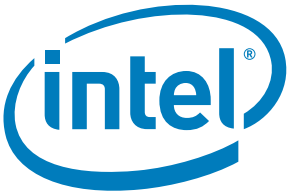


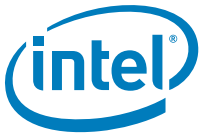
3rd Gen Intel® Xeon® Processor Scalable Family, Codename Ice Lake, Uncore Performance Monitoring

Reference Manual

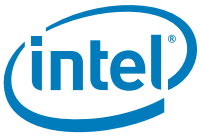
Revision 1.00

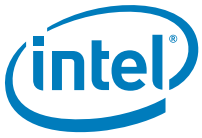
May 2021

Reference Number: 639778



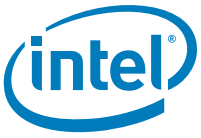
Contents

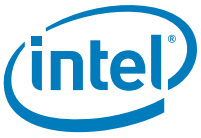


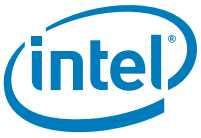


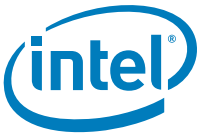
Figures

Tables









Revision History

Document Number	Revision Number	Description	Date
639778	1.0	• Initial Release	May 2021

§

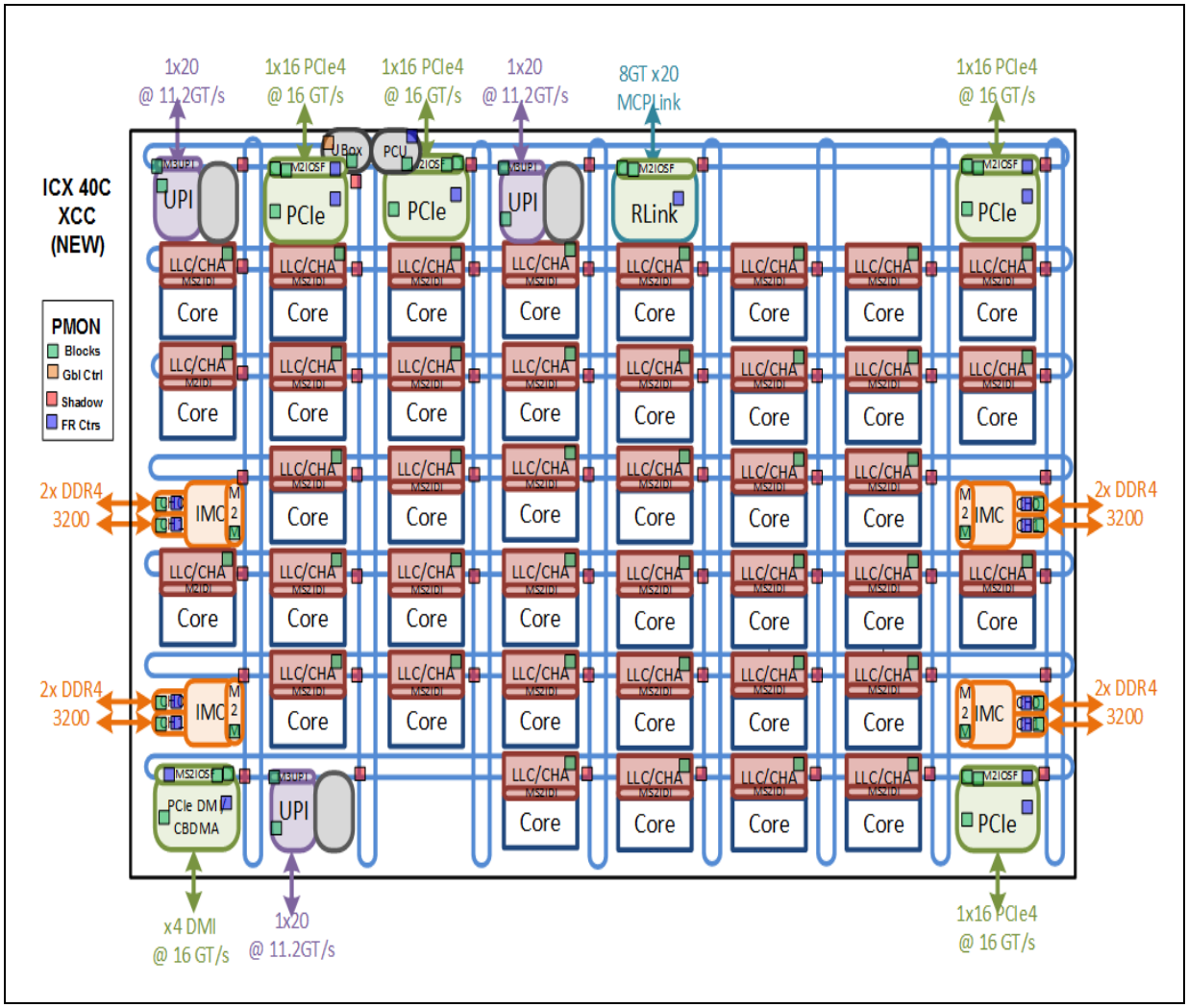
1 Introduction

1.1 Introduction

'Uncore' roughly equates to logic outside the CPU cores but residing on the same die. Traffic (for example, Data Reads) generated by threads executing on CPU cores or IO devices may be operated on by logic in the Uncore. Logic responsible for managing coherency, managing access to the DIMMs, managing power distribution and sleep states, and so forth.

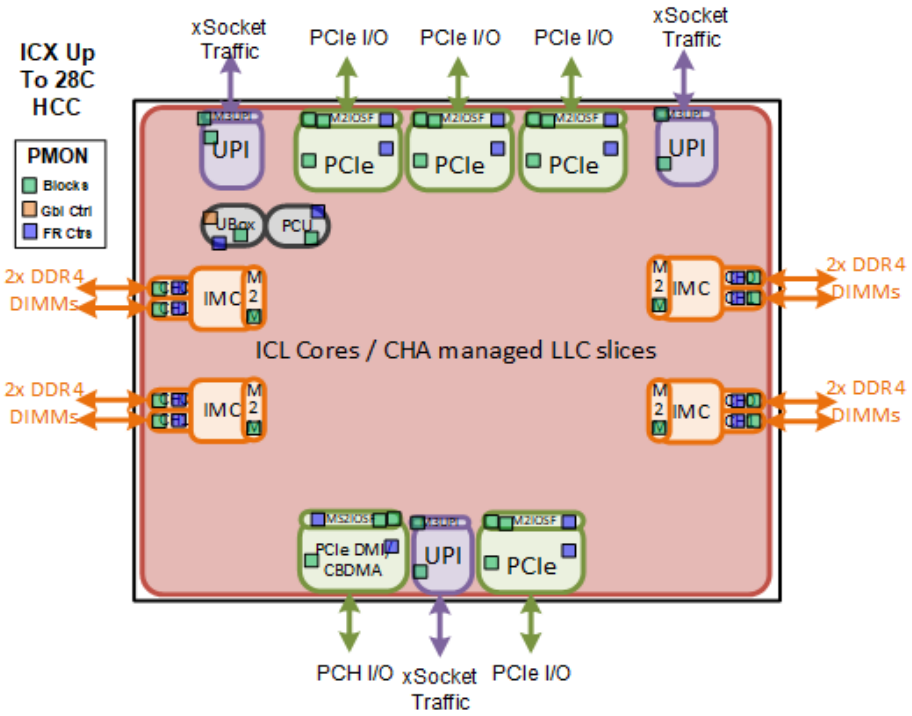
The uncore sub-system of the next generation 3rd Gen Intel® Xeon® Processor Scalable Family Server is shown in the following figure. The uncore sub-system consists of a variety of components, many assigned to the aforementioned responsibilities, ranging from the CHA cache/home agent to the power controller unit (PCU) and integrated memory controller (IMC), to name a few. Most of these components provide similar performance monitoring capabilities.

Figure 1-1. 3rd Gen Intel Xeon Processor Scalable Family Server-40C XCC Block Diagram



Note: This diagram represents one possible configuration. The number of supported cores and Intel® Ultra Path Interconnect (Intel® UPI) links vary by SKU. Not all features supported on all SKUs.

Figure 1-2. 3rd Gen Intel Xeon Processor Scalable Family HCC Block Diagram



Note: This diagram represents one possible configuration. The number of supported cores and Intel UPI links vary by SKU. Not all features supported on all SKUs.

Before going in to the details of 3rd Gen Intel Xeon Processor Scalable Family Server's Uncore PMON, the following sections will provide

- A general overview of Uncore PMON operation and the state provided SW to manage its operation.
- Functionality common to individual units with the common logic to support the functionality.
- A summary of 3rd Gen Intel Xeon Processor Scalable Family Server's uncore performance monitoring capabilities.
- Addressing all 3rd Gen Intel Xeon Processor Scalable Family Server uncore performance monitoring state.
- Introduction to new Discovery mechanism.
- Some guidance to SW including how to manage a monitoring session, find the base address to the page of Discovery and find the base addresses for PMON registers addressed in PCICFG or MMIO space.

1.2 Section References

The following sections provide a breakdown of the performance monitoring capabilities for each box.

- Section 2.1, “UBox Performance Monitoring”
- Section 2.2, “Mesh Performance Monitoring”
- Section 2.3, “Caching/Home Agent (CHA) Performance Monitoring”
- Section 2.4, “Memory Controller (IMC) Performance Monitoring”
- Section 2.5, “IIO Performance Monitoring”
- Section 2.6, “IRP Performance Monitoring”
- Section 2.7, “Intel® UPI Link Layer Performance Monitoring”
- Section 2.8, “M2M Performance Monitoring”
- Section 2.9, “M2PCIe Performance Monitoring”
- Section 2.10, “M3UPI Performance Monitoring”
- Section 2.11, “PCIe3/DMI Performance Monitoring”
- Section 2.12, “Power Control (PCU) Performance Monitoring”
- Section 3.1, “Packet Matching Reference(s)”

1.3 Uncore PMON Overview

1.3.1 A Simple Hierarchy

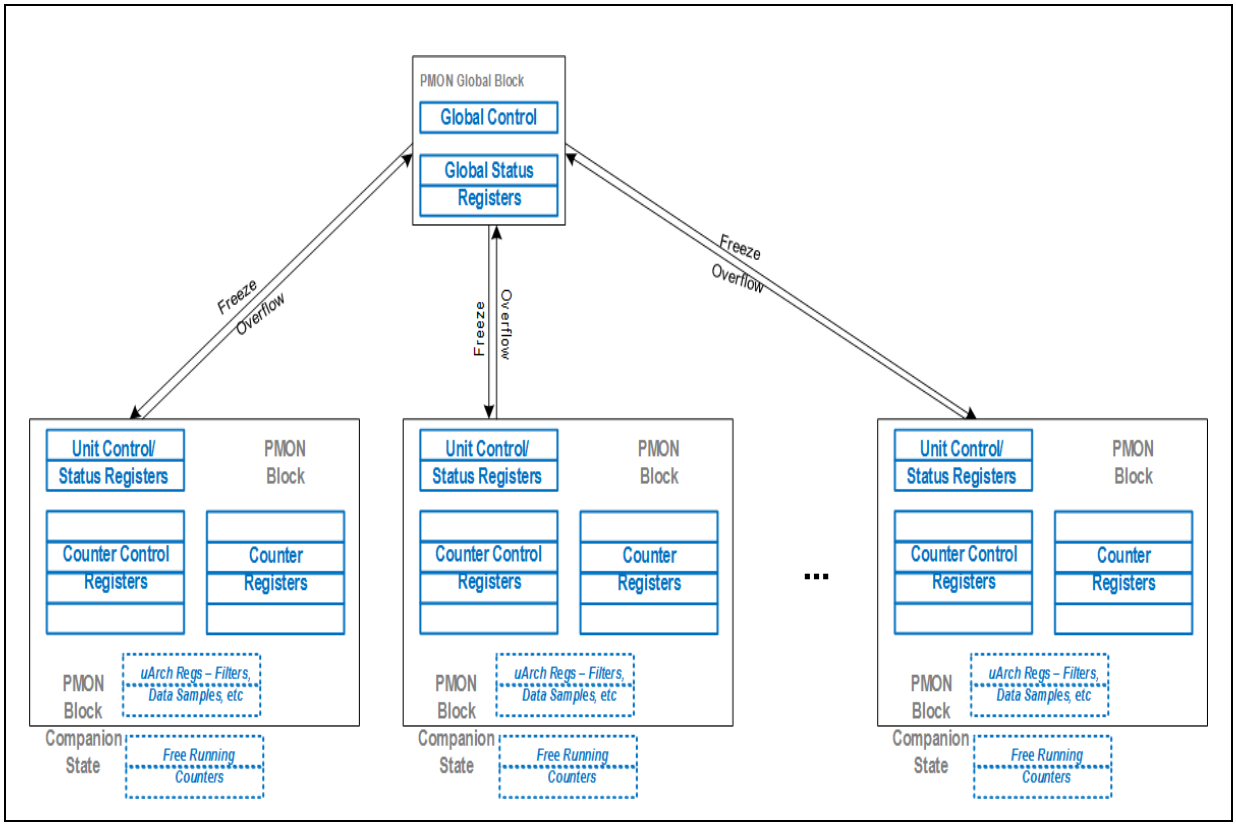
Uncore performance monitoring is managed through a very simple hierarchy. There are some number of performance monitoring (or ‘**PMON**’) units governed by a global control.

Each PMON block contains a set of counters with paired control registers. Each unit provides a set of events for SW to select from. SW can ask HW to collect an event by specifying what to count in a counter’s control register. SW can then periodically read the collected value from the paired counter.

Some units offer an expanded event set that require additional counter control bits. (For example, CHA, IIO and Intel UPI).

Some units offer the ability to further refine, or ‘filter’, the monitored events through additional counter control registers.

Figure 1-3. Uncore PMON Components and Hierarchy



Note: Uncore performance monitors represent a per-socket resource not meant to be affected by context switches and thread migration performed by the OS. It is recommended that the monitoring software agent establish a fixed affinity binding to prevent event count cross-talk across uncore PMON collected from different sockets.

To manage the large number of counter registers distributed across so many units and collect event data efficiently, each block has a modest amount of control/status governed by a similar global control/status.

SW can directly synchronize actions across counters (for example, to start/stop/reset counting) within each PMON block or across all PMON blocks through this control state.

SW can indirectly synchronize actions across counters (for example, stop counting) in all the PMON blocks by telling HW what to do when a counter overflows. SW can set a counter to overflow, after a set number of events have been captured, by pre-seeding the counter. For each counter, SW can then choose whether to notify the global PMON control that a counter has overflowed.

Upon receipt of an overflow, the global control will assert the global freeze signal. Once the global freeze has been detected, each box will disable (or 'freeze') all of its counters. In the process of generating a global freeze, SW can configure the global control to send a PMI signal to the core executing the monitoring software.

The following sections will detail the basic control state provided to SW to control performance monitoring in the uncore.

1.3.2 Global PMON State

1.3.2.1 Global PMON Global Control/Status Registers

The following registers represent state governing all PMUs in the uncore, both to exert global control and collect unit-level information.

U_MSR_PMON_GLOBAL_CTL contains bits that can stop (*.frz_all*) / restart (*.unfrz_all*) all the uncore counters.

Figure 1-4. PMON Global Control Register for 3rd Gen Intel Xeon Processor Scalable Family Server

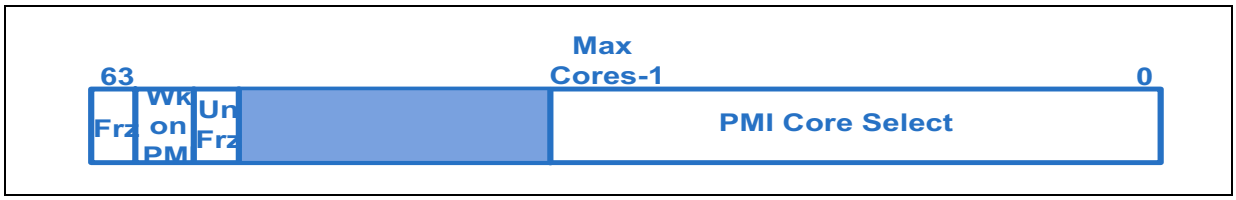


Table 1-1. U_MSR_PMON_GLOBAL_CTL Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
frz_all	63	WO	0	Freeze all uncore performance monitors.
wk_on_pmi	62	RW	0	If PMI event requested to send to core... 0 - Send event to cores already awakened 1 - Wake any sleeping core and send PMI to all cores.
unfrz_all	61	WO	0	Unfreeze all uncore performance monitors.
rsv	60:MaxCores	RV	0	Reserved
pmi_core_sel	MaxCores-1:0	RW	0	PMI Core Select Ex: If counter overflow is sent to UBox... 00000000000000000000000000 - No PMI sent 00000000000000000000000000001 - Send PMI to core 0 00000000000000000000001000000 - Send PMI to core 6 00000000000000000000001100010 - Send PMI to core 2, 5 & 6 and so forth. NOTE: If wk_on_pmi is set to 1, a wake will be sent to any sleeping core in the mask prior to sending the PMI.

If an overflow is detected in any of the uncore's PMON registers, it will be summarized in one or more U_MSR_PMON_GLOBAL_STATUS registers. These registers accumulate overflows sent to it from uncore boxes with PMON blocks. To reset these overflow bits, a user must set the corresponding bits in U_MSR_PMON_GLOBAL_STATUS to 1, which will act to clear them.

Figure 1-5. PMON Global Status Register for 3rd Gen Intel Xeon Processor Scalable Family Server

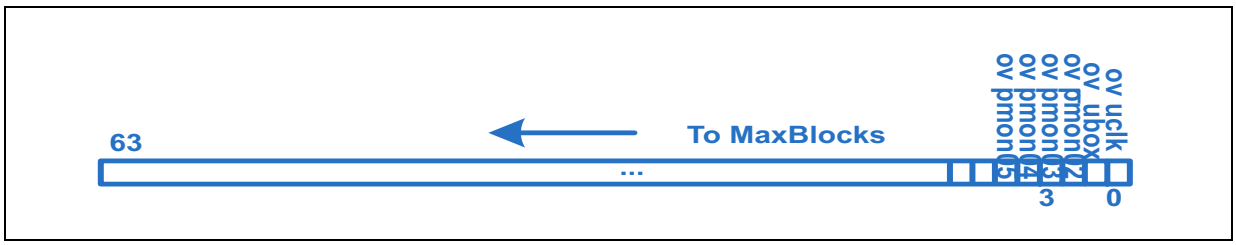


Table 1-2. U_MSR_PMON_GLOBAL_STATUS Register – Field Definitions

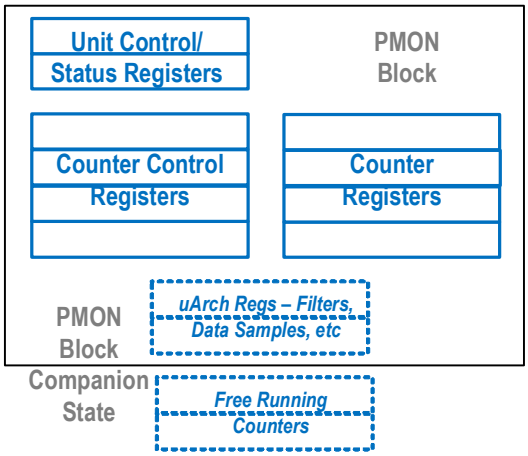
Field	Bits	Attr	HW Reset Val	Description
rsv	63:MaxBlocks	RV	0	Reserved
ov_pmonX	MaxBlocks-1:4	RW1C	0	Overflow detected in PMON register from Block with "Global Status Position" of "MaxBlocks-1" as reported through Global Discovery.
ov_pmonx-1:ov_pmon04	MaxBlocks-2:4	RW1C	0	Overflow detected in PMON register(s) from Blocks with a Global Status Position between MaxBlocks-1 and 3
ov_pmon03	3	RW1C	0	Overflow detected in PMON register from Block with "Global Status Position" of 3
ov_pmon02	2	RW1C	0	Overflow detected in PMON register from Block with "Global Status Position" of 2
ov_u	1	RW1C	0	Overflow detected in UBox PMON register
ov_uclk	0	RW1C	0	Overflow detected in UBox fixed UCLK register

As of 3rd Gen Intel® Xeon® Processor Scalable Family, the mapping of Global Status bits in the Global Status register(s) to PMON blocks will be provided through the new PMON Discovery mechanism. The first two status bits correspond to the UCLK fixed register and the UBlock respectively. The rest of the status bits correspond to overflows detected from PMON Block's IDed through Discovery. Discovery for each PMON block will report its "Global Status Position" (i.e. which bit in the global status register records its overflows).

