bit tag group identifier. Software can use this identifier to check or wait on the completion of all queued commands in one or more tag groups. All DMA commands except **getllar**, **putllc**, and **putlluc** are associated with a tag group.
TCU	Pervasive unit, used for test control logic.
thread	A sequence of instructions executed within the global context (shared memory space and other global resources) of a process that has created (spawned) the thread. Multiple threads (including multiple instances of the same sequence of instructions) can run simultaneously, if each thread has its own architectural state (registers, program counter, flags, and other program-visible state). Each SPE can support only a single thread at any one time. The multiple SPEs can simultaneously support multiple threads. The PPE supports two threads at any one time, without the need for software to create the threads. The PPE does this by duplicating architectural state.
time base	Chip-level time base, as defined in the PowerPC Architecture.
TKM	Token management unit. Part of the element interconnect bus (EIB) that software can program to regulate the rate at which particular devices are allowed to make EIB command requests.
TLB	Translation lookaside buffer. An on-chip cache that translates virtual addresses (VAs) to real addresses (RAs). A TLB caches page-table entries for the most recently accessed pages, thereby eliminating the necessity to access the page table from memory during load/store operations.
tlbie	Translation lookaside buffer invalidate entry instruction.
TS	The transfer-size parameter in an MFC command.
VA	See virtual address.
vector	An instruction operand containing a set of data elements packed into a one-dimensional array. The elements can be fixed-point or floating-point values. Most Vector/SIMD Multimedia Extension and SPU SIMD instructions operate on vector operands. Vectors are also called SIMD operands or packed operands.
Vector/SIMD Multimedia Extension	The SIMD instruction set of the PowerPC Architecture, supported on the PPE.
virtual address	An address to the virtual-memory space, which is much larger than the physical address space and includes pages stored on disk. It is translated from an effective address (EA) by a segmentation mechanism and used by the paging mechanism to obtain the real address (RA). The maximum size of the virtual-address space is 2^{65} bytes.
VPN	Virtual page number. The number of the page in virtual memory.
VSID	Virtual segment ID.
VSU	Vector/scalar unit.

VXU	Vector/SIMD multimedia extension unit.
word	Four bytes.
XDR	Rambus external data representation (XDR) DRAM memory technology.
XIO	A Rambus extreme data rate I/O memory channel.
XU	Execution unit.