For instance, SW may discover a PMON block of Unit Type = CHA, Unit ID 5 has a Global Status Position of 5.

1.4 Unit Level PMON State

Each PMON block in the uncore is composed of the following state:



- A Unit Control register to aid software sample collection.
- Status registers to record when a counter within the Block overflows.
- A set of data registers
- A set of control registers, each paired to a data register, to allow SW to specify what event should be captured.
- Additional micro-architectural specific state designed to enhance performance monitoring collection within a block. For example, event or traffic Filters.
- Some free running counters, although not subject to the PMON hierarchy, may be included in this document with the Unit they are associated with.

Every PMON block in the system is governed by a modest amount of Unit level Control. Each bit intended to assist SW in more efficiently managing the PMON state within the block. Reset bits help reduce the time SW needs to setup a new sample.

Note: If the PMON registers within the unit are shared among different users, either those users should leave this register untouched or they should agree on the user allowed to affect the Unit level control state.

Figure 1-6. PMON Unit Control Register for 3rd Gen Intel Xeon Processor Scalable Family Server - Common to all PMON Blocks

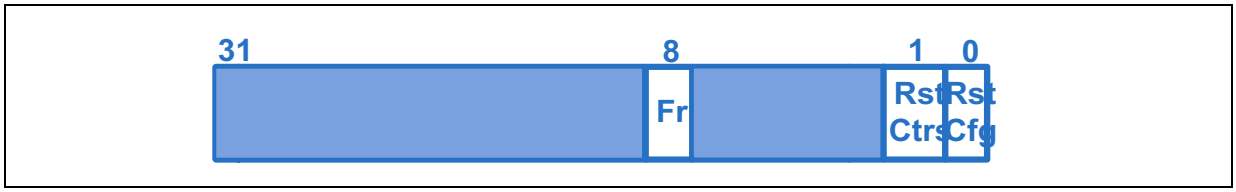


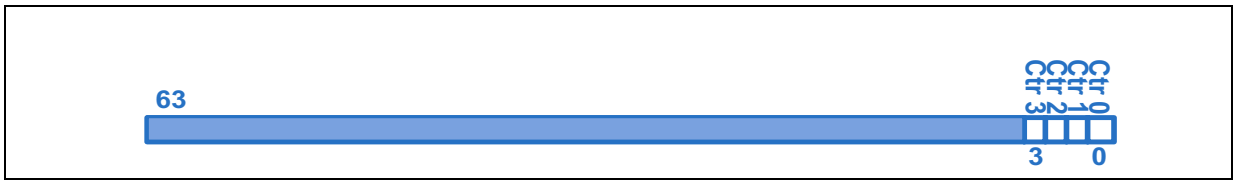
Table 1-3. PMON_UNIT_CTL Register – Field Definitions (Sheet 1 of 2)

Field	Bits	Attr	HW Reset Val	Description
rsv	31:18	RV	0	Reserved
rsv	17:16	RV	0	Reserved; SW must write to 1 else behavior is undefined.
rsv	15:9	RV	0	Reserved

Table 1-3. PMON_UNIT_CTL Register – Field Definitions (Sheet 2 of 2)

Field	Bits	Attr	HW Reset Val	Description
frz	8	WO	0	Freeze. If set to 1 the counters in this box will be frozen.
rsv	7:2	RV	0	Reserved
rst_ctrs	1	WO	0	Reset Counters. When set to 1, the Counter Registers will be reset to 0.
rst_ctrl	0	WO	0	Reset Control. When set to 1, the Counter Control Registers will be reset to 0.

Figure 1-7. PMON Unit Status Register for 3rd Gen Intel Xeon Processor Scalable Family Server - Format common to all PMON Blocks



If an overflow is detected from one of the Unit's PMON registers, the corresponding bit in the PMON_UNIT_STATUS.ov field will be set. To reset these overflow bits, a user must write a value of '1' to them (which will clear the bits). There are typically four counters per PMON block. But that number may vary. As of 3rd Gen Intel Xeon Processor Scalable Family, the number of paired counter/counter control registers is reported through the Unit Discovery associated with each PMON block. The Unit Status register will contain "NumControlRegs" valid bits.

Note: (Can also check Table 1-7, "Per-Box Performance Monitoring Capabilities" or the section detailing each unit's functionality for the number counters it supports).

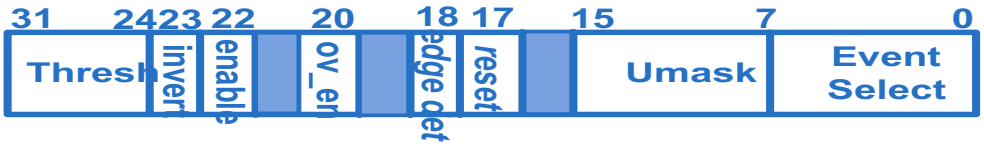
Table 1-4. PMON_UNIT_STATUS Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
rsv	31:4	RV	0	Reserved
ov	NumControl Regs-1:0	RW1C	0	If an overflow is detected from the corresponding PMON_CTR register, it's overflow bit will be set. NOTE: Write of '1' will clear the bit. Although 4 is very common, the number of overflow bits can vary by PMON block. The number can be discovered in the NumControlRegs field of the Unit's discovery.

1.4.1 Unit PMON state - Counter/Control Pairs

The following table defines the layout for the standard performance monitor control registers. Their main task is to select the event to be monitored by their respective data counter (*.ev_sel*, *.umask*). Additional control bits are provided to shape the incoming events (for example, *.invert*, *.edge_det*, *.thresh*) as well as provide additional functionality for monitoring software (*.rst,.ov_en*).

Figure 1-8. PMON Counter Control Register for 3rd Gen Intel® Xeon® Processor Scalable Family Server - Fields common to all PMON Blocks



Note: Per Unit considerations - please refer to each unit section for more detail on:

- Certain units may make use of additional bits in these counter control registers.
- The width of the Thresh field is dependent on a unit's 'widest' event (that is, the event that can increment the most per cycle, typically measuring per-cycle occupancy of a large queue).
- Several unit counter control registers are still 32b, some 64b. All are addressable as 64b registers.

An overview of the Counter Control logic is in the next section.

Table 1-5. Baseline *_PMON_CTLx Register – Field Definitions (Sheet 1 of 2)

Field	Bits	Attr	HW Reset Val	Description
rsv	63:32	RV	0	Reserved - Only relevant to unit's that use 64b control registers
thresh	31:24	RW	0	Threshold is used, along with the invert bit, to compare against the counter's incoming increment value. i.e. the value that will be added to the counter. For events that increment by more than 1 per cycle, if the threshold is set to a value greater than 1, the data register will accumulate instances in which the event increment is >= threshold. For example. say you have an event to accumulate the occupancy of a 64-entry queue every cycle. By setting the threshold value to 60, the data register would count the number of cycles the queue's occupancy was >= 60.
invert	23	RW	0	Invert comparison against Threshold. 0 - comparison will be 'is event increment >= threshold?'. 1 - comparison is inverted - 'is event increment < threshold?' For example, for a 64-entry queue, if SW wanted to know how many cycles the queue had fewer than 4 entries, SW should set the threshold to 4 and set the invert bit to 1. ***Note:*** .invert is in series following .thresh, Due to this, the.thresh field must be set to a non-0 value. For events that increment by no more than 1 per cycle, set .thresh to 0x1. Also, if .edge_det is set to 1, the counter will increment when a 1 to 0 transition (i.e. falling edge) is detected.
en	22	RW	0	Local Counter Enable
rsv	21	RV	0	Reserved. SW must write to 0 else behavior is undefined.

Table 1-5. Baseline *_PMON_CTLx Register – Field Definitions (Sheet 2 of 2)

Field	Bits	Attr	HW Reset Val	Description
ov_en	20	RW	0	When this bit is set to 1 and the corresponding counter overflows, an overflow message is sent to the UBox's global logic. The message identifies the unit that sent it. Once received, the global status register will record the overflow in the corresponding U_MSR_PMON_GLOBAL_STATUS bit.
rsv	19	RV	0	Reserved
edge_det	18	RW	0	When set to 1, rather than measuring the event in each cycle it is active, the corresponding counter will increment when a 0 to 1 transition (i.e. rising edge) is detected. When 0, the counter will increment in each cycle that the event is asserted. ***Note:*** .edge_det is in series following .thresh, due to this, the .thresh field must be set to a non-0 value. For events that increment by no more than 1 per cycle, set .thresh to 0x1.
rst	17	WO	0	When set to 1, the corresponding counter will be cleared to 0.
rsv	16	RV	0	Reserved. SW must write to 0 else behavior is undefined.
umask	15:8	RW	0	Select subevents to be counted within the selected event.
ev_sel	7:0	RW	0	Select event to be counted.

The default width for performance monitor data registers are 48b wide. A counter overflow occurs when a carry out from bit 47 is detected. Software can force all uncore counting to freeze after N events by preloading a monitor with a count value of 2^{48} - N and setting the control register to send an overflow message to the UBox (refer to Section 1.3.2, "Global PMON State"). During the interval of time between overflow and global disable, the counter value will wrap and continue to collect events.

To ensure accuracy, SW should stop the counter and check the overflow status before reading its value. But, if accessible, software can continuously read the data registers without disabling event collection.

Figure 1-9. PMON Counter Register for 3rd Gen Intel® Xeon® Processor Scalable Family Server - Common to all PMON Blocks

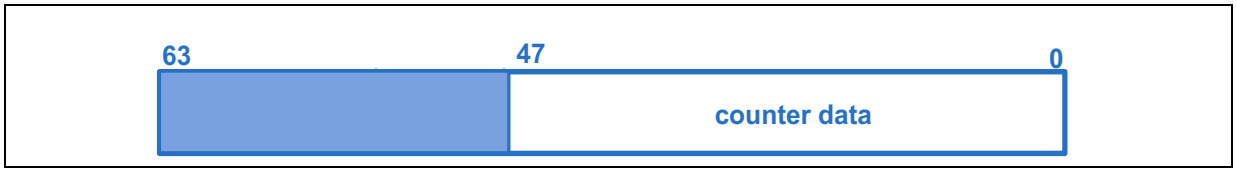


Table 1-6. Baseline *_PMON_CTRx Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
rsv	63:48	RV	0	Reserved
event_count	47:0	RW-V	0	48-bit performance event counter

1.4.2 Unit PMON Registers - On Overflow and the Consequences (PMI/Freeze)

If an overflow is detected from a Unit's performance counter, the overflow bit is set at the unit level (*_PMON_UNIT_STATUS.ov).

If the counter is enabled to communicate the overflow (*_PMON_CTL.ov_en is set to 1), an overflow message is sent to the UBox. When the UBox receives the overflow signal, the *_PMON_GLOBAL_STATUS.ov_x bit is set, a global freeze signal is sent and a PMI can be generated. 'x' represents the box generating the overflow (see Table 1-3, "U_MSR_PMON_GLOBAL_STATUS Register – Field Definitions").

Once a freeze has occurred, in order to see a new freeze, the overflow responsible for the freeze must be cleared by setting the corresponding bit in *_PMON_UNIT_STATUS.ov and U_MSR_PMON_GLOBAL_STATUs.ov_x to 1 (which acts to clear the bits).

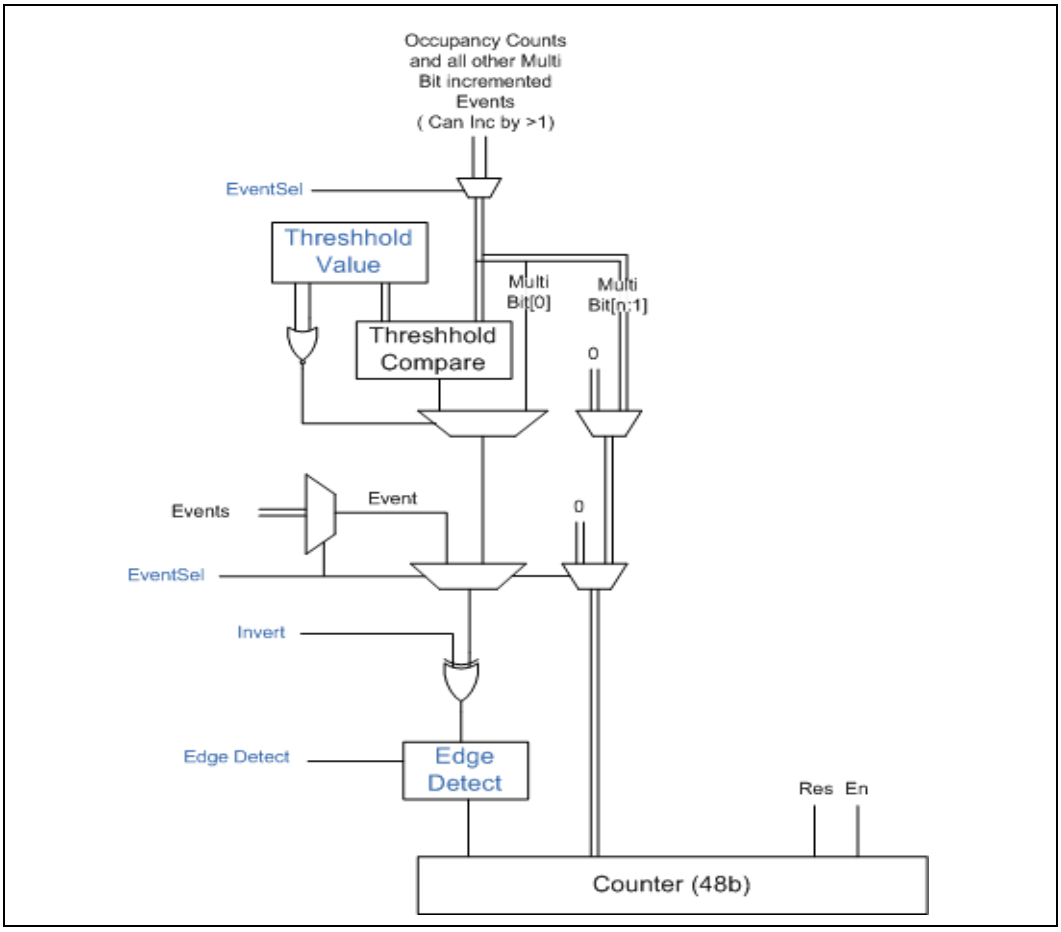
Assuming all counters have been locally enabled (.*en* bit is set to 1 in every control register meant to monitor events) and the overflow bits have been cleared, the Unit is prepared for a new sample interval. Once the global controls have been re-enabled (Section 1.9.7, "Enabling a New Sample Interval from Frozen Counters"), counting will resume.

1.5 Uncore PMON - Typical Counter Control Logic

Following is a logic diagram for the standard perfmon counter control. It illustrates how event information is routed, selected, filtered (by other bits in the control register) and sent to the paired data register for storage.

Note: The PCU uses an adaptation of this block (refer to Section 2.12, "Power Control (PCU) Performance Monitoring" more information). Also note that only a subset of the available control bits are presented in the diagram.

Figure 1-10. Perfmon Counter Control Block Diagram



Selecting What To Monitor: The main task of a configuration register is to select the event to be monitored by its respective data counter. Setting the *.ev_sel* and *.umask* fields performs the event selection.

Note: Only the *.ev_sel* is pictured in the previous figure. The *.umask* field is generally used to select subevents of the event. Once the proper subevent combination has been selected, it is passed on to the per Counter EventSel Mux.

Additional control bits used to filter and create information related to the selected Event:

Applying a Threshold to Incoming Events: *.thresh* - since most counters can increment by a value greater than 1, a threshold can be applied to generate an event based on the outcome of the comparison. If *.thresh* is set to a non-zero value, that value is compared against the incoming count for that event in each cycle. If the incoming count is >= the threshold value, then the event count captured in the data register will be incremented by 1.

Using the threshold field to generate additional events can be particularly useful when applied to a queue occupancy count. For example, if a queue is known to contain eight entries, it may be useful to know how often it contains 6 or more entries (that is, Almost Full) or when it contains 1 or more entries (i.e. Not Empty).

Note: For 3rd Gen Intel® Xeon® Processor Scalable Family the *.invert* and *.edge_det* bits follow the threshold comparison in sequence. If a user wishes to apply these bits to events that only increment by 1 per cycle, *thresh* must be set to 0x1.

Inverting the Threshold Comparison: *.invert* - Changes *.thresh* test condition to '<'.

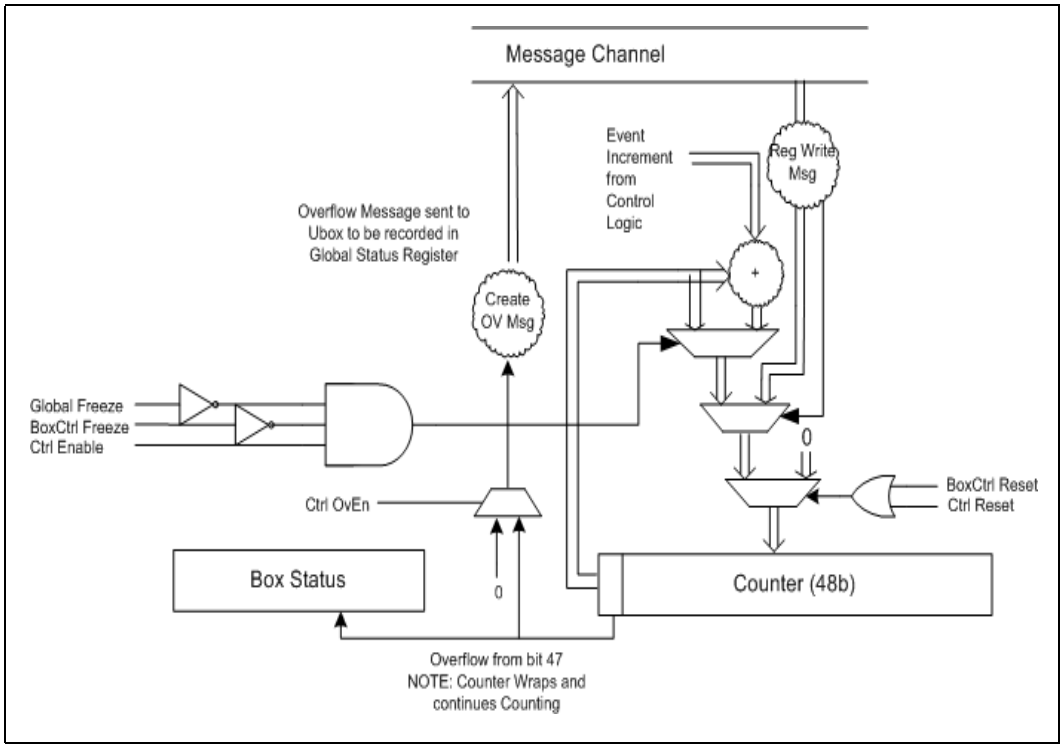
Counting State Transitions Instead of per-Cycle Events: *.edge_det* - Rather than accumulating the raw count each cycle (for events that can increment by 1 per cycle), the register can capture transitions from no event to an event incoming (that is, the 'Rising Edge').

1.6 Uncore PMON - Typical Counter Logic

Following is a diagram of a standard perfmon counter. It illustrates the control involved in managing the Data Register. Functionality includes how to start/stop the register, reset it, indicate an overflow and capture the information sent from the Counter Control block. The diagram contains bits from all levels in the Counter Control Hierarchy - Global, Unit level as well as from the paired Counter Control register.

Details on how to perform counter management, including how to set up a Monitoring Session and periodically sample the counters can be found in Section 1.9.5, "Setting up a Monitoring Session."

Figure 1-11. Perfmon Counter Block Diagram



Telling HW that the Control Register Is Set: *.en* bit must be set to 1 to enable counting. Once counting has been enabled at all levels of the Performance Monitoring Hierarchy (refer to Section 1.9.5, "Setting up a Monitoring Session" for more information), the paired data register will begin to collect events.

Notification after X events: *.ov_en* - Instead of manually stopping the counters at intervals (often wall clock time) pre-determined by software, hardware can be set to notify monitoring software when a set number of events has occurred. The Overflow Enable bit is provided for just that purpose. See Section 1.4.2, "Unit PMON Registers - On Overflow and the Consequences (PMI/Freeze)" for more information on how to use this mechanism.

1.7 3rd Gen Intel Xeon Processor Scalable Family Server's Uncore PMON

The general performance monitoring capabilities of each box are outlined in the following table.

Table 1-7. Per-Box Performance Monitoring Capabilities

Box	# Boxes	# Counters/ Box	# Queue Enabled	Packet Match/ Mask Filters?	Bit Width
CHA	up to 40	4	1	Y	48
IIO	up to 6 between C, P and M flavors	4 (+1) per stack (+4 per port)	0	N	48
IRP	up to 6	2	4	N	48
IMC	up to 4 (each with up to 2 channels)	4 (+6) (per channel)	4	N	48
Intel® UPI	up to 3 links	4 (per link)	4	Y	48
M3UPI	up to 3 links	3 (per link)	1	N	48
M2M	up to 2	4	1	Y	48
M2PCIe	up to 6	4	1	N	48
PCU	1	4 (+2)	4	N	48
UBox	1	2 (+1)	0	N	48

The programming interface of the counter registers and control registers fall into three address spaces:

- CHA, M2PCIe, IIO, IRP, PCU, and U-Box PMON registers are accessed through x86 RD/WRMSR instructions. See Table 1-9, "Uncore Performance Monitoring Registers (MSR)".
- iMC PMON registers are accessed through MMIO address space. M2M, Intel UPI, PCIe3 and M3UPI PMON registers are accessed through PCI device configuration space. See Table 1-10, "Free-Running IIO Bandwidth Out Counters in MSR space" for details.

Irrespective of the address-space difference and with only minor exceptions, the bit-granular layout of the control registers to program event code, unit mask, start/stop, and signal filtering via threshold/edge detect are the same.

1.7.1 Querying number of CHAs

The number of CHAs varies with the number of Cores in a system. To determine the number of CHAs, SW should read bits 31:0 in the CAPID6 register and bits 8:0 from CAPID7 located at Device 30, Function 3, Offset 0x9C and Offset 0xA0. These 40 bits form a bit vector of available LLC slices and the CHAs that manage those slices. For example: If bits 39:0 read 0x000F0F, the PMON blocks corresponding to CHAs 0-3 and 8-11 are available and CHAs 4-7 and 12-39 are not available.

1.7.2 Querying number of Intel UPI Links

The number of Intel UPI Links varies according to the specific version of the product. To determine the number of Intel UPI Links, SW should read bits 7:6 in the CAPID4 register located at Device 30, Function 3, Offset 0x94.

00 = 2 Intel UPI Links

01 = 2 Intel UPI Links

10 = 3 Intel UPI Links

11 = 3 Intel UPI Links

1.8 Addressing Uncore PMON State

Following is a list of registers provided in the 3rd Gen Intel® Xeon® Processor Scalable Family Server Uncore for Performance Monitoring. The registers are split between MSR space and PCICFG space.

1.8.1 Uncore Performance Monitoring State in MSR Space

First off, the Global Control / Status Registers

Table 1-8. Global Performance Monitoring Registers (MSR)

MSR Addresses	Description
0x700	Global Control
0x70E,0x70F	Global Status
0x703	UCLK Counter Control
0x704	UCLK Counter

As mentioned previously, PMON blocks in the Uncore have some number of paired counter/control (typically 4) registers, a unit status and unit control register (with the exception of the UBox). Many Units may offer extra PMON state such as event Filters or Fixed counters.

The addresses for all basic PMON state addressed through MSR space are laid out in the following table.

Table 1-9. Uncore Performance Monitoring Registers (MSR) (Sheet 1 of 2)

Unit	Unit Status	Unit Ctrl	Ctr3	Ctr2	Ctr1	Ctr0	Ctrl3	Ctrl2	Ctrl1	Ctrl0	Extra
CHA											Filter0
CHA 0	0x0E07	0x0E00	0x0E0B	0x0E0A	0x0E09	0x0E08	0x0E04	0x0E03	0x0E02	0x0E01	0x0E05
CHA 1	0x0E15	0x0E0E	0x0E19	0x0E18	0x0E17	0x0E16	0x0E12	0x0E11	0x0E10	0x0E0F	0x0E13
CHA 2	0x0E23	0x0E1C	0x0E27	0x0E26	0x0E25	0x0E24	0x0E20	0x0E1F	0x0E1E	0x0E1D	0x0E21
CHA 3	0x0E31	0x0E2A	0x0E35	0x0E34	0x0E33	0x0E32	0x0E2E	0x0E2D	0x0E2C	0x0E2B	0x0E2F
CHA 4	0x0E3F	0x0E38	0x0E43	0x0E42	0x0E41	0x0E40	0x0E3C	0x0E3B	0x0E3A	0x0E39	0x0E3D
CHA 5	0x0E4D	0x0E46	0x0E51	0x0E50	0x0E4F	0x0E4E	0x0E4A	0x0E49	0x0E48	0x0E47	0x0E4B
CHA 6	0x0E5B	0x0E54	0x0E5F	0x0E5E	0x0E5D	0x0E5C	0x0E58	0x0E57	0x0E56	0x0E55	0x0E4F
CHA 7	0x0E69	0x0E62	0x0E6D	0x0E6C	0x0E6B	0x0E6A	0x0E66	0x0E65	0x0E64	0x0E63	0x0E67
CHA 8	0x0E77	0x0E70	0x0E7B	0x0E7A	0x0E79	0x0E78	0x0E74	0x0E73	0x0E72	0x0E71	0x0E75
CHA 9	0x0E85	0x0E7E	0x0E89	0x0E88	0x0E87	0x0E86	0x0E82	0x0E81	0x0E80	0x0E7F	0x0E83
CHA 10	0x0E93	0x0E8C	0x0E97	0x0E96	0x0E95	0x0E94	0x0E90	0x0E8F	0x0E8E	0x0E8D	0x0E91
CHA 11	0x0EA1	0x0E9A	0x0EA5	0x0EA4	0x0EA3	0x0EA2	0x0E9E	0x0E9D	0x0E9C	0x0E9B	0x0E9F
CHA 12	0x0EAF	0x0EA8	0x0EB3	0x0EB2	0x0EB1	0x0EB0	0x0EAC	0x0EAB	0x0EAA	0x0EA9	0x0EAD
CHA 13	0x0EBD	0x0EB6	0x0EC1	0x0EC0	0x0EBF	0x0EBE	0x0EBA	0x0EB9	0x0EB8	0x0EB7	0x0EBB
CHA 14	0x0ECB	0x0EC4	0x0ECF	0x0ECE	0x0ECD	0x0ECC	0x0EC8	0x0EC7	0x0EC6	0x0EC5	0x0EBF
CHA 15	0x0ED9	0x0ED2	0x0EDD	0x0EDC	0x0EDB	0x0EDA	0x0ED6	0x0ED5	0x0ED4	0x0ED3	0x0ED7
CHA 16	0x0EE7	0x0EE0	0x0EEB	0x0EEA	0x0EE9	0x0EE8	0x0EE4	0x0EE3	0x0EE2	0x0EE1	0x0EE5
CHA 17	0x0EF5	0x0EEE	0x0EF9	0x0EF8	0x0EF7	0x0EF6	0x0F14	0x0EF1	0x0EF0	0x0EEF	0x0EF3
CHA 18	0x0F11	0x0F0A	0x0F15	0x0F14	0x0F13	0x0F12	0x0F0E	0x0F0D	0x0F0C	0x0F0B	0x0F0F
CHA 19	0x0F1F	0x0F18	0x0F23	0x0F22	0x0F21	0x0F20	0x0F1C	0x0F1B	0x0F1A	0x0F19	0x0F1D
CHA 20	0x0F2D	0x0F26	0x0F31	0x0F30	0x0F2F	0x0F2E	0x0F2A	0x0F29	0x0F28	0x0F27	0x0F2B
CHA 21	0x0F3B	0x0F34	0x0F3F	0x0F3E	0x0F3D	0x0F3C	0x0F38	0x0F37	0x0F36	0x0F35	0x0F2F
CHA 22	0x0F49	0x0F42	0x0F4D	0x0F4C	0x0F4B	0x0F4A	0x0F46	0x0F45	0x0F44	0x0F43	0x0F47
CHA 23	0x0F57	0x0F50	0x0F5B	0x0F5A	0x0F59	0x0F58	0x0F54	0x0F53	0x0F52	0x0F51	0x0F55
CHA 24	0x0F65	0x0F5E	0x0F69	0x0F68	0x0F67	0x0F66	0x0F62	0x0F61	0x0F60	0x0F5F	0x0F63
CHA 25	0x0F73	0x0F6C	0x0F77	0x0F76	0x0F75	0x0F74	0x0F70	0x0F6F	0x0F6E	0x0F6D	0x0F71
CHA 26	0x0F81	0x0F7A	0x0F85	0x0F84	0x0F83	0x0F82	0x0F7E	0x0F7D	0x0F7C	0x0F7B	0x0F7F
CHA 27	0x0F8F	0x0F88	0x0F93	0x0F92	0x0F91	0x0F90	0x0F8C	0x0F8B	0x0F8A	0x0F89	0x0F8D
CHA 28	0x0F9D	0x0F96	0x0FA1	0x0FA0	0x0F9F	0x0F9E	0x0F9A	0x0F99	0x0F98	0x0F97	0x0F9B
CHA 29	0x0FAB	0x0FA4	0x0FAF	0x0FAE	0x0FAD	0x0FAC	0x0FA8	0x0FA7	0x0FA6	0x0FA5	0x0F9F
CHA 30	0x0FB9	0x0FB2	0x0FBD	0x0FBC	0x0FBB	0x0FBA	0x0FB6	0x0FB5	0x0FB4	0x0FB3	0x0FB7
CHA 31	0x0FC7	0x0FC0	0x0FCB	0x0FCA	0x0FC9	0x0FC8	0x0FC4	0x0FC3	0x0FC2	0x0FC1	0x0FC5
CHA 32	0x0FD5	0x0FCE	0x0FD9	0x0FD8	0x0FD7	0x0FD6	0x0FD2	0x0FD1	0x0FD0	0x0FCF	0x0FD3
CHA 33	0x0FE3	0x0FDC	0x0FE7	0x0FE6	0x0FE5	0x0FE4	0x0FE0	0x0FDF	0x0FDE	0x0FDD	0x0FE1
CHA 34	0x0B67	0x0B60	0x0B6B	0x0B6A	0x0B69	0x0B68	0x0B64	0x0B63	0x0B62	0x0B61	0x0B65
CHA 35	0x0B75	0x0B6E	0x0B79	0x0B78	0x0B77	0x0B76	0x0B72	0x0B71	0x0B70	0x0B6F	0x0B73
CHA 36	0x0B83	0x0B7C	0x0B87	0x0B86	0x0B85	0x0B84	0x0B80	0x0B7F	0x0B7E	0x0B7D	0x0B81
CHA 37	0x0B91	0x0B8A	0x0B95	0x0B94	0x0B93	0x0B92	0x0B8E	0x0B8D	0x0B8C	0x0B8B	0x0B8F
CHA 38	0x0B9F	0x0B98	0x0BA3	0x0BA2	0x0BA1	0x0BA0	0x0B9C	0x0B9B	0x0B9A	0x0B99	0x0B9D

Table 1-9. Uncore Performance Monitoring Registers (MSR) (Sheet 2 of 2)

Unit	Unit Status	Unit Ctrl	Ctr3	Ctr2	Ctr1	Ctr0	Ctrl3	Ctrl2	Ctrl1	Ctrl0	Extra
CHA 39	0x0BAD	0x0BA6	0x0BB1	0x0BB0	0x0BAF	0x0BAE	0x0BAA	0x0BA9	0x0BA8	0x0BA7	0x0BAB
M2IOSF Blocks											
M2PCIe											
M2IOSF 0	0x0A45	0x0A40	0x0A44	0x0A43	0x0A42	0x0A41	0x0A49	0x0A48	0x0A47	0x0A46	
M2IOSF 1	0x0A65	0x0A60	0x0A64	0x0A63	0x0A62	0x0A61	0x0A69	0x0A68	0x0A67	0x0A66	
M2IOSF 2	0x0A85	0x0A80	0x0A84	0x0A83	0x0A82	0x0A81	0x0A89	0x0A88	0x0A87	0x0A86	
M2IOSF 3	0x0AD5	0x0AD0	0x0AD4	0x0AD3	0x0AD2	0x0AD1	0x0AD9	0x0AD8	0x0AD7	0x0AD6	
M2IOSF 4	0x0AF5	0x0AF0	0x0AF4	0x0ACF	0x0AF2	0x0AF1	0x0AF9	0x0AF8	0x0AF7	0x0AF6	
M2IOSF 5	0x0B15	0x0B10	0x0B14	0x0B13	0x0B12	0x0B11	0x0B19	0x0B18	0x0B17	0x0B16	
IRP											
M2IOSF 0	0x0A4F	0x0A4A			0x0A4C	0x0A4B			0x0A4E	0x0A4D	
M2IOSF 1	0x0A6F	0x0A6A			0x0A6C	0x0A6B			0x0A6E	0x0A6D	
M2IOSF 2	0x0A8F	0x0A8A			0x0A8C	0x0A8B			0x0A8E	0x0A8D	
M2IOSF 3	0x0ADF	0x0ADA			0x0ADC	0x0ADB			0x0ADE	0x0ADD	
M2IOSF 4	0x0AFF	0x0AFA			0x0AFC	0x0AFB			0x0AFE	0x0AFD	
M2IOSF 5	0x0B1F	0x0B1A			0x0B1C	0x0B1B			0x0B1E	0x0B1D	
TC											IIO Clock
M2IOSF 0	0x0A57	0x0A50	0x0A54	0x0A53	0x0A52	0x0A51	0x0A5B	0x0A5A	0x0A59	0x0A58	0x0A55
M2IOSF 1	0x0A77	0x0A70	0x0A74	0x0A73	0x0A72	0x0A71	0x0A7B	0x0A7A	0x0A79	0x0A78	0x0A75
M2IOSF 2	0x0A97	0x0A90	0x0A94	0x0A93	0x0A92	0x0A91	0x0A9B	0x0A9A	0x0A99	0x0A98	0x0A95
M2IOSF 3	0x0AE7	0x0AE0	0x0AE4	0x0AE3	0x0AE2	0x0AE1	0x0AEB	0x0AEA	0x0AE9	0x0AE8	0x0AE5
M2IOSF 4	0x0B07	0x0B00	0x0B04	0x0B03	0x0B02	0x0B01	0x0B0B	0x0B0A	0x0B09	0x0B08	0x0B05
M2IOSF 5	0x0B27	0x0B20	0x0B24	0x0B23	0x0B22	0x0B21	0x0B2B	0x0B2A	0x0B29	0x0B28	0x0B25
PCU											
PCU	0x0716	0x0710	0x071A	0x0719	0x0718	0x0717	0x0714	0x0713	0x0712	0x0711	
UBox											
UBox	0x0708				0x070A	0x0709			0x0706	0x0705	

There are a number of free-running counters in each IIO Stack that collect counts for Input Bandwidth for each Port. The MSR addresses used to access that state are detailed in the following tables.

Table 1-10. Free-Running IIO Bandwidth Out Counters in MSR space

	Port 7 BW Out	Port 6 BW Out	Port 5 BW Out	Port 4 BW Out	Port 3 BW Out	Port 2 BW Out	Port 1 BW Out	Port 0 BW Out
M2IOSF 0	0x0AAF	0x0AAE	0x0AAD	0x0AAC	0x0AAB	0x0AAA	0x0AA9	0x0AA8
M2IOSF 1	0x0ABF	0x0ABE	0x0ABD	0x0ABC	0x0ABB	0x0ABA	0x0AB9	0x0AB8
M2IOSF 2	0x0ACF	0x0ACE	0x0ACD	0x0ACC	0x0ACB	0x0ACA	0x0AC9	0x0AC8
M2IOSF 3	0x0B3F	0x0B3E	0x0B3D	0x0B3C	0x0B3B	0x0B3A	0x0B39	0x0B38
M2IOSF 4	0x0B4F	0x0B4E	0x0B4D	0x0B4C	0x0B4B	0x0B4A	0x0B49	0x0B48
M2IOSF 5	0x0B5F	0x0B5E	0x0B5D	0x0B5C	0x0B5B	0x0B5A	0x0B59	0x0B58

Table 1-11. Free-running IIO Bandwidth In Counters in MSR space

	Port 7 BW In	Port 6 BW In	Port 5 BW In	Port 4 BW In	Port 3 BW In	Port 2 BW In	Port 1 BW In	Port 0 BW In
M2IOSF 0	0x0AA7	0x0AA6	0x0AA5	0x0AA4	0x0AA3	0x0AA2	0x0AA1	0x0AA0
M2IOSF 1	0x0AB7	0x0AB6	0x0AB5	0x0AB4	0x0AB3	0x0AB2	0x0AB1	0x0AB0
M2IOSF 2	0x0AC7	0x0AC6	0x0AC5	0x0AC4	0x0AC3	0x0AC2	0x0AC1	0x0AC0
M2IOSF 3	0x0B37	0x0B36	0x0B35	0x0B34	0x0B33	0x0B32	0x0B31	0x0B30
M2IOSF 4	0x0B47	0x0B46	0x0B45	0x0B44	0x0B43	0x0B42	0x0B41	0x0B40
M2IOSF 5	0x0B57	0x0B56	0x0B55	0x0B54	0x0B53	0x0B52	0x0B51	0x0B50

Note: Please refer to each Unit's performance monitoring section for any related state not covered here.

1.8.2 Uncore Performance Monitoring State in PCICFG space

The addresses for all basic PMON state addressed through PCICFG space are laid out in the table below. Each such block will have a PCICFG B:D:F and DeviceID. The registers are presented as offsets to the PMON block's base address.

As of 3rd Gen Intel® Xeon® Processor Scalable Family, there are a couple free-running counters in each MC to collect counts for RD/WR Bandwidth.

Table 1-12. Uncore Performance Monitoring Registers (PCICFG) (Sheet 1 of 2)

Unit	Unit Status	Unit Ctrl	Ctr3	Ctr2	Ctr1	Ctr0	Ctrl3	Ctrl2	Ctrl1	Ctrl0	Extra	
Unit - PMON Block			PCICFG B:D:F Base Address			Device ID						
								0x3451				
IMC0			MEM0_BAR B30:D0:F, 0xD8									
IMC1			MEM1_BAR B30:D0:F, 0xD8									
IMC2			MEM2_BAR B30:D0:F, 0xD8									
IMC3			MEM3_BAR B30:D0:F, 0xD8								DCLK Ctr	DCLK Ctrl
Channel 0	0x2285C	0x22800	0x22820	0x22818	0x22810	0x22808	0x2284c	0x22848	0x22844	0x22840	0x22838	0x22854

Table 1-12. Uncore Performance Monitoring Registers (PCICFG) (Sheet 2 of 2)

Unit	Unit Status	Unit Ctrl	Ctr3	Ctr2	Ctr1	Ctr0	Ctrl3	Ctrl2	Ctrl1	Ctrl0	Extra	
Channel 1	0x2685C	0x26800	0x26820	0x26818	0x26810	0x26808	0x2684c	0x26848	0x26844	0x26840	0x26838	0x26854
Channel 2 (*)	0x2A85C	0x2A800	0x2A81C	0x2A818	0x2A810	0x2A808	0x2A84c	0x2A848	0x2A844	0x2A840	0x2A838	0x2A854
iMC Free Running Counters			RD DDR	WR DDR	RD PMM	WR PMM	DCLK					
			0x2290	0x2298	0x22A0	0x22A8	0x22B0					
(*) NOTE: Only TWO channels will ever be active. It may be necessary for tools to check which one is active by checking if any count (other than clockticks), such as CAS counts, is returning a non-zero value.												
M2M - for iMC0			B30:D12:F0			0x344A						
M2M - for iMC1			B30:D13:F0			0x344A						
M2M - for iMC2			B30:D14:F0			0x344A						
M2M - for iMC3			B30:D15:F0			0x344A					See M2M Section	
M2M	0x4A8	*0x438*	0x458	0x450	0x448	0x440	0x480	0x478	0x470	0x468		
DMI			B0:D3:F0			0x334A?					See PCIe3/ DMI Section	
DMI	0x4E0	*0x4E4*	0x500	0x4F8	0x4F0	0x4E8	0x520	0x518	0x510	0x508		
PCIe3 (Across All Ports)			B1:D4:F0			0x334A					See PCIe3/ DMI Section	
PCIe3	0x4E0	*0x4E4*	0x500	0x4F8	0x4F0	0x4E8	0x520	0x518	0x510	0x508		
M3UPI - Link 0			B30:D5:F1			0x3446						
M3UPI - Link 1			B30:D6:F1			0x3446						
M3UPI - Link 2			B30:D7:F1			0x3446						
M3UPI	0xF8	0xA0	0xC0	0xB8	0xB0	0xA8	0xE4	0xE0	0xDC	0xD8		
UPI LL - Link 0			B30:D2:F1			0x3441						
UPI LL - Link 1			B30:D3:F1			0x3441						
UPI LL - Link 2			B30:D4:F1			0x3441						
UPI LL	0x37C	0x318	0x338	0x330	0x328	0x320	0x368	0x360	0x358	0x350		

1.9 Some Guidance for SW

1.9.1 On Finding the Package's Bus number for Uncore PMON registers in PCICFG Space

PCI-based uncore units in 3rd Gen Intel® Xeon® Processor Scalable Family can be found using bus, device and functions numbers. However, the **busno** has to be found dynamically in each package. The code is embedded below.

First, for each package, it is necessary to read the node ID offset in the Ubox. That needs to match the GID offset of the Ubox in a specific pattern to get the busno for the package. This busno can then be used with the given D:F (device:function) listed with each box's counters that are accessed through PCICfg space (Table 1-11, "Uncore Performance Monitoring Registers (MMIO/PCICFG)").

Note: ed: The one undefined piece in the following code is PCI_Read_Ulong, a function that simply reads the value from the PCI address. This function, or ones like it, can be found in a more general PCI library the composition of which is OS dependent.

Unfortunately, a link to a suitable version of the library was not readily available. Below are links to a comparable open source version of the library. Included for reference:

https://github.com/opcm/pcm/blob/master/ pci.h and pci.cpp

```
#define DRV_IS_PCI_VENDOR_ID_INTEL            0x8086
#define VENDOR_ID_MASK                        0x0000FFFF
#define DEVICE_ID_MASK                        0xFFFF00000
#define DEVICE_ID_BITSHIFT                    16

#define PCI_ENABLE                            0x80000000
#define FORM_PCI_ADDR(bus,dev,fun,off)        (((PCI_ENABLE))      | \
                                               ((bus & 0xFF) << 16)| \
                                               ((dev & 0x1F) << 11)| \
                                               ((fun & 0x07) << 8) | \
                                               ((off & 0xFF) << 0))


#define ICELAKE_SERVER_SOCKETID_UBOX_DID   0x3450

//the below LNID and GID applies to 3rd Gen Intel® Xeon® Processor
Scalable Family Server
#define UNC_SOCKETID_UBOX_LNID_OFFSET   0xC0
#define UNC_SOCKETID_UBOX_GID_OFFSET    0xD4

for (bus_no = 0; bus_no < 256; bus_no++) {
  for (device_no = 0; device_no < 32; device_no++) {
   for (function_no = 0; function_no < 8; function_no++) {

     // find bus, device, and function number for socket ID UBOX device
     pci_address = FORM_PCI_ADDR(bus_no, device_no, function_no, 0);
     value = PCI_Read_Ulong(pci_address);
```

```
        vendor_id = value & VENDOR_ID_MASK;
        device_id = (value & DEVICE_ID_MASK) >> DEVICE_ID_BITSHIFT;

        if (vendor_id != DRV_IS_PCI_VENDOR_ID_INTEL) {
          continue;
           }
        if (device_id == ICELAKE_SERVER_SOCKETID_UBOX_DID) {
         // first get node id for the local socket
         pci_address = FORM_PCI_ADDR(bus_no, device_no, function_no,
                       UNC_SOCKETID_UBOX_LNID_OFFSET);
         gid = PCI_Read_Ulong(pci_address) & 0x00000007;

         // Get the node id mapping register:
         // Basic idea is to read the Node ID Mapping Register (below)
         // and match one of the nodes with gid that we read above
         // from the Node ID configuration register (above).
         // Every three bits in the Node ID Mapping Register maps to a
         // particular node (or package). Bits 2:0 maps to package 0,
         // bits 5:3 maps to package 1, and so on. Thus, we have to
         // parse every triplet of bits to find the match.

         pci_address = FORM_PCI_ADDR(bus_no, device_no, function_no,
                       UNC_SOCKETID_UBOX_GID_OFFSET);
         mapping = PCI_Read_Ulong(pci_address);


          for (i = 0; i < 8; i++){
           if (nodeid == ((mapping >> (3 * i)) & 0x7)) {
                 gid = i;
                 break;
           }
          }

          UNC_UBOX_package_to_bus_map[gid] = bus_no;
          }
      }
     }

}
```

1.9.2 On Resolving addresses for Uncore PMON registers in MMIO space

MMIO-based uncore units in 3rd Gen Intel® Xeon® Processor Scalable Family can be found by taking the DeviceID and looking up the BAR (base address offset) that governs that unit's registers. For 3rd Gen Intel® Xeon® Processor Scalable Family, the BAR lookup is a two-step process as outlined below.

Once the base address has been resolved, simply add the published offsets to reference the PMON registers (Table 1-12, "Uncore Performance Monitoring Registers (PCICFG)").

```
/* MMIO_BASE found at Bus U0, Device 0, Function 1, offset D0h. */
#define ICX_IMC_MMIO_BASE_OFFSET      0xd0
#define ICX_IMC_MMIO_BASE_MASK        0x1FFFFFFF
/* MEM0_BAR found at Bus U0, Device 0, Function 1, offset D8h. */
#define ICX_IMC_MMIO_MEM0_OFFSET      0xd8
#define ICX_IMC_MMIO_MEM_STRIDE       0xd04
#define ICX_IMC_MMIO_MEM_MASK        0x7FF
/*
* Each IMC has two channels. But there is addressing for three. Need to
determine which two channels are active on the system.
* The offset starts from 0x22800 with stride 0x4000
*/
#define ICX_IMC_MMIO_CHN_OFFSET       0x22800
#define ICX_IMC_MMIO_CHN_STRIDE       0x4000
/* IMC MMIO size*/
#define ICX_IMC_MMIO_SIZE             0x4000


/*
* pkg_id: Socket id
* imc_idx: The IMC index
* channel_idx: The channel index
*/
Void *map_imc_pmon(int pkg_id, int imc_idx, int channel_idx)
{
       struct pci_dev *pdev = NULL;
       resource_size_t addr;
       u32 pci_dword;
       void *io_addr;
       int mem_offset;

/*
       * Device ID of Bus U0, Device 0, Function 1 is 0x3451 */
       * Get its pdev on the specific socket.
*/
       while(1){
              pdev = pci_get_device(PCI_VENDOR_ID_INTEL, 0x3451, pdev);
                            if ((!pdev) || (pdev->bus ==
UNC_UBOX_package_to_bus_map[pkg_id]))
                                          break;
               }
       if (!pdev)
              return NULL;

       /* read MEMn addr (51:23) from MMIO_BASE register */
       pci_read_config_dword(pdev, ICX_IMC_MMIO_BASE_OFFSET, &pci_dword);
       addr = (pci_dword & ICX_IMC_MMIO_BASE_MASK) << 23;

       /* read MEMn addr (22:12) from MEMn_BAR register */
```

```
        mem_offset = ICX_IMC_MMIO_MEM0_OFFSET + mem_idx *
ICX_IMC_MMIO_MEM_STRIDE;
      pci_read_config_dword(pdev, mem_offset, &pci_dword);
      addr |= (pci_dword & ICX_IMC_MMIO_MEM_MASK) << 12;

      /* IMC PMON registers start from PMONUNITCTRL */
      addr += ICX_IMC_MMIO_CHN_OFFSET + channel_idx *
ICX_IMC_MMIO_CHN_STRIDE;

      /* map the IMC PMON registers */
      io_addr = ioremap(addr, ICX_IMC_MMIO_SIZE);

      return io_addr;
}
```

1.9.3 On Finding PCIe Root Port Bus Numbers for IO PMUs

```
      #define VENDOR_ID_MASK                        0x0000FFFF
      #define DEVICE_ID_MASK                        0xFFFF0000
      #define DEVICE_ID_BITSHIFT                16

      #define DRV_IS_PCI_VENDOR_ID_INTEL        0x8086
      #define MMAP_REG_DEV_ID                        0x09a2

      #define SAD_CONTROL_CFG_OFFSET                0x3f4

      #define PCI_ENABLE                            0x80000000
      #define FORM_PCI_ADDR(bus,dev,fun,off)        (((PCI_ENABLE)) | \
                              ((bus & 0xFF) << 16)| \
                              ((dev & 0x1F) << 11)| \
                              ((fun & 0x07) << 8) | \
                              ((off & 0xFF) << 0))

      for (bus_no = 0; bus_no < 256; bus_no++) {
           for (device_no = 0; device_no < 32; device_no++) {
                for (function_no = 0; function_no < 8; function_no++) {
                     pci_address = FORM_PCI_ADDR(bus_no, device_no,
                     function_no, 0);
                     value = PCI_Read_Ulong(pci_address);
                     vendor_id = value & VENDOR_ID_MASK;
                     device_id = (value & DEVICE_ID_MASK) >>
                     DEVICE_ID_BITSHIFT;
                     if ((vendor_id != DRV_IS_PCI_VENDOR_ID_INTEL) &&
                         (device_id != MMAP_REG_DEV_ID)) {
                          continue;
                     }

                     pci_address = FORM_PCI_ADDR(bus_no, device_no,
                     function_no,
```

```
                                    SAD_CONTROL_CFG_OFFSET);
                            value = PCI_Read_Ulong(pci_address);

                            m2iosf_bus = bus_no;
                            m2iosf_socket_id = value & 0xf;
                            m2iosf_stack_id = (((value >> 4) & 0x7) + 5) % 6;

                            m2iosf_bus_numbers[m2iosf_socket_id][m2iosf_stack_i
                            d] = m2iosf_bus;
                        }
                    }
                }
```

1.9.4 On determining the number of CHAs on the system

```
#define ICX_CAPID6                      0x9c
#define ICX_CAPID7                      0xa0

static u64 icx_count_chabox(void)
{
                struct pci_dev *dev = NULL;
                u64 caps = 0;

                dev = pci_get_device(PCI_VENDOR_ID_INTEL, 0x345b, dev);
                if (!dev)
                                goto out;

                pci_read_config_dword(dev, ICX_CAPID6, (u32 *)&caps);
                pci_read_config_dword(dev, ICX_CAPID7, (u32 *)&caps + 1);
out:
                pci_dev_put(dev);
                return hweight64(caps);
}
```

Alternatively,

1.9.5 Setting up a Monitoring Session

On HW reset, all the counters are disabled. Enabling is hierarchical. So the following steps, which include programming the event control registers and enabling the counters to begin collecting events, must be taken to set up a monitoring session. Section 1.9.6 covers the steps to stop/re-start counter registers during a monitoring session.

Global Settings in the UBox: (NOTE: Necessary for U-Box monitoring).

a) Freeze all the uncore counters by setting U_MSR_PMON_GLOBAL_CTL.frz_all to 1

OR (if box level freeze control preferred)

a) Freeze the box's counters while setting up the monitoring session.

for example, set Cn_MSR_PMON_BOX_CTL.frz to 1

For each event to be measured within each box:

b) Enable counting for each monitor

For example, set C0_MSR_PMON_CTL2.en to 1

Note: Recommended: set the *.en* bit for all counters in each box a user intends to monitor, and leave alone for the duration of the monitoring session.

Note: For cases where there is no sharing of these counters among software agents that are independently sampling the counters, software could set the enable bits for all counters it intends to use during the setup phase. For cases where sharing is expected, each agent could use the individual enable bits in order to perform sampling rather than using the box-level freeze from steps (a) and (d).

c) Select event to monitor if the event control register hasn't been programmed:

Program the *.ev_sel* and *.umask* bits in the control register with the encoding necessary to capture the requested event along with any signal conditioning bits (*.thresh*/*.edge_det*/*.invert*) used to qualify the event.

Back to the box level:

d) Reset counters in each box to ensure no stale values have been acquired from previous sessions. Resetting the control registers, particularly those that won't be used is also recommended if for no other reason than to prevent errant overflows. To reset both the counters and control registers write the following registers:

- For each CHAx, set Cn_MSR_PMON_UNIT_CTL[1:0] to 0x3.
- For each DRAM Channel, set MCn_CHy_PCI_PMON_UNIT_CTL[1:0] to 0x3.
- Set PCU_MSR_PMON_UNIT_CTL[1:0] to 0x3.
- For each Intel® UPI Link, set M3_Ly_PCI_PMON_UNIT_CTL[1:0] to 0x3.
- For each Intel® UPI Link, set UPI_Ly_PCI_PMON_UNIT_CTL[1:0] to 0x3.
- For each IIO stack, set M2n_PCI_PMON_UNIT_CTL[1:0] to 0x3.
- For each IIO stack, set IIOn_MSR_PMON_UNIT_CTL[1:0] to 0x3
- For each IIO stack, set IRPn_MSR_PMON_UNIT_CTL[1:0] to 0x3

Note: The UBox does not have a Unit Control register. The counters will need to be manually reset by writing a 0 in each data register.

e) Select how to gather data. *If polling, skip to f*. If sampling:

To set up a **sample interval**, software can pre-program the data register with a value of [2^(register bit width - up to 48) - sample interval length]. Doing so allows software, through use of the pmi mechanism, to be **notified** when the number of events in the sample have been captured. Capturing a performance monitoring sample every 'X cycles' (the fixed counter in the UBox counts uncore clock cycles) is a common use of this mechanism.

That is, to stop counting and receive notification when the 1,000,000th data flit is transmitted from Intel UPI on Link 0

- set UPI_L0_PCI_PMON_CTR1 to (2^48- 1000)
- set UPI_L0_PCI_PMON_CTL1.ev_sel to 0x2

— set UPI_L0_PCI_PMON_CTL1.umask to 0xF

— set U_MSR_PMON_GLOBAL_CTL.pmi_core_sel to which core the monitoring thread is executing on.

f) Enable counting at the global level by setting the U_MSR_PMON_GLOBAL_CTL.unfrz_all bit to 1.

OR

f) Enable counting at the box level by unfreezing the counters in each box

For example, set Cn_MSR_PMON_BOX_CTL.frz to 0

And with that, counting begins.

Note: The UBox does not have a Unit Control register, so there's no box-level freeze to help isolate the UBox from agents counting in other boxes. Once enabled and programmed with a valid event, the UBox counters will collect events. For somewhat better synchronization, a user can keep the U_MSR_PMON_CTL.ev_sel at 0x0 while enabled and write it with a valid value just prior to unfreezing the registers in other boxes.

1.9.6 Reading the Sample Interval

Software can **poll** the counters whenever it chooses, or wait to be **notified** that a counter has overflowed (by receiving a PMI).

a) **Polling** - before reading, it is recommended that software freeze the counters at either the Global level (U_MSR_PMON_GLOBAL_CTL.frz_all) or in each box with active counters (by setting *_PMON_UNIT_CTL.frz to 1). After reading the event counts from the counter registers, the monitoring agent can choose to reset the event counts to avoid event-count wrap-around; or resume the counter register without resetting their values. The latter choice will require the monitoring agent to check and adjust for potential wrap-around situations.

b) **Frozen** counters - If software set the counters to freeze on overflow and send notification when it happens, the next question is: Who caused the freeze?

Overflow bits are stored hierarchically within the 3rd Gen Intel® Xeon® Processor Scalable Family uncore. First, software should read the U_MSR_PMON_GLOBAL_STATUS.ov_* bits to determine which box(es) sent an overflow. Then read that box's *_PMON_GLOBAL_STATUS.ov field to find the overflowing counter.

Note: More than one counter may overflow at any given time.

Note: Certain boxes may have more than one PMON block (for example, IMC has a PMON block in each Channel). It may be necessary to read all STATUS registers in the box to determine which counter overflowed.

1.9.7 Enabling a New Sample Interval from Frozen Counters

a) **Clear all uncore counters**: For each box in which counting occurred, set *_PMON_BOX_CTL.rst_ctrs to 1.

b) **Clear all overflow bits**. This includes clearing U_MSR_PMON_GLOBAL_STATUS.ov_* as well as any *_BOX_STATUS registers that have their overflow bits set.

For example, if counter 3 in Intel UPI Link 1 overflowed, software should set UPI_L1_PCI_PMON_BOX_STATUS.ov[3] to 1 to clear the overflow.

c) **Create the next sample**: Reinitialize the sample by setting the monitoring data register to (2^48 - sample_interval). Or set up a new sample interval as outlined in Section 1.9.5, "Setting up a Monitoring Session."

d) **Re-enable counting**: Set U_MSR_PMON_GLOBAL_CTL.unfrz_all to 1.

1.10 On Parsing and Using Derived Events

For many of the sections covering each box's Performance Monitoring capabilities, a set of commonly measured metrics (or 'Derived Events') has been included. For the most part, these derived events are simple mathematical combinations of events found within the box. However, there are some extensions to the notation used by the metrics.

The following is a breakdown of a CHA Derived Event to illustrate a couple of the notations used.

To calculate "Average Number of Data Read Entries that Miss the LLC when the TOR is not empty".

(TOR_OCCUPANCY.MISS_OPCODE / COUNTER0_OCCUPANCY{edge_det,thresh=0x1})) with:Cn_MSR_PMON_BOX_FILTER1.opc=0x182.

Note: On Ice Lake the opcode filter has been added to each counter control register. The new version of the equation is (TOR_OCCUPANCY.IA_MISS_DRD / COUNTER0_OCCUPANCY{edge_det,thresh=0x1})). However, the old version of the equation is used to illustrate the concepts behind the Derived Event syntax.

First term is a normal Event/Subevent.

Second Term requires setting extra control bits in the register the event has been programmed in:

- event_name[.subevent_name]{ctrl_bit[=value],}
- For example, `COUNTER0_OCCUPANCY{edge_det,thresh=0x1}`

Note: If there is no [=value] specified it is assumed that the bit must be set to 1.

Third Term requires programming an extra control register (often for filtering):

- For a single field: with:`Register_Name.field=value1`
- For multiple fields:
  with:`Register_Name.{field1,field2,...}={value1,value2,...}`
- For example,
  `with:Cn_MSR_PMON_BOX_FILTER1.{opc,nid}={0x182,my_node}`

Following is a breakdown of an IMC Derived Event to illustrate a couple more of the notations used.

To calculate "Percent Cycles DRAM Rank x in CKE".

POWER_CKE_CYCLES.RANKx / MC_Chy_PCI_PMON_CTR_FIXED

First Term requires more input to software to determine the specific event/subevent

- In some cases, there may be multiple events/subevents that cover the same information across multiple like hardware units. Rather than manufacturing a

derived event for each combination, the derived event will use a lower case variable in the event name.

- For example, `POWER_CKE_CYCLES.RANKx` where 'x' is a variable to cover events `POWER_CKE_CYCLES.RANK0` through `POWER_CKE_CYCLES.RANK7`

Second Term requires reading a fixed data register

- For the case where the metric requires the information contained in a fixed data register, the pnemonic for the register will be included in the equation. Software will be responsible for configuring the data register and setting it to start counting with the other events used by the metric.
- For example, `MC_Chy_PCI_PMON_CTR_FIXED`

In addition to these formats, some equations require gathering of extra information outside the box (often for common terms):

- See following section for a breakdown of common terms found in Derived Events.

1.10.1 On Common Terms found in Derived Events

To convert a Latency term from a count of clocks to a count of nanoseconds:

- (**Latency Metric**) - {Box}_CLOCKTICKS * (1000 / UNCORE_FREQUENCY)

To convert a Bandwidth term from a count of raw bytes at the operating clock to GB/sec:

- ((**Traffic Metric in Bytes**) / (TOTAL_INTERVAL / (TSC_SPEED * 1000000))) / GB_CONVERSION
- For example, for READ_MEM_BW, an event derived from IMC:CAS_COUNT.RD * 64, which is the amount of memory bandwidth consumed by read requests, put 'READ_MEM_BW' into the bandwidth term to convert the measurement from raw bytes to GB/sec.

Following are some other terms that may be found within Metrics and how they should be interpreted.

- GB_CONVERSION: 1024^3
- TSC_SPEED: Time Stamp Counter frequency in MHz
- TOTAL_INTERVAL: Overall sample interval (TSC) for the instructions retired event. Typically used to compute a per send metric. Dividing the TOTAL_INTERVAL by CPU_SPEED * 1,000,000 is the number of seconds in the sample interval.
- TOTAL_PROC_CYC: Total number of CPU cycles for a processor event value. Used with processor event data to determine time or work per time as in MB/sec. UPI_LINKS: 2-3 for 3rd Gen Intel® Xeon® Processor Scalable Family
- IMC_CHANNELS: Up to 8 for 3rd Gen Intel® Xeon® Processor Scalable Family

§

2 3rd Gen Intel® Xeon® Processor Scalable Family, Uncore Performance Monitoring

2.1 UBox Performance Monitoring

The UBox serves as the system configuration controller for 3rd Gen Intel Xeon Processor Scalable Family

In this capacity, the UBox acts as the central unit for a variety of functions:

- The master for reading and writing physically distributed registers across using the Message Channel.
- The UBox is the intermediary for interrupt traffic, receiving interrupts from the system and dispatching interrupts to the appropriate core.
- The UBox serves as the system lock master used when quiescing the platform (for example, Intel® UPI bus lock).

2.1.1 UBox Performance Monitoring Overview

The UBox supports event monitoring through two programmable 48-bit wide counters (U_MSR_PMON_CTR{1:0}), and a 48-bit fixed counter which increments each U-clock. Each of these counters can be programmed (U_MSR_PMON_CTL{1:0}) to monitor any UBox event.

2.1.2 Additional UBox Performance Monitoring

Note: The UBox is the only PMON block without a separate Unit Control register. The only way to freeze the UBox counters is to use the global freeze.

Figure 2-1. PMON Control Register for UCLK

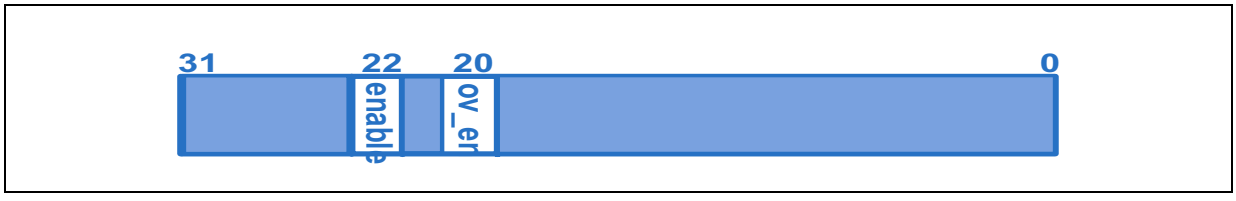


Table 2-1. U_MSR_PMON_FIXED_CTL Register – Field Definitions (Sheet 1 of 2)

Field	Bits	Attr	HW Reset Val	Description
rsv	31:23	RV	0	Reserved
en	22	RW-V	0	Enable counter when global enable is set.

Table 2-1. U_MSR_PMON_FIXED_CTL Register – Field Definitions (Sheet 2 of 2)

Field	Bits	Attr	HW Reset Val	Description
rsv	21	RV	0	Reserved. SW must write to 0 else behavior is undefined.
ov_en	20	RW-V	0	When this bit is set to 1 and the corresponding counter overflows, a UBox overflow message is sent to the UBox's global logic. Once received, the global status register will record the overflow in U_MSR_PMON_GLOBAL_STATUS.ov_u_fixed.
rsv	19:0	RV	0	Reserved

Table 2-2. U_MSR_PMON_FIXED_CTR Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
rsv	63:48	RV	0	Reserved
event_count	47:0	RW-V	0	48-bit performance event counter

2.1.3 UBox Performance Monitoring Events

The set of events that can be monitored in the UBox are as follows:

2.1.4 UBOX Box Events Ordered By Code

The following table summarizes the directly measured UBOX Box events.

Symbol Name	Event Code	Ctrs	Extra Select Bit	Max Inc/ Cyc	Description
EVENT_MSG	0x42	0-1	0	1	Message Received
LOCK_CYCLES	0x44	0-1	0	1	IDI Lock/SplitLock Cycles
PHOLD_CYCLES	0x45	0-1	0	1	Cycles PHOLD Assert to Ack
RACU_REQUESTS	0x46	0-1	0	1	RACU Request
RACU_DRNG	0x4c		0	0	
M2U_MISC1	0x4d		0	0	
M2U_MISC2	0x4e		0	0	
M2U_MISC3	0x4f		0	0	

2.1.5 UBOX Box Performance Monitor Event List

The section enumerates the 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the UBOX Box.

EVENT_MSG

- **Title:** Message Received
- **Category:** EVENT_MSG Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

Table 2-3. Unit Masks for EVENT_MSG

Extension	umask [15:8]	Description
VLW_RCVD	bxxxxxxx1	VLW Virtual Logical Wire (legacy) message were received from Uncore.
MSI_RCVD	bxxxxxx1x	MSI Message Signaled Interrupts - interrupts sent by devices (including PCIe via IOxAPIC) (Socket Mode only)
IPI_RCVD	bxxxxx1xx	IPI Inter Processor Interrupts
DOORBELL_RCVD	bxxxx1xxx	Doorbell
INT_PRIO	bxxx1xxxx	Interrupt Interrupts

LOCK_CYCLES

- **Title:** IDI Lock/SplitLock Cycles
- **Category:** LOCK Events
- **Event Code:** 0x44
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Number of times an IDI Lock/SplitLock sequence was started

M2U_MISC1

- **Title:**
- **Category:** M2U Events
- **Event Code:** 0x4d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-4. Unit Masks for M2U_MISC1

Extension	umask [15:8]	Description
RxC_CYCLES_NE_CBO_NCB	bxxxxxxx1	
RxC_CYCLES_NE_CBO_NCS	bxxxxxx1x	
RxC_CYCLES_NE_UPI_NCB	bxxxxx1xx	
RxC_CYCLES_NE_UPI_NCS	bxxxx1xxx	
TxC_CYCLES_CRD_OVF_CBO_NCB	bxxx1xxxx	
TxC_CYCLES_CRD_OVF_CBO_NCS	bxx1xxxxx	
TxC_CYCLES_CRD_OVF_UPI_NCB	bx1xxxxxx	
TxC_CYCLES_CRD_OVF_UPI_NCS	b1xxxxxxx	

M2U_MISC2

- **Title:**
- **Category:** M2U Events
- **Event Code:** 0x4e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-5. Unit Masks for M2U_MISC2

Extension	umask [15:8]	Description
RxC_CYCLES_FULL_BL	bxxxxxxx1	
RxC_CYCLES_EMPTY_BL	bxxxxxx1x	
TxC_CYCLES_CRD_OVF_VN0_NCB	bxxxxx1xx	
TxC_CYCLES_CRD_OVF_VN0_NCS	bxxxx1xxx	
TxC_CYCLES_EMPTY_BL	bxxx1xxxx	
TxC_CYCLES_EMPTY_AK	bxx1xxxxx	
TxC_CYCLES_EMPTY_AKC	bx1xxxxxx	
TxC_CYCLES_FULL_BL	b1xxxxxxx	

M2U_MISC3

- **Title:**
- **Category:** M2U Events
- **Event Code:** 0x4f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-6. Unit Masks for M2U_MISC3

Extension	umask [15:8]	Description
TxC_CYCLES_FULL_AK	bxxxxxxx1	
TxC_CYCLES_FULL_AKC	bxxxxxx1x	

PHOLD_CYCLES

- **Title:** Cycles PHOLD Assert to Ack
- **Category:** PHOLD Events
- **Event Code:** 0x45
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** PHOLD cycles.

Table 2-7. Unit Masks for PHOLD_CYCLES

Extension	umask [15:8]	Description
ASSERT_TO_ACK	bxxxxxxx1	Assert to ACK

RACU_DRNG

- **Title:**
- **Category:** RACU Events
- **Event Code:** 0x4c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-8. Unit Masks for RACU_DRNG

Extension	umask [15:8]	Description
RDRAND	bxxxxxxx1	
RDSEED	bxxxxxx1x	
PFTCH_BUF_EMPTY	bxxxxx1xx	

RACU_REQUESTS

- **Title:** RACU Request
- **Category:** RACU Events
- **Event Code:** 0x46
- **Max. Inc/Cyc:.** 1

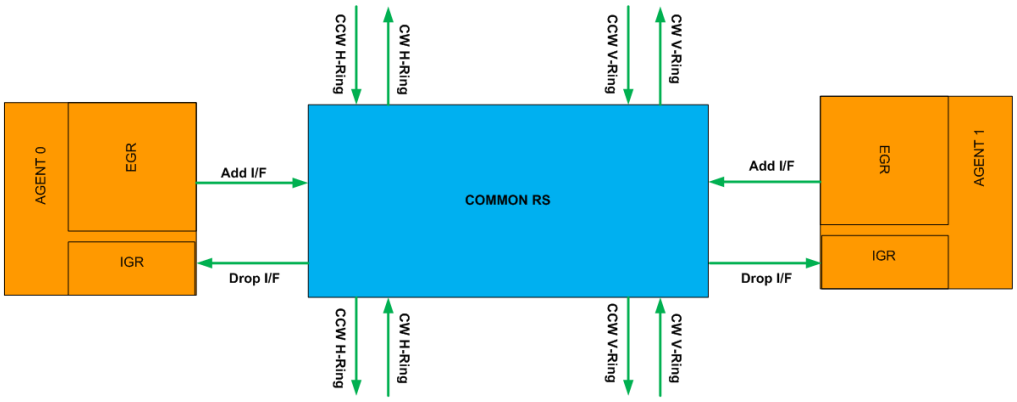
Register Restrictions: 0-1

- **Definition:** Number outstanding register requests within message channel tracker
- **NOTE:** RACU = Register Access Control Unit

2.2 Mesh Performance Monitoring

For all Boxes that must communicate with the Mesh, there are a common set of events to capture various kinds of information about traffic flowing through their connection to the Mesh. The same encodings are used to request the mesh events in each box.

This common mesh stop event list is available in the CHA, M2M, M2PCIe and M3UPI



2.2.1 Mesh Performance Monitoring Events

There are events to track information related to all traffic passing through each box's connection to the Mesh.

- Credit Tracking and Stalls due to lack of credits

Mesh

2.2.2 CMS Box Events Ordered By Code

The following table summarizes the directly measured CMS Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
AG0_AD_CRD_ACQUIRED0	0x80		0	CMS Agent0 AD Credits Acquired
AG0_AD_CRD_ACQUIRED1	0x81		0	CMS Agent0 AD Credits Acquired
AG0_AD_CRD_OCCUPANCY0	0x82		0	CMS Agent0 AD Credits Occupancy
AG0_AD_CRD_OCCUPANCY1	0x83		0	CMS Agent0 AD Credits Occupancy
AG1_AD_CRD_ACQUIRED0	0x84		0	CMS Agent1 AD Credits Acquired
AG1_AD_CRD_ACQUIRED1	0x85		0	CMS Agent1 AD Credits Acquired
AG1_AD_CRD_OCCUPANCY0	0x86		0	CMS Agent1 AD Credits Occupancy
AG1_AD_CRD_OCCUPANCY1	0x87		0	CMS Agent1 AD Credits Occupancy
AG0_BL_CRD_ACQUIRED0	0x88		0	CMS Agent0 BL Credits Acquired
AG0_BL_CRD_ACQUIRED1	0x89		0	CMS Agent0 BL Credits Acquired
AG0_BL_CRD_OCCUPANCY0	0x8a		0	CMS Agent0 BL Credits Occupancy
AG0_BL_CRD_OCCUPANCY1	0x8b		0	CMS Agent0 BL Credits Occupancy
AG1_BL_CRD_ACQUIRED0	0x8c		0	CMS Agent1 BL Credits Acquired
AG1_BL_CRD_ACQUIRED1	0x8d		0	CMS Agent1 BL Credits Acquired
AG1_BL_CRD_OCCUPANCY0	0x8e		0	CMS Agent1 BL Credits Occupancy
AG1_BL_CRD_OCCUPANCY1	0x8f		0	CMS Agent1 BL Credits Occupancy
TxR_VERT_OCCUPANCY0	0x90		0	CMS Vert Egress Occupancy
TxR_VERT_OCCUPANCY1	0x91		0	CMS Vert Egress Occupancy
TxR_VERT_INSERTS0	0x92		0	CMS Vert Egress Allocations
TxR_VERT_INSERTS1	0x93		0	CMS Vert Egress Allocations
TxR_VERT_CYCLES_FULL0	0x94		0	Cycles CMS Vertical Egress Queue Is Full
TxR_VERT_CYCLES_FULL1	0x95		0	Cycles CMS Vertical Egress Queue Is Full
TxR_VERT_CYCLES_NE0	0x96		0	Cycles CMS Vertical Egress Queue Is Not Empty
TxR_VERT_CYCLES_NE1	0x97		0	Cycles CMS Vertical Egress Queue Is Not Empty
TxR_VERT_NACK0	0x98		0	CMS Vertical Egress NACKs
TxR_VERT_NACK1	0x99		0	CMS Vertical Egress NACKs
TxR_VERT_STARVED0	0x9a		0	CMS Vertical Egress Injection Starvation
TxR_VERT_STARVED1	0x9b		0	CMS Vertical Egress Injection Starvation
TxR_VERT_ADS_USED	0x9c		0	CMS Vertical ADS Used

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
TxR_VERT_BYPASS	0x9d		0	CMS Vertical ADS Used
TxR_VERT_BYPASS_1	0x9e		0	CMS Vertical ADS Used
TxR_HORZ_OCCUPANCY	0xa0		0	CMS Horizontal Egress Occupancy
TxR_HORZ_INSERTS	0xa1		0	CMS Horizontal Egress Inserts
TxR_HORZ_CYCLES_FULL	0xa2		0	Cycles CMS Horizontal Egress Queue is Full
TxR_HORZ_CYCLES_NE	0xa3		0	Cycles CMS Horizontal Egress Queue is Not Empty
TxR_HORZ_NACK	0xa4		0	CMS Horizontal Egress NACKs
TxR_HORZ_STARVED	0xa5		0	CMS Horizontal Egress Injection Starvation
TxR_HORZ_ADS_USED	0xa6		0	CMS Horizontal ADS Used
TxR_HORZ_BYPASS	0xa7		0	CMS Horizontal Bypass Used
RING_BOUNCES_VERT	0xaa		2	Messages that bounced on the Vertical Ring.
RING_SINK_STARVED_VERT	0xab		1	Sink Starvation on Vertical Ring
RING_BOUNCES_HORZ	0xac		0	Messages that bounced on the Horizontal Ring.
RING_SINK_STARVED_HORZ	0xad		0	Sink Starvation on Horizontal Ring
RING_SRC_THRTL	0xae		0	Source Throttle
DISTRESS_ASSERTED	0xaf		0	Distress signal asserted
VERT_RING_AD_IN_USE	0xb0		0	Vertical AD Ring In Use
VERT_RING_AK_IN_USE	0xb1		0	Vertical AK Ring In Use
VERT_RING_BL_IN_USE	0xb2		0	Vertical BL Ring in Use
VERT_RING_IV_IN_USE	0xb3		0	Vertical IV Ring in Use
VERT_RING_AKC_IN_USE	0xb4		0	Vertical AKC Ring In Use
VERT_RING_TGC_IN_USE	0xb5		0	Vertical TGC Ring In Use
HORZ_RING_AD_IN_USE	0xb6		0	Horizontal AD Ring In Use
HORZ_RING_AK_IN_USE	0xb7		0	Horizontal AK Ring In Use
HORZ_RING_BL_IN_USE	0xb8		0	Horizontal BL Ring in Use
HORZ_RING_IV_IN_USE	0xb9		0	Horizontal IV Ring in Use
EGRESS_ORDERING	0xba		0	Egress Blocking due to Ordering requirements
HORZ_RING_AKC_IN_USE	0xbb		0	Horizontal AK Ring In Use
CMS_CLOCKTICKS	0xc0		0	CMS Clockticks
STALL0_NO_TxR_HORZ_CRD_AD_AG0	0xd0		0	Stall on No AD Agent0 Transgress Credits
STALL1_NO_TxR_HORZ_CRD_AD_AG0	0xd1		0	Stall on No AD Agent0 Transgress Credits
STALL0_NO_TxR_HORZ_CRD_AD_AG1	0xd2		0	Stall on No AD Agent1 Transgress Credits
STALL1_NO_TxR_HORZ_CRD_AD_AG1_1	0xd3		0	Stall on No AD Agent1 Transgress Credits
STALL0_NO_TxR_HORZ_CRD_BL_AG0	0xd4		0	Stall on No BL Agent0 Transgress Credits
STALL1_NO_TxR_HORZ_CRD_BL_AG0_1	0xd5		0	Stall on No BL Agent0 Transgress Credits
STALL0_NO_TxR_HORZ_CRD_BL_AG1	0xd6		0	Stall on No BL Agent1 Transgress Credits
STALL1_NO_TxR_HORZ_CRD_BL_AG1_1	0xd7		0	Stall on No BL Agent1 Transgress Credits
RxR_OCCUPANCY	0xe0		0	Transgress Ingress Occupancy
RxR_INSERTS	0xe1		0	Transgress Ingress Allocations
RxR_BYPASS	0xe2		0	Transgress Ingress Bypass

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
RxR_CRD_STARVED	0xe3		0	Transgress Injection Starvation
RxR_CRD_STARVED_1	0xe4		0	Transgress Injection Starvation
RxR_BUSY_STARVED	0xe5		0	Transgress Injection Starvation
MISC_EXTERNAL	0xe6		0	Miscellaneous Events (mostly from MS2IDI)

2.2.3 CMS Box Performance Monitor Event List

The section enumerates the 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the CMS Box.

AG0_AD_CRD_ACQUIRED0

- **Title:** CMS Agent0 AD Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x80
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 AD credits acquired in a given cycle, per transgress.
- **NOTE:** If multiple masks are selected, will count the OR of all selected

Table 2-9. Unit Masks for AG0_AD_CRD_ACQUIRED0

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

AG0_AD_CRD_ACQUIRED1

- **Title:** CMS Agent0 AD Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x81
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 AD credits acquired in a given cycle, per transgress.
- **NOTE:** If multiple masks are selected, will count the OR of all selected. Extension not used by Ice Lake.

Table 2-10. Unit Masks for AG0_AD_CRD_ACQUIRED1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

AG0_AD_CRD_OCCUPANCY0

- **Title:** CMS Agent0 AD Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x82
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 AD credits in use in a given cycle, per transgress
- **NOTE:** If multiple masks are selected, will count the SUM of all selected

Table 2-11. Unit Masks for AG0_AD_CRD_OCCUPANCY0

Extension	umask [15:8]	Description
TGR0	b00000001	For Transgress 0
TGR1	b00000010	For Transgress 1
TGR2	b00000100	For Transgress 2
TGR3	b00001000	For Transgress 3
TGR4	b00010000	For Transgress 4
TGR5	b00100000	For Transgress 5
TGR6	b01000000	For Transgress 6
TGR7	b10000000	For Transgress 7

AG0_AD_CRD_OCCUPANCY1

- **Title:** CMS Agent0 AD Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x83
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 AD credits in use in a given cycle, per transgress
- **NOTE:** If multiple masks are selected, will count the SUM of all selected. Extension not used by Ice Lake.

Table 2-12. Unit Masks for AG0_AD_CRD_OCCUPANCY1

Extension	umask [15:8]	Description
TGR8	b00000001	For Transgress 8
TGR9	b00000010	For Transgress 9
TGR10	b00000100	For Transgress 10

AG0_BL_CRD_ACQUIRED0

- **Title:** CMS Agent0 BL Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x88
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 BL credits acquired in a given cycle, per transgress.

Table 2-13. Unit Masks for AG0_BL_CRD_ACQUIRED0

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

AG0_BL_CRD_ACQUIRED1

- **Title:** CMS Agent0 BL Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x89
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 BL credits acquired in a given cycle, per transgress.
- **NOTE:** Extension not used by Ice Lake.

Table 2-14. Unit Masks for AG0_BL_CRD_ACQUIRED1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

AG0_BL_CRD_OCCUPANCY0

- **Title:** CMS Agent0 BL Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x8a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 BL credits in use in a given cycle, per transgress

Table 2-15. Unit Masks for AG0_BL_CRD_OCCUPANCY0

Extension	umask [15:8]	Description
TGR0	b00000001	For Transgress 0
TGR1	b00000010	For Transgress 1
TGR2	b00000100	For Transgress 2
TGR3	b00001000	For Transgress 3
TGR4	b00010000	For Transgress 4
TGR5	b00100000	For Transgress 5
TGR6	b01000000	For Transgress 6
TGR7	b10000000	For Transgress 7

AG0_BL_CRD_OCCUPANCY1

- **Title:** CMS Agent0 BL Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x8b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 0 BL credits in use in a given cycle, per transgress
- **NOTE:** Extension not used by Ice Lake.

Table 2-16. Unit Masks for AG0_BL_CRD_OCCUPANCY1

Extension	umask [15:8]	Description
TGR8	b00000001	For Transgress 8
TGR9	b00000010	For Transgress 9
TGR10	b00000100	For Transgress 10

AG1_AD_CRD_ACQUIRED0

- **Title:** CMS Agent1 AD Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x84
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 AD credits acquired in a given cycle, per transgress.
- **NOTE:** If multiple masks are selected, will count the OR of all selected

Table 2-17. Unit Masks for AG1_AD_CRD_ACQUIRED0

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3

Table 2-17. Unit Masks for AG1_AD_CRD_ACQUIRED0

Extension	umask [15:8]	Description
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

AG1_AD_CRD_ACQUIRED1

- **Title:** CMS Agent1 AD Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x85
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 AD credits acquired in a given cycle, per transgress.
- **NOTE:** If multiple masks are selected, will count the OR of all selected. Extension not used by Ice Lake.

Table 2-18. Unit Masks for AG1_AD_CRD_ACQUIRED1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

AG1_AD_CRD_OCCUPANCY0

- **Title:** CMS Agent1 AD Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x86
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 AD credits in use in a given cycle, per transgress
- **NOTE:** If multiple masks are selected, will count the SUM of all selected

Table 2-19. Unit Masks for AG1_AD_CRD_OCCUPANCY0

Extension	umask [15:8]	Description
TGR0	b00000001	For Transgress 0
TGR1	b00000010	For Transgress 1
TGR2	b00000100	For Transgress 2
TGR3	b00001000	For Transgress 3
TGR4	b00010000	For Transgress 4
TGR5	b00100000	For Transgress 5
TGR6	b01000000	For Transgress 6
TGR7	b10000000	For Transgress 7

AG1_AD_CRD_OCCUPANCY1

- **Title:** CMS Agent1 AD Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x87
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 AD credits in use in a given cycle, per transgress
- **NOTE:** If multiple masks are selected, will count the SUM of all selected. Extension not used by Ice Lake.

Table 2-20. Unit Masks for AG1_AD_CRD_OCCUPANCY1

Extension	umask [15:8]	Description
TGR8	b00000001	For Transgress 8
TGR9	b00000010	For Transgress 9
TGR10	b00000100	For Transgress 10

AG1_BL_CRD_ACQUIRED0

- **Title:** CMS Agent1 BL Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x8c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 BL credits acquired in a given cycle, per transgress.

Table 2-21. Unit Masks for AG1_BL_CRD_ACQUIRED0

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 4
TGR7	b1xxxxxxx	For Transgress 5

AG1_BL_CRD_ACQUIRED1

- **Title:** CMS Agent1 BL Credits Acquired
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x8d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 BL credits acquired in a given cycle, per transgress.
- **NOTE:** Extension not used by Ice Lake.

Table 2-22. Unit Masks for AG1_BL_CRD_ACQUIRED1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

AG1_BL_CRD_OCCUPANCY0

- **Title:** CMS Agent1 BL Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x8e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 BL credits in use in a given cycle, per transgress

Table 2-23. Unit Masks for AG1_BL_CRD_OCCUPANCY0

Extension	umask [15:8]	Description
TGR0	b00000001	For Transgress 0
TGR1	b00000010	For Transgress 1
TGR2	b00000100	For Transgress 2
TGR3	b00001000	For Transgress 3
TGR4	b00010000	For Transgress 4
TGR5	b00100000	For Transgress 5
TGR6	b01000000	For Transgress 6
TGR7	b10000000	For Transgress 7

AG1_BL_CRD_OCCUPANCY1

- **Title:** CMS Agent1 BL Credits Occupancy
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0x8f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of CMS Agent 1 BL credits in use in a given cycle, per transgress
- **NOTE:** Extension not used by Ice Lake.

Table 2-24. Unit Masks for AG1_BL_CRD_OCCUPANCY1

Extension	umask [15:8]	Description
TGR8	b00000001	For Transgress 8
TGR9	b00000010	For Transgress 9
TGR10	b00000100	For Transgress 10

CMS_CLOCKTICKS

- **Title:** CMS Clockticks
- **Category:** Misc Events
- **Event Code:** 0xc0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

DISTRESS_ASSERTED

- **Title:** Distress signal asserted
- **Category:** Horizontal RING Events
- **Event Code:** 0xaf
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles either the local or incoming distress signals are asserted.

Table 2-25. Unit Masks for DISTRESS_ASSERTED

Extension	umask [15:8]	Description
VERT	b00000001	Vertical If IRQ egress is full, then agents will throttle outgoing AD IDI transactions
HORZ	b00000010	Horizontal If TGR egress is full, then agents will throttle outgoing AD IDI transactions
DPT_LOCAL	bxxxxx1xx	DPT Local Dynamic Prefetch Throttle triggered by this tile
DPT_NONLOCAL	bxxxx1xxx	DPT Remote Dynamic Prefetch Throttle received by this tile
PMM_LOCAL	bxxx1xxxx	PMM Local If the CHA TOR has too many PMM transactions, this signal will throttle outgoing MS2IDI traffic
PMM_NONLOCAL	bxx1xxxxx	PMM Remote If another CHA TOR has too many PMM transactions, this signal will throttle outgoing MS2IDI traffic
DPT_STALL_IV	bx1xxxxxx	DPT Stalled - IV DPT occurred while regular IVs were received, causing DPT to be stalled
DPT_STALL_NOCRD	b1xxxxxxx	DPT Stalled - No Credit DPT occurred while credit not available causing DPT to be stalled

EGRESS_ORDERING

- **Title:** Egress Blocking due to Ordering requirements
- **Category:** Horizontal In Use RING Events
- **Event Code:** 0xba
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of cycles IV was blocked in the TGR Egress due to SNP/GO Ordering requirements

Table 2-26. Unit Masks for EGRESS_ORDERING

Extension	umask [15:8]	Description
IV_SNOOPGO_UP	bxxxxxxx1	Up
IV_SNOOPGO_DN	bxxxxx1xx	Down

HORZ_RING_AD_IN_USE

- **Title:** Horizontal AD Ring In Use
- **Category:** Horizontal In Use RING Events
- **Event Code:** 0xb6
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Horizontal AD ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop. We really have two rings -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-27. Unit Masks for HORZ_RING_AD_IN_USE

Extension	umask [15:8]	Description
LEFT_EVEN	bxxxxxxx1	Left and Even
LEFT_ODD	bxxxxxx1x	Left and Odd
RIGHT_EVEN	bxxxxx1xx	Right and Even
RIGHT_ODD	bxxxx1xxx	Right and Odd

HORZ_RING_AKC_IN_USE

- **Title:** Horizontal AK Ring In Use
- **Category:** Horizontal In Use RING Events
- **Event Code:** 0xbb
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Horizontal AKC ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop.We really have two rings in JKT -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-28. Unit Masks for HORZ_RING_AKC_IN_USE

Extension	umask [15:8]	Description
LEFT_EVEN	bxxxxxxx1	Left and Even
LEFT_ODD	bxxxxxx1x	Left and Odd
RIGHT_EVEN	bxxxxx1xx	Right and Even
RIGHT_ODD	bxxxx1xxx	Right and Odd

HORZ_RING_AK_IN_USE

- **Title:** Horizontal AK Ring In Use
- **Category:** Horizontal In Use RING Events
- **Event Code:** 0xb7
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Horizontal AK ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop.We really have two rings -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-29. Unit Masks for HORZ_RING_AK_IN_USE

Extension	umask [15:8]	Description
LEFT_EVEN	bxxxxxxx1	Left and Even
LEFT_ODD	bxxxxxx1x	Left and Odd
RIGHT_EVEN	bxxxxx1xx	Right and Even
RIGHT_ODD	bxxxx1xxx	Right and Odd

HORZ_RING_BL_IN_USE

- **Title:** Horizontal BL Ring in Use
- **Category:** Horizontal In Use RING Events
- **Event Code:** 0xb8
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Horizontal BL ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop.We really have two rings -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-30. Unit Masks for HORZ_RING_BL_IN_USE

Extension	umask [15:8]	Description
LEFT_EVEN	bxxxxxxx1	Left and Even
LEFT_ODD	bxxxxxx1x	Left and Odd
RIGHT_EVEN	bxxxxx1xx	Right and Even
RIGHT_ODD	bxxxx1xxx	Right and Odd

HORZ_RING_IV_IN_USE

- **Title:** Horizontal IV Ring in Use
- **Category:** Horizontal In Use RING Events
- **Event Code:** 0xb9
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Horizontal IV ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop. There is only 1 IV ring. Therefore, if one wants to monitor the "Even" ring, they should select both UP_EVEN and DN_EVEN. To monitor the "Odd" ring, they should select both UP_ODD and DN_ODD.

Table 2-31. Unit Masks for HORZ_RING_IV_IN_USE

Extension	umask [15:8]	Description
LEFT	bxxxxxxx1	Left
RIGHT	bxxxxx1xx	Right

MISC_EXTERNAL

- **Title:** Miscellaneous Events (mostly from MS2IDI)
- **Category:** External Misc Events (for example, from MS2IDI)
- **Event Code:** 0xe6
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** ONLY relevant to the CHA's CMS

Table 2-32. Unit Masks for MISC_EXTERNAL

Extension	umask [15:8]	Description
MBE_INST0	bxxxxxxx1	Number of cycles MBE is high for MS2IDI0
MBE_INST1	bxxxxxx1x	Number of cycles MBE is high for MS2IDI1

RING_BOUNCES_HORZ

- **Title:** Messages that bounced on the Horizontal Ring.
- **Category:** Horizontal RING Events
- **Event Code:** 0xac
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles incoming messages from the Horizontal ring that were bounced, by ring type.

Table 2-33. Unit Masks for RING_BOUNCES_HORZ

Extension	umask [15:8]	Description
AD	bxxxxxxx1	AD
AK	bxxxxxx1x	AK
BL	bxxxxx1xx	BL
IV	bxxxx1xxx	IV

RING_BOUNCES_VERT

- **Title:** Messages that bounced on the Vertical Ring.
- **Category:** Vertical RING Events
- **Event Code:** 0xaa
- **Max. Inc/Cyc:.** 2

Register Restrictions:

- **Definition:** Number of cycles incoming messages from the Vertical ring that were bounced, by ring type.

Table 2-34. Unit Masks for RING_BOUNCES_VERT

Extension	umask [15:8]	Description
AD	bxxxxxxx1	AD
AK	bxxxxxx1x	Acknowledgments to core
BL	bxxxxx1xx	Data Responses to core
IV	bxxxx1xxx	Snoops of processor's cache.
AKC	bxxx1xxxx	

RING_SINK_STARVED_HORZ

- **Title:** Sink Starvation on Horizontal Ring
- **Category:** Horizontal RING Events
- **Event Code:** 0xad
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-35. Unit Masks for RING_SINK_STARVED_HORZ

Extension	umask [15:8]	Description
AD	bxxxxxxx1	AD
AK	bxxxxxx1x	AK
BL	bxxxxx1xx	BL
IV	bxxxx1xxx	IV
AK_AG1	bxx1xxxxx	Acknowledgments to Agent 1

RING_SINK_STARVED_VERT

- **Title:** Sink Starvation on Vertical Ring
- **Category:** Vertical RING Events
- **Event Code:** 0xab
- **Max. Inc/Cyc:.** 1

Register Restrictions:

- **Definition:**

Table 2-36. Unit Masks for RING_SINK_STARVED_VERT

Extension	umask [15:8]	Description
AD	bxxxxxxx1	AD
AK	bxxxxxx1x	Acknowledgments to core
BL	bxxxxx1xx	Data Responses to core
IV	bxxxx1xxx	Snoops of processor's cache.
AKC	bxxx1xxxx	

RING_SRC_THRTL

- **Title:** Source Throttle
- **Category:** Horizontal RING Events
- **Event Code:** 0xae
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxR_BUSY_STARVED

- **Title:** Transgress Injection Starvation
- **Category:** CMS Transgress INGRESS Events
- **Event Code:** 0xe5
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts cycles under injection starvation mode. This starvation is triggered when the CMS Ingress cannot send a transaction onto the mesh for a long period of time. In this case, because a message from the other queue has higher priority
- **NOTE:** If both masks are selected for one ring type (ex: AD CRD + BNC), will count the OR of the two. Selecting multiple ring types NOT supported

Table 2-37. Unit Masks for RxR_BUSY_STARVED

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
BL_UNCRD	b00000100	BL - Uncredited
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited

RxR_BYPASS

- **Title:** Transgress Ingress Bypass
- **Category:** CMS Transgress INGRESS Events
- **Event Code:** 0xe2
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of packets bypassing the CMS Ingress
- **NOTE:** If both masks are selected for one ring type (ex: AD CRD + BNC), will count the SUM of the two. Selecting multiple ring types NOT supported

Table 2-38. Unit Masks for RxR_BYPASS

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

RxR_CRD_STARVED

- **Title:** Transgress Injection Starvation
- **Category:** CMS Transgress INGRESS Events
- **Event Code:** 0xe3
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts cycles under injection starvation mode. This starvation is triggered when the CMS Ingress cannot send a transaction onto the mesh for a long

period of time. In this case, the Ingress is unable to forward to the Egress due to a lack of credit.

- **NOTE:** If both masks are selected for one ring type (ex: AD CRD + BNC), will count the OR of the two. For this purpose IFV is considered an AK ring type. Selecting multiple ring types NOT supported

Table 2-39. Unit Masks for RxR_CRD_STARVED

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
IFV	b10000000	IFV - Credited

RxR_CRD_STARVED_1

- **Title:** Transgress Injection Starvation
- **Category:** CMS Transgress INGRESS Events
- **Event Code:** 0xe4
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts cycles under injection starvation mode. This starvation is triggered when the CMS Ingress cannot send a transaction onto the mesh for a long period of time. In this case, the Ingress is unable to forward to the Egress due to a lack of credit.
- **NOTE:** If both masks are selected for one ring type (ex: AD CRD + BNC), will count the OR of the two. For this purpose IFV is considered an AK ring type. Selecting multiple ring types NOT supported

RxR_INSERTS

- **Title:** Transgress Ingress Allocations
- **Category:** CMS Transgress INGRESS Events
- **Event Code:** 0xe1
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of allocations into the CMS Ingress The Ingress is used to queue up requests received from the mesh
- **NOTE:** If both masks are selected for one ring type (ex: AD CRD + BNC), will count the SUM of the two. Selecting multiple ring types NOT supported

Table 2-40. Unit Masks for RxR_INSERTS

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK

Table 2-40. Unit Masks for RxR_INSERTS

Extension	umask [15:8]	Description
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

RxR_OCCUPANCY

- **Title:** Transgress Ingress Occupancy
- **Category:** CMS Transgress INGRESS Events
- **Event Code:** 0xe0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Occupancy event for the Ingress buffers in the CMS. The Ingress is used to queue up requests received from the mesh
- **NOTE:** If both masks are selected for one ring type (ex: AD CRD + BNC), will count the SUM of the two. Selecting multiple ring types NOT supported

Table 2-41. Unit Masks for RxR_OCCUPANCY

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b00100000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

STALL0_NO_TxR_HORZ_CRD_AD_AG0

- **Title:** Stall on No AD Agent0 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the AD Agent 0 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.

Table 2-42. Unit Masks for STALL0_NO_TxR_HORZ_CRD_AD_AG0

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

STALL0_NO_TxR_HORZ_CRD_AD_AG1

- **Title:** Stall on No AD Agent1 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd2
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the AD Agent 1 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.

Table 2-43. Unit Masks for STALL0_NO_TxR_HORZ_CRD_AD_AG1

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

STALL0_NO_TxR_HORZ_CRD_BL_AG0

- **Title:** Stall on No BL Agent0 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd4
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the BL Agent 0 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.

Table 2-44. Unit Masks for STALL0_NO_TxR_HORZ_CRD_BL_AG0

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

STALL0_NO_TxR_HORZ_CRD_BL_AG1

- **Title:** Stall on No BL Agent1 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd6
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the BL Agent 1 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.

Table 2-45. Unit Masks for STALL0_NO_TxR_HORZ_CRD_BL_AG1

Extension	umask [15:8]	Description
TGR0	bxxxxxxx1	For Transgress 0
TGR1	bxxxxxx1x	For Transgress 1
TGR2	bxxxxx1xx	For Transgress 2
TGR3	bxxxx1xxx	For Transgress 3
TGR4	bxxx1xxxx	For Transgress 4
TGR5	bxx1xxxxx	For Transgress 5
TGR6	bx1xxxxxx	For Transgress 6
TGR7	b1xxxxxxx	For Transgress 7

STALL1_NO_TxR_HORZ_CRD_AD_AG0

- **Title:** Stall on No AD Agent0 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd1
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the AD Agent 0 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.
- **NOTE:** Extension not used by Ice Lake.

Table 2-46. Unit Masks for STALL1_NO_TxR_HORZ_CRD_AD_AG0

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

STALL1_NO_TxR_HORZ_CRD_AD_AG1_1

- **Title:** Stall on No AD Agent1 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd3
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the AD Agent 1 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.
- **NOTE:** Extension not used by Ice Lake.

Table 2-47. Unit Masks for STALL1_NO_TxR_HORZ_CRD_AD_AG1_1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

STALL1_NO_TxR_HORZ_CRD_BL_AG0_1

- **Title:** Stall on No BL Agent0 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd5
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the BL Agent 0 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.
- **NOTE:** Extension not used by Ice Lake.

Table 2-48. Unit Masks for STALL1_NO_TxR_HORZ_CRD_BL_AG0_1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

STALL1_NO_TxR_HORZ_CRD_BL_AG1_1

- **Title:** Stall on No BL Agent1 Transgress Credits
- **Category:** CMS Transgress Credit Events
- **Event Code:** 0xd7
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the BL Agent 1 Egress Buffer is stalled waiting for a TGR credit to become available, per transgress.
- **NOTE:** Extension not used by Ice Lake.

Table 2-49. Unit Masks for STALL1_NO_TxR_HORZ_CRD_BL_AG1_1

Extension	umask [15:8]	Description
TGR8	bxxxxxxx1	For Transgress 8
TGR9	bxxxxxx1x	For Transgress 9
TGR10	bxxxxx1xx	For Transgress 10

TxR_HORZ_ADS_USED

- **Title:** CMS Horizontal ADS Used
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa6
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of packets using the Horizontal Anti-Deadlock Slot, broken down by ring type and CMS Agent.

Table 2-50. Unit Masks for TxR_HORZ_ADS_USED

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
BL_UNCRD	b00000100	BL - Uncredited
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited

TxR_HORZ_BYPASS

- **Title:** CMS Horizontal Bypass Used
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa7
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of packets bypassing the Horizontal Egress, broken down by ring type and CMS Agent.

Table 2-51. Unit Masks for TxR_HORZ_BYPASS

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_HORZ_CYCLES_FULL

- **Title:** Cycles CMS Horizontal Egress Queue is Full
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa2
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Cycles the Transgress buffers in the Common Mesh Stop are Full. The egress is used to queue up requests destined for the Horizontal Ring on the Mesh.

Table 2-52. Unit Masks for TxR_HORZ_CYCLES_FULL

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_HORZ_CYCLES_NE

- **Title:** Cycles CMS Horizontal Egress Queue is Not Empty
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa3
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Cycles the Transgress buffers in the Common Mesh Stop are Not-Empty. The egress is used to queue up requests destined for the Horizontal Ring on the Mesh.

Table 2-53. Unit Masks for TxR_HORZ_CYCLES_NE

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_HORZ_INSERTS

- **Title:** CMS Horizontal Egress Inserts
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa1
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of allocations into the Transgress buffers in the Common Mesh Stop. The egress is used to queue up requests destined for the Horizontal Ring on the Mesh.

Table 2-54. Unit Masks for TxR_HORZ_INSERTS

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_HORZ_NACK

- **Title:** CMS Horizontal Egress NACKs
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa4
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of Egress packets NACK'ed on to the Horizontal Ring

Table 2-55. Unit Masks for TxR_HORZ_NACK

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_HORZ_OCCUPANCY

- **Title:** CMS Horizontal Egress Occupancy
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Occupancy event for the Transgress buffers in the Common Mesh Stop. The egress is used to queue up requests destined for the Horizontal Ring on the Mesh.

Table 2-56. Unit Masks for TxR_HORZ_OCCUPANCY

Extension	umask [15:8]	Description
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AD_CRD	b00010000	AD - Credited
AD_ALL	b00010001	AD - All All == Credited + Uncredited
BL_CRD	b01000000	BL - Credited
BL_ALL	b01000100	BL - All All == Credited + Uncredited
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_HORZ_STARVED

- **Title:** CMS Horizontal Egress Injection Starvation
- **Category:** CMS Horizontal EGRESS Events
- **Event Code:** 0xa5
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts injection starvation. This starvation is triggered when the CMS Transgress buffer cannot send a transaction onto the Horizontal ring for a long period of time.

Table 2-57. Unit Masks for TxR_HORZ_STARVED

Extension	umask [15:8]	Description
AD_ALL	b00000001	AD - All All == Credited + Uncredited
AD_UNCRD	b00000001	AD - Uncredited
AK	b00000010	AK
BL_ALL	b00000100	BL - All All == Credited + Uncredited
BL_UNCRD	b00000100	BL - Uncredited
IV	b00001000	IV
AKC_UNCRD	b10000000	AKC - Uncredited

TxR_VERT_ADS_USED

- **Title:** CMS Vertical ADS Used
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x9c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of packets using the Vertical Anti-Deadlock Slot, broken down by ring type and CMS Agent.

Table 2-58. Unit Masks for TxR_VERT_ADS_USED

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0
BL_AG0	bxxxxx1xx	BL - Agent 0
AD_AG1	bxxx1xxxx	AD - Agent 1
BL_AG1	bx1xxxxxx	BL - Agent 1

TxR_VERT_BYPASS

- **Title:** CMS Vertical ADS Used
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x9d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of packets bypassing the Vertical Egress, broken down by ring type and CMS Agent.

Table 2-59. Unit Masks for TxR_VERT_BYPASS

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0
AK_AG0	bxxxxxx1x	AK - Agent 0
BL_AG0	bxxxxx1xx	BL - Agent 0
IV_AG1	bxxxx1xxx	IV - Agent 1
AD_AG1	bxxx1xxxx	AD - Agent 1
AK_AG1	bxx1xxxxx	AK - Agent 1
BL_AG1	bx1xxxxxx	BL - Agent 1

TxR_VERT_BYPASS_1

- **Title:** CMS Vertical ADS Used
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x9e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of packets bypassing the Vertical Egress, broken down by ring type and CMS Agent.

Table 2-60. Unit Masks for TxR_VERT_BYPASS_1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0
AKC_AG1	bxxxxxx1x	AKC - Agent 1

TxR_VERT_CYCLES_FULL0

- **Title:** Cycles CMS Vertical Egress Queue Is Full
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x94
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the Common Mesh Stop Egress was Not Full. The Egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-61. Unit Masks for TxR_VERT_CYCLES_FULL0

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AK_AG0	bxxxxxx1x	AK - Agent 0 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.
BL_AG0	bxxxxx1xx	BL - Agent 0 Ring transactions from Agent 0 destined for the BL ring. This is commonly used to send data from the cache to various destinations.

Table 2-61. Unit Masks for TxR_VERT_CYCLES_FULL0

Extension	umask [15:8]	Description
IV_AG0	bxxxx1xxx	IV - Agent 0 Ring transactions from Agent 0 destined for the IV ring. This is commonly used for snoops to the cores.
AD_AG1	bxxx1xxxx	AD - Agent 1 Ring transactions from Agent 1 destined for the AD ring. This is commonly used for outbound requests.
AK_AG1	bxx1xxxxx	AK - Agent 1 Ring transactions from Agent 1 destined for the AK ring.
BL_AG1	bx1xxxxxx	BL - Agent 1 Ring transactions from Agent 1 destined for the BL ring. This is commonly used for transferring writeback data to the cache.

TxR_VERT_CYCLES_FULL1

- **Title:** Cycles CMS Vertical Egress Queue Is Full
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x95
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the Common Mesh Stop Egress was Not Full. The Egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-62. Unit Masks for TxR_VERT_CYCLES_FULL1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AKC_AG1	bxxxxxx1x	AKC - Agent 1 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.

TxR_VERT_CYCLES_NE0

- **Title:** Cycles CMS Vertical Egress Queue Is Not Empty
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x96
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the Common Mesh Stop Egress was Not Empty. The Egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-63. Unit Masks for TxR_VERT_CYCLES_NE0 (Sheet 1 of 2)

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AK_AG0	bxxxxxx1x	AK - Agent 0 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.

Table 2-63. Unit Masks for TxR_VERT_CYCLES_NE0 (Sheet 2 of 2)

Extension	umask [15:8]	Description
BL_AG0	bxxxxx1xx	BL - Agent 0 Ring transactions from Agent 0 destined for the BL ring. This is commonly used to send data from the cache to various destinations.
IV_AG0	bxxxx1xxx	IV - Agent 0 Ring transactions from Agent 0 destined for the IV ring. This is commonly used for snoops to the cores.
AD_AG1	bxxx1xxxx	AD - Agent 1 Ring transactions from Agent 1 destined for the AD ring. This is commonly used for outbound requests.
AK_AG1	bxx1xxxxx	AK - Agent 1 Ring transactions from Agent 1 destined for the AK ring.
BL_AG1	bx1xxxxxx	BL - Agent 1 Ring transactions from Agent 1 destined for the BL ring. This is commonly used for transferring writeback data to the cache.

TxR_VERT_CYCLES_NE1

- **Title:** Cycles CMS Vertical Egress Queue Is Not Empty
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x97
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of cycles the Common Mesh Stop Egress was Not Empty. The Egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-64. Unit Masks for TxR_VERT_CYCLES_NE1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AKC_AG1	bxxxxxx1x	AKC - Agent 1 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.

TxR_VERT_INSERTS0

- **Title:** CMS Vert Egress Allocations
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x92
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of allocations into the Common Mesh Stop Egress. The Egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-65. Unit Masks for TxR_VERT_INSERTS0

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AK_AG0	bxxxxxx1x	AK - Agent 0 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.
BL_AG0	bxxxxx1xx	BL - Agent 0 Ring transactions from Agent 0 destined for the BL ring. This is commonly used to send data from the cache to various destinations.
IV_AG0	bxxxx1xxx	IV - Agent 0 Ring transactions from Agent 0 destined for the IV ring. This is commonly used for snoops to the cores.
AD_AG1	bxxx1xxxx	AD - Agent 1 Ring transactions from Agent 1 destined for the AD ring. This is commonly used for outbound requests.
AK_AG1	bxx1xxxxx	AK - Agent 1 Ring transactions from Agent 1 destined for the AK ring.
BL_AG1	bx1xxxxxx	BL - Agent 1 Ring transactions from Agent 1 destined for the BL ring. This is commonly used for transferring writeback data to the cache.

TxR_VERT_INSERTS1

- **Title:** CMS Vert Egress Allocations
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x93
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of allocations into the Common Mesh Stop Egress. The Egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-66. Unit Masks for TxR_VERT_INSERTS1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AKC_AG1	bxxxxxx1x	AKC - Agent 1 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.

TxR_VERT_NACK0

- **Title:** CMS Vertical Egress NACKs
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x98
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of Egress packets NACK'ed on to the Vertical Ring

Table 2-67. Unit Masks for TxR_VERT_NACK0

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0
AK_AG0	bxxxxxx1x	AK - Agent 0
BL_AG0	bxxxxx1xx	BL - Agent 0
IV_AG0	bxxxx1xxx	IV
AD_AG1	bxxx1xxxx	AD - Agent 1
AK_AG1	bxx1xxxxx	AK - Agent 1
BL_AG1	bx1xxxxxx	BL - Agent 1

TxR_VERT_NACK1

- **Title:** CMS Vertical Egress NACKs
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x99
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of Egress packets NACK'ed on to the Vertical Ring

Table 2-68. Unit Masks for TxR_VERT_NACK1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0
AKC_AG1	bxxxxxx1x	AKC - Agent 1

TxR_VERT_OCCUPANCY0

- **Title:** CMS Vert Egress Occupancy
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x90
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Occupancy event for the Egress buffers in the Common Mesh Stop. The egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-69. Unit Masks for TxR_VERT_OCCUPANCY0 (Sheet 1 of 2)

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AK_AG0	bxxxxxx1x	AK - Agent 0 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.
BL_AG0	bxxxxx1xx	BL - Agent 0 Ring transactions from Agent 0 destined for the BL ring. This is commonly used to send data from the cache to various destinations.

Table 2-69. Unit Masks for TxR_VERT_OCCUPANCY0 (Sheet 2 of 2)

Extension	umask [15:8]	Description
IV_AG0	bxxxx1xxx	IV - Agent 0 Ring transactions from Agent 0 destined for the IV ring. This is commonly used for snoops to the cores.
AD_AG1	bxxx1xxxx	AD - Agent 1 Ring transactions from Agent 1 destined for the AD ring. This is commonly used for outbound requests.
AK_AG1	bxx1xxxxx	AK - Agent 1 Ring transactions from Agent 1 destined for the AK ring.
BL_AG1	bx1xxxxxx	BL - Agent 1 Ring transactions from Agent 1 destined for the BL ring. This is commonly used for transferring writeback data to the cache.

TxR_VERT_OCCUPANCY1

- **Title:** CMS Vert Egress Occupancy
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x91
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Occupancy event for the Egress buffers in the Common Mesh Stop.The egress is used to queue up requests destined for the Vertical Ring on the Mesh.

Table 2-70. Unit Masks for TxR_VERT_OCCUPANCY1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0 Ring transactions from Agent 0 destined for the AD ring. Some example include outbound requests, snoop requests, and snoop responses.
AKC_AG1	bxxxxxx1x	AKC - Agent 1 Ring transactions from Agent 0 destined for the AK ring. This is commonly used for credit returns and GO responses.

TxR_VERT_STARVED0

- **Title:** CMS Vertical Egress Injection Starvation
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x9a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts injection starvation. This starvation is triggered when the CMS Egress cannot send a transaction onto the Vertical ring for a long period of time.

Table 2-71. Unit Masks for TxR_VERT_STARVED0

Extension	umask [15:8]	Description
AD_AG0	bxxxxxxx1	AD - Agent 0
AK_AG0	bxxxxxx1x	AK - Agent 0
BL_AG0	bxxxxx1xx	BL - Agent 0
IV_AG0	bxxxx1xxx	IV

Table 2-71. Unit Masks for TxR_VERT_STARVED0

Extension	umask [15:8]	Description
AD_AG1	bxxx1xxxx	AD - Agent 1
AK_AG1	bxx1xxxxx	AK - Agent 1
BL_AG1	bx1xxxxxx	BL - Agent 1

TxR_VERT_STARVED1

- **Title:** CMS Vertical Egress Injection Starvation
- **Category:** CMS Vertical EGRESS Events
- **Event Code:** 0x9b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts injection starvation. This starvation is triggered when the CMS Egress cannot send a transaction onto the Vertical ring for a long period of time.

Table 2-72. Unit Masks for TxR_VERT_STARVED1

Extension	umask [15:8]	Description
AKC_AG0	bxxxxxxx1	AKC - Agent 0
AKC_AG1	bxxxxxx1x	AKC - Agent 1
TGC	bxxxxx1xx	AKC - Agent 0

VERT_RING_AD_IN_USE

- **Title:** Vertical AD Ring In Use
- **Category:** Vertical In Use RING Events
- **Event Code:** 0xb0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Vertical AD ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop. We really have two rings -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-73. Unit Masks for VERT_RING_AD_IN_USE

Extension	umask [15:8]	Description
UP_EVEN	bxxxxxxx1	Up and Even
UP_ODD	bxxxxxx1x	Up and Odd
DN_EVEN	bxxxxx1xx	Down and Even
DN_ODD	bxxxx1xxx	Down and Odd

VERT_RING_AKC_IN_USE

- **Title:** Vertical AKC Ring In Use
- **Category:** Vertical In Use RING Events
- **Event Code:** 0xb4
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Vertical AKC ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop.We really have two rings in JKT -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-74. Unit Masks for VERT_RING_AKC_IN_USE

Extension	umask [15:8]	Description
UP_EVEN	bxxxxxxx1	Up and Even
UP_ODD	bxxxxxx1x	Up and Odd
DN_EVEN	bxxxxx1xx	Down and Even
DN_ODD	bxxxx1xxx	Down and Odd

VERT_RING_AK_IN_USE

- **Title:** Vertical AK Ring In Use
- **Category:** Vertical In Use RING Events
- **Event Code:** 0xb1
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Vertical AK ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop.We really have two rings in -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-75. Unit Masks for VERT_RING_AK_IN_USE

Extension	umask [15:8]	Description
UP_EVEN	bxxxxxxx1	Up and Even
UP_ODD	bxxxxxx1x	Up and Odd
DN_EVEN	bxxxxx1xx	Down and Even
DN_ODD	bxxxx1xxx	Down and Odd

VERT_RING_BL_IN_USE

- **Title:** Vertical BL Ring in Use
- **Category:** Vertical In Use RING Events
- **Event Code:** 0xb2
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Vertical BL ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop.We really have two rings -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-76. Unit Masks for VERT_RING_BL_IN_USE

Extension	umask [15:8]	Description
UP_EVEN	bxxxxxxx1	Up and Even
UP_ODD	bxxxxxx1x	Up and Odd
DN_EVEN	bxxxxx1xx	Down and Even
DN_ODD	bxxxx1xxx	Down and Odd

VERT_RING_IV_IN_USE

- **Title:** Vertical IV Ring in Use
- **Category:** Vertical In Use RING Events
- **Event Code:** 0xb3
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Vertical IV ring is being used at this ring stop. This includes when packets are passing by and when packets are being sunk, but does not include when packets are being sent from the ring stop. There is only 1 IV ring. Therefore, if one wants to monitor the "Even" ring, they should select both UP_EVEN and DN_EVEN. To monitor the "Odd" ring, they should select both UP_ODD and DN_ODD.

Table 2-77. Unit Masks for VERT_RING_IV_IN_USE

Extension	umask [15:8]	Description
UP	bxxxxxxx1	Up
DN	bxxxxx1xx	Down

VERT_RING_TGC_IN_USE

- **Title:** Vertical TGC Ring In Use
- **Category:** Vertical In Use RING Events
- **Event Code:** 0xb5
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles that the Vertical TGC ring is being used at this ring stop. This includes when packets are passing by and when packets are being

sunk, but does not include when packets are being sent from the ring stop.We really have two rings in JKT -- a clockwise ring and a counter-clockwise ring. On the left side of the ring, the "UP" direction is on the clockwise ring and "DN" is on the counter-clockwise ring. On the right side of the ring, this is reversed. The first half of the CBos are on the left side of the ring, and the 2nd half are on the right side of the ring. In other words (for example), in a 4c part, Cbo 0 UP AD is NOT the same ring as CBo 2 UP AD because they are on opposite sides of the ring.

Table 2-78. Unit Masks for VERT_RING_TGC_IN_USE

Extension	umask [15:8]	Description
UP_EVEN	bxxxxxxx1	Up and Even
UP_ODD	bxxxxxx1x	Up and Odd
DN_EVEN	bxxxxx1xx	Down and Even
DN_ODD	bxxxx1xxx	Down and Odd

2.3 Caching/Home Agent (CHA) Performance Monitoring

The LLC coherence engine and Home agent (CHA) merges the caching agent and home agent (HA) responsibilities of the chip into a single block. In its capacity as a caching agent the CHA manages the interface between the core the IIO devices and the last level cache (LLC). In its capacity as a home agent the CHA manages the interface between the LLC and the rest of the UPI coherent fabric as well as the on die memory controller.

All core and IIO DMA transactions that access the LLC are directed from their source to a CHA via the mesh interconnect. The CHA is responsible for managing data delivery from the LLC to the requestor and maintaining coherence between the all the cores and IIO devices within the socket that share the LLC. It is also responsible for generating snoops and collecting snoop responses from the local cores when the MESIF protocol requires it.

Similarly, all incoming traffic from remote sockets that maps to the socket's local memory are directed from the Intel UPI link(s) to a CHA via the mesh interconnect. The CHA is responsible for managing the coherence across all sockets in the system for the socket's memory following the protocols defined in the Intel UPI Specification. It manages directory state for the local memory, conflicts, and memory ordering rules for such requests.

In the process of maintaining cache coherency within the socket, and across the system in a multi-socket system, the CHA is the gate keeper for all Intel® UPI Interconnect messages that have addresses mapping to the socket's memory as well as the originator of all Intel® UPI Interconnect messages the originate from cores within its socket when attempts are made to access memory in another socket. It is responsible for ensuring that all Intel® UPI messages that pass through the socket remain coherent.

The CHA can manage a large number of simultaneous requests in parallel, but in order to maintain proper memory ordering it does ensure that whenever multiple incoming requests to the same address are pending (whether they originated from a core or IIO device within the socket or came in from another socket through one of the UPI links) only one of those requests is being processed at a time.

The 3rd Gen Intel Xeon Processor Scalable Family uncore contains up to 40 instances of the CHA, each assigned to manage a distinct 1.5 MB slice of the processor's total LLC capacity. Since this LLC cache is not inclusive of the IA cores' internal caches, the total cache capacity of the socket is much larger than the LLC capacity (with an additional caching capacity for each IA core in the socket), and each CHA is responsible for monitor a portion of that available IA core cache capacity for the purpose of maintaining coherence between the IA core caches and the rest of the UPI coherent fabric.

Every physical memory address in the system is uniquely associated with a single CHA instance via a proprietary hashing algorithm that is designed to keep the distribution of traffic across the CHA instances relatively uniform for a wide range of possible address patterns. This enables the individual CHA instances to operate independently, each managing its slice of the physical address space without any CHA in a given socket ever needing to communicate with the other CHAs in that same socket.

2.3.1 CHA Performance Monitoring Overview

Each of the CHAs in the 3rd Gen Intel Xeon Processor Scalable Family uncore supports event monitoring through four 48-bit wide counters (Cn_MSR_PMON_CTR{3:0}). With but a small number of exceptions, each of these counters can be programmed (Cn_MSR_PMON_CTL{3:0}) for any available event.

NOTE: Occupancy Events can only be measured in Counter 0.

CHA counter 0 can increment by a maximum of 20 per cycle; counters 1-3 can increment by 1 per cycle.

Some uncore performance events that monitor transaction activities require additional details that must be programmed in a filter register. Each CHA provides one filter register and allows only one such event to be programmed at a given time, see Section 2.3.2.2, "CHA Filter Registers (Cn_MSR_PMON_BOX_FILTER)."

2.3.1.1 Special Note on CHA Occupancy Events

Although only counter 0 supports occupancy events, it is possible to program counters 1-3 to monitor the same occupancy event by selecting the "OCCUPANCY_COUNTER0" event code on counters 1-3.

This allows:

- Thresholding on all four counters.
  While no more than one queue can be monitored at a time, it is possible to setup different queue occupancy thresholds on each of the four counters. For example, if one wanted to monitor the IRQ, one could setup thresholds of 1, 7, 14, and 18 to get a picture of the time spent at different occupancies in the IRQ.
- Average Latency and Average Occupancy
  It can be useful to monitor the average occupancy in a queue as well as the average number of items in the queue. One could program counter 0 to accumulate the occupancy, counter 1 with the queue's allocations event, and counter 2 with the OCCUPANCY_COUNTER0 event and a threshold of 1. Latency could then be calculated by counter 0 / counter 1, and occupancy by counter 0 / counter 2.

2.3.2 Additional CHA Performance Monitoring

2.3.2.1 CHA PMON Counter Control - Difference from Baseline

CHA performance monitoring control registers provide a small amount of additional functionality. The following table defines those cases.

Figure 2-2. CHA Counter Control Register for 3rd Gen Intel Xeon Processor Scalable Family Server

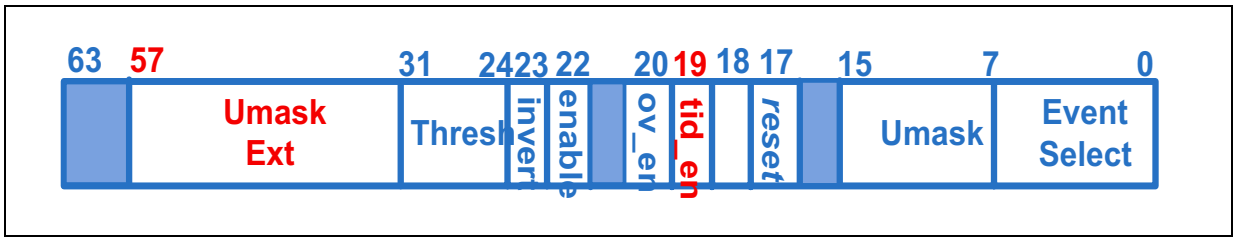


Table 2-79. Cn_MSR_PMON_CTL{3-0} Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
umask_ext	57:32	RW-V	0	Extension to umask. Adds additional filtering capabilities to certain special events such as TOR_INSERTS/OCCUPANCY and LLC_LOOKUP.
tid_en	19	RW-V	0	TID Filter Enable
rsv	16	RV	0	Reserved. SW must write to 0 else behavior is undefined.

Figure 2-3. UmaskExt Filter Details for TOR_INSERT/OCCUPANCY Events

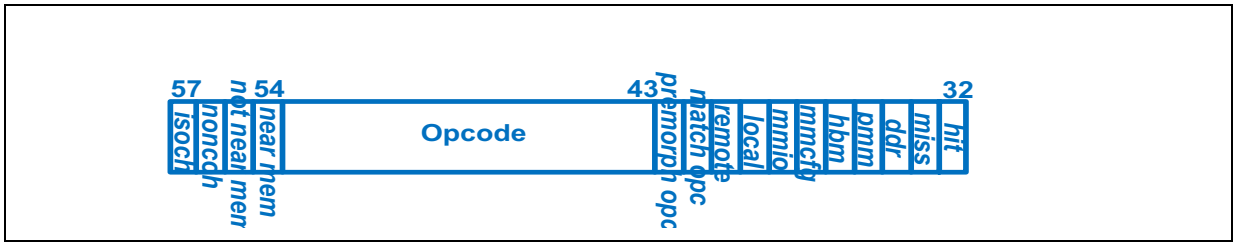


Table 2-80. UmaskExt Filter Details for TOR_INSERT/OCCUPANCY Events (Sheet 1 of 2)

Field	Bits	Atrtr	HW Reset Val	Description
isoc	57	RW	0	Match on ISOC Requests
nc	56	RW	0	Match on Non-Coherent Requests
not_nm	55	RW	0	Just Match on Non Near Memory Cacheable Accesses b55 is XORed with b54. No filtering applied if both bits are either 0 or 1
nm	54	RW	0	Just Match on Near Memory Cacheable Accesses
opc (11b IDI Opcode w/top 2b 0x3)	53:43	RW	0	Match on Opcode (see Section 3.1.1, "Reference for CHA Packet Matching") Can be used to track transaction by opcode relevant to each key queue in the CHA pipeline: IPQ, IRQ, RRQ and WBQ

Table 2-80. UmaskExt Filter Details for TOR_INSERT/OCCUPANCY Events (Sheet 2 of 2)

Field	Bits	Atrtr	HW Reset Val	Description
premorph_opc	42	RW	0	Filter by PreMorphed Opcodes
match_opc	41	RW	0	Filter by Opcodes
loc	40	RW	0	Just Match on Local Node Target b40 is XORed with b39. No filtering applied if both bits are either 0 or 1
rem	39	RW	0	Just Match on Remote Node Target
mmio	38	RW	0	Filter on requests to memory mapped to MMIO space
mmcfg	37	RW	0	Filter on requests to memory mapped to MMCFG space
hbm	36	RW	0	Filter on requests to memory mapped to HBM
pmm	35	RW	0	Filter on requests to memory mapped to PMM
ddr	34	RW	0	Filter on requests to memory mapped to DDR
miss	33	RW	0	Just entries that Missed the LLC b33 is XORed with b32. No filtering applied if both bits are either 0 or 1
hit	32	RW	0	Just entries that Hit the LLC

Figure 2-4. UmaskExt Filter Details for the LLC_LOOKUP Event

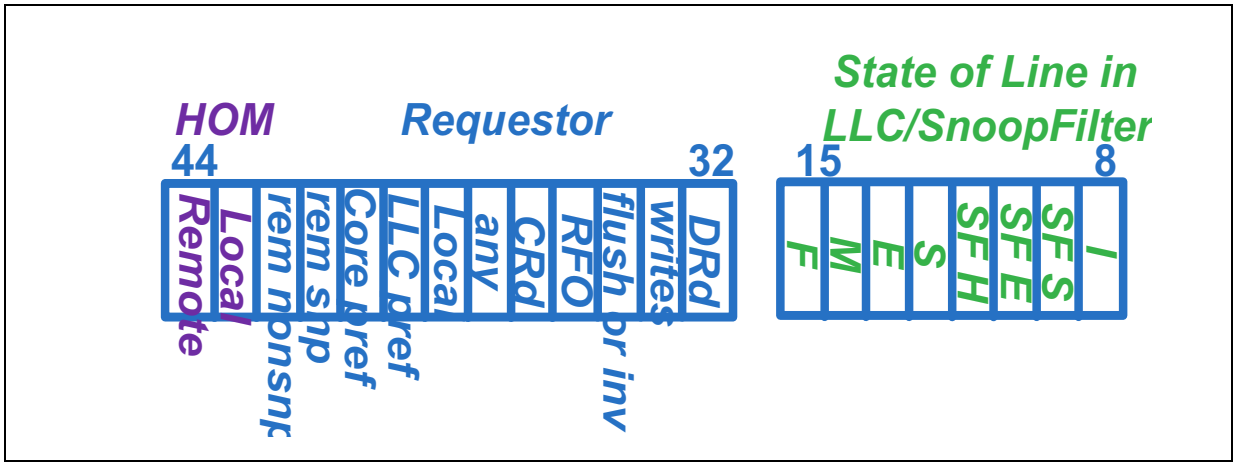


Table 2-81. UmaskExt Filter Details for LLC_LOOKUP Events (Sheet 1 of 2)

Field	Bits	Atrtr	HW Reset Val	Description
remote hom	44	RW	0	Transactions to remotely homed addresses
local hom	43	RW	0	Transactions to locally homed addresses
remote non-snoop	42	RW	0	Non-snoop transactions to the LLC from a remote agent
remote snoop	41	RW	0	Snoop transactions to the LLC from a remote agent
Core prefetch	40	RW	0	Any local prefetch to LLC from Core
LLC prefetch	39	RW	0	Any local prefetch to LLC from an LLC
local	38	RW	0	Any local transaction to LLC, including prefetches from Core

Table 2-81. UmaskExt Filter Details for LLC_LOOKUP Events (Sheet 2 of 2)

Field	Bits	Atrtr	HW Reset Val	Description
any	37	RW	0	Any local or remote transaction to the LLC. Includes prefetches
CRd	36	RW	0	Code Reads- local or remote. includes prefetches
RFO	35	RW	0	RFOs - local or remote. includes prefetches
flush or inv	34	RW	0	Flush or Invalidates
writes	33	RW	0	All write transactions to the LLC - including writebacks to LLC and uncacheable write transactions Does not include evict cleans or invalidates
DRd	32	RW	0	Data Reads- local or remote. includes prefetches
state	umask field 15:8	RW	0	Select state to monitor for **LLC_LOOKUP** event. Setting multiple bits in this field will allow a user to track multiple states. bxx1xxxxxxx - LLC - F state. bxxx1xxxxxx - LLC - M state. bxxxx1xxxxx - LLC - E state. bxxxxx1xxxx - LLC - S state. bxxxxxx1xxx - SF - H state bxxxxxxx1xx - SF - E state. bxxxxxxxx1x - SF - S state. bxxxxxxxxx1 - LLC - I state.

Note: The Request field will be ANDed with State and HOM fields

2.3.2.2 CHA Filter Registers (Cn_MSR_PMON_BOX_FILTER)

Any of the CHA events may be filtered by Thread/Core-ID. To do so, the control register's.*tid_en* bit must be set to 1 and the TID field in the FILTER register filled out.

Figure 2-5. CHA PMON Filter Register

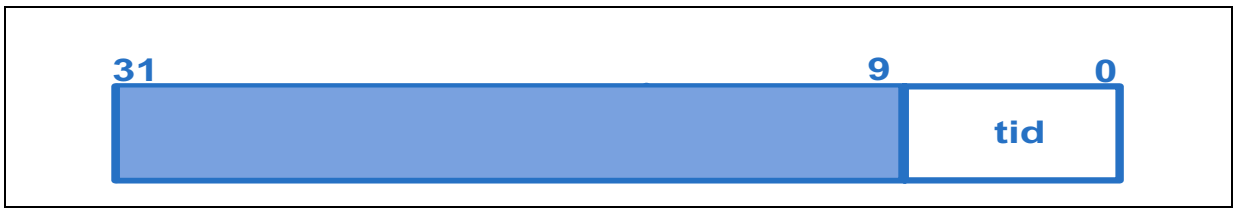


Table 2-82. Cn_MSR_PMON_BOX_FILTER Register – Field Definitions

Field	Bits	Atrtr	HW Reset Val	Description
rsv	31:9	RV	0	Reserved SW must set to 0 else behavior is undefined
tid	8:0	0	0	[8:3] Core-ID [2:0] Thread 3-0 When .tid_en is 0; the specified counter will count ALL events. On 3rd Gen Intel Xeon Processor Scalable Family, 'ALL events' includes traffic from the CPU (for example, Data Requests) and IIO. To filter on a specific logical core, set Core-ID to the desired core number and set the TID field to the desired thread. To filter on a source/destination other than an iA core, set Core-ID to one of the following and set TID to 0: UPI Link 0 or 1: 0x28 UPI Link 2: 0x2C IIO0 (CBDMA or DMI) or IIO1: 3E IIO2: 0x3F IIO3: 0x3B

2.3.3 CHA Performance Monitoring Events

The performance monitoring events within the CHA include all events internal to the LLC and HA as well as events which track mesh related activity at the CHA/Core mesh stops (see Section 2.2.1, "Mesh Performance Monitoring Events" for the available Mesh Stop events).

CHA performance monitoring events can be used to track LLC access rates, LLC hit/miss rates, LLC eviction and fill rates, HA access rates, HA conflicts, and to detect evidence of back pressure on the internal CHA pipelines. In addition, the CHA has performance monitoring events for tracking MESIF state transitions that occur as a result of data sharing across sockets in a multi-socket system.

Every event in the CHA is from the point of view of the CHA and is not associated with any specific core since all cores in the socket send their LLC transactions to all CHAs in the socket. However, the 3rd Gen Intel Xeon Processor Scalable Family CHA provides a thread-id field in the Cn_MSR_PMON_BOX_FILTER register which can be applied to the CHA events to obtain the interactions between specific cores and threads.

There are separate sets of counters for each CHA instance. For any event, to get an aggregate count of that event for the entire LLC, the counts across the CHA instances must be added together. The counts can be averaged across the CHA instances to get a view of the typical count of an event from the perspective of the individual CHAs. Individual per-CHA deviations from the average can be used to identify hot-spotting across the CHAs or other evidences of non-uniformity in LLC behavior across the CHAs. Such hot-spotting should be rare, though a repetitive polling on a fixed physical address is one obvious example of a case where an analysis of the deviations across the CHAs would indicate hot-spotting.

2.3.3.1 Acronyms frequently used in CHA Events

The Rings:

AD (Address) Ring - Core Read/Write Requests and Intel UPI Snoops. Carries Intel UPI requests and snoop responses from C to Intel UPI.

BL (Block or Data) Ring - Data == 2 transfers for 1 cache line

AK (Acknowledge) Ring - Acknowledges Intel UPI to CHA and CHA to Core. Carries snoop responses from Core to CHA.

IV (Invalidate) Ring - CHA Snoop requests of core caches

2.3.3.2 Key Queues

IRQ - Requests from iA Cores

IPQ - Ingress Probe Queue on AD Ring. Remote socket snoops sent from Intel UPI LL.

ISMQ - Ingress Subsequent Messages (response queue). Associated with message responses to ingress requests (for example, data responses, Intel UPI completion messages, core snoop response messages and the GO reset queue).

PRQ - Requests from IIO

RRQ - Remote Request Queue. Remote socket read requests, from UPI to the local home agent.

WBQ - Writeback Queue. Remote socket write requests, from UPI to the local home agent.

TOR - Table Of Requests. Tracks pending CHA transactions.

RxC (aka IGR) /TxC (aka EGR) - Ingress, requests from Cores (by way of the CMS), and Egress, requests headed for the Mesh (by way of the CMS), queues.

2.3.4 CHA Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the CHA Box.

2.3.5 CHA Box Events Ordered By Code

The following table summarizes the directly measured CHA Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
CLOCKTICKS	0x00	0-3	1	Clockticks of the uncore caching and home agent (CHA)
RxC_OCCUPANCY	0x11	0	20	Ingress (from CMS) Occupancy
RxC_INSERTS	0x13	0-3	1	Ingress (from CMS) Allocations
RxC_IRQ0_REJECT	0x18		0	IRQ Requests (from CMS) Rejected - Set 0
RxC_IRQ1_REJECT	0x19		0	IRQ Requests (from CMS) Rejected - Set 1
COUNTER0_OCCUPANCY	0x1f	0-3	20	Counter 0 Occupancy
RxC_PRQ0_REJECT	0x20		0	PRQ Requests (from CMS) Rejected - Set 0
RxC_PRQ1_REJECT	0x21		0	PRQ Requests (from CMS) Rejected - Set 1
RxC_IPQ0_REJECT	0x22		0	IPQ Requests (from CMS) Rejected - Set 0

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
RxC_IPQ1_REJECT	0x23		0	IPQ Requests (from CMS) Rejected - Set 1
RxC_ISMQ0_REJECT	0x24		0	ISMQ Rejects - Set 0
RxC_ISMQ1_REJECT	0x25		0	ISMQ Rejects - Set 1
RxC_RRQ0_REJECT	0x26		0	RRQ Rejects - Set 0
RxC_RRQ1_REJECT	0x27		0	RRQ Rejects - Set 1
RxC_WBQ0_REJECT	0x28		0	WBQ Rejects - Set 0
RxC_WBQ1_REJECT	0x29		0	WBQ Rejects - Set 1
RxC_REQ_Q0_RETRY	0x2a		0	Request Queue Retries - Set 0
RxC_REQ_Q1_RETRY	0x2b		0	Request Queue Retries - Set 1
RxC_ISMQ0_RETRY	0x2c		0	ISMQ Retries - Set 0
RxC_ISMQ1_RETRY	0x2d		0	ISMQ Retries - Set 1
RxC_OTHER0_RETRY	0x2e		0	Other Retries - Set 0
RxC_OTHER1_RETRY	0x2f		0	Other Retries - Set 1
CORE_SNP	0x33		0	Core Cross Snoops Issued
LLC_LOOKUP	0x34	0-3	1	Cache Lookups
TOR_INSERTS	0x35	0-3	1	TOR Inserts
TOR_OCCUPANCY	0x36	0	20	TOR Occupancy
LLC_VICTIMS	0x37	0-3	1	Lines Victimized
MISC	0x39	0-3	1	Cbo Misc
SF_EVICTION	0x3d		0	Snoop Filter Capacity Evictions
PIPE_REJECT	0x42	0-3	1	Pipe Rejects
REQUESTS	0x50	0-3	1	HA Read and Write Requests
SNOOPS_SENT	0x51		0	Snoops Sent
DIR_LOOKUP	0x53		0	Multi-socket cacheline directory state lookups
DIR_UPDATE	0x54		0	Multi-socket cacheline directory state updates
OSB	0x55		0	OSB Snoop Broadcast
WB_PUSH_MTOI	0x56		0	WbPushMtoI
BYPASS_CHA_IMC	0x57		0	CHA to iMC Bypass
READ_NO_CREDITS	0x58		0	CHA iMC CHNx READ Credits Empty
IMC_READS_COUNT	0x59		0	HA to iMC Reads Issued
WRITE_NO_CREDITS	0x5a		0	CHA iMC CHNx WRITE Credits Empty
IMC_WRITES_COUNT	0x5b		0	CHA to iMC Full Line Writes Issued
SNOOP_RESP	0x5c		0	Snoop Responses Received
SNOOP_RESP_LOCAL	0x5d		0	Snoop Responses Received Local
HITME_LOOKUP	0x5e		0	Counts Number of times HitMe Cache is accessed
HITME_HIT	0x5f		0	Counts Number of Hits in HitMe Cache
HITME_MISS	0x60		0	Counts Number of Misses in HitMe Cache
HITME_UPDATE	0x61		0	Counts the number of Allocate/Update to HitMe Cache
PMM_MEMMODE_NM_SETCONFLICTS	0x64		0	PMM Memory Mode related events
PMM_MEMMODE_NM_INVITOX	0x65		0	

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
PMM_QOS	0x66		0	
PMM_QOS_OCCUPANCY	0x67		0	
SNOOP_RSP_MISC	0x6b		0	Misc Snoop Responses Received
DIRECT_GO_OPC	0x6d		0	Direct GO
DIRECT_GO	0x6e		0	Direct GO
XPT_PREF	0x6f		0	XPT Prefetches
PMM_MEMMODE_NM_SETCONFLICTS2	0x70		0	

2.3.6 CHA Box Common Metrics (Derived Events)

The following table summarizes metrics commonly calculated from CHA Box events.

Symbol Name: Definition	Equation
AVG_CRD_MISS_LATENCY: Average Latency of Code Reads from an iA Core that miss the LLC	(TOR_OCCUPANCY.IA_MISS_CRD + TOR_OCCUPANCY.IA_MISS_CRD_PREF) / (TOR_INSERTS.IA_MISS_CRD + TOR_INSERTS.IA_MISS_CRD_PREF)
AVG_DEMAND_RD_HIT_LATENCY: Average Latency of Data Reads that hit the LLC	TOR_OCCUPANCY.IA_HIT_DRD / TOR_INSERTS.IA_HIT_DRD
AVG_DEMAND_RD_MISS_LOCAL_LATENCY: Average Latency of Data Reads from an IA Core that miss the LLC and were satisfied by Local Memory	TOR_OCCUPANCY.IA_MISS_DRD_LOCAL / TOR_INSERTS.IA_MISS_DRD_LOCAL
AVG_DEMAND_RD_MISS_REMOTE_LATENCY: Average Latency of Data Reads from an iA Core that miss the LLC and were satisfied by a Remote cache or Remote Memory	TOR_OCCUPANCY.IA_MISS_DRD_REMOTE / TOR_INSERTS.IA_MISS_DRD_REMOTE
AVG_DRD_MISS_LATENCY: Average Latency of Data Reads or Data Read Prefetches from an IA Core that miss the LLC	(TOR_OCCUPANCY.IA_MISS_DRD + TOR_OCCUPANCY.IA_MISS_DRD_PREF) / (TOR_INSERTS.IA_MISS_DRD + TOR_INSERTS.IA_MISS_DRD_PREF)
AVG_IA_CRD_LLC_HIT_LATENCY: Average Latency of Code Reads from an iA Core that miss the LLC	TOR_OCCUPANCY.IA_HIT_CRD / TOR_INSERTS.IA_HIT_CRD
AVG_INGRESS_DEPTH: Average Depth of the Ingress Queue through the sample interval	RxC_OCCUPANCY.IRQ / SAMPLE_INTERVAL
AVG_INGRESS_LATENCY: Average Latency of Requests through the Ingress Queue in Uncore Clocks	RxC_OCCUPANCY.IRQ / RxC_INSERTS.IRQ
AVG_INGRESS_LATENCY_WHEN_NE: Average Latency of Requests through the Ingress Queue in Uncore Clocks when Ingress Queue has at least one entry	RxC_OCCUPANCY.IRQ / COUNTER0_OCCUPANCY{edge_det,thresh=0x1}
AVG_RFO_MISS_LATENCY: Average Latency of RFOs from an iA Core that miss the LLC	(TOR_OCCUPANCY.IA_MISS_RFO + TOR_OCCUPANCY.IA_MISS_RFO_PREF) / (TOR_INSERTS.IA_MISS_RFO + TOR_INSERTS.IA_MISS_RFO_PREF)
AVG_TOR_DRDS_MISS_WHEN_NE: Average Number of Data Read Entries that Miss the LLC when the TOR is not empty.	TOR_OCCUPANCY.IA_MISS_DRD / COUNTER0_OCCUPANCY{edge_det,thresh=0x1}

Symbol Name: Definition	Equation
AVG_TOR_DRDS_WHEN_NE: Average Number of Data Read Entries when the TOR is not empty.	TOR_OCCUPANCY.IA_DRD / COUNTER0_OCCUPANCY{edge_det,thresh=0x1}
CYC_INGRESS_BLOCKED: Cycles the Ingress Request Queue arbiter was Blocked	RxC_EXT_STARVED.IRQ / SAMPLE_INTERVAL
FAST_STR_LLC_HIT: Number of ItoM (fast string) operations that reference the LLC	TOR_INSERTS.IA_HIT_ITOM
FAST_STR_LLC_MISS: Number of ItoM (fast string) operations that miss the LLC	TOR_INSERTS.IA_MISS_ITOM
INGRESS_REJ_V_INS: Ratio of Ingress Request Entries that were rejected vs. inserted	RxC_INSERTS.IRQ_REJECTED / RxC_INSERTS.IRQ
LLC_CRD_MISS_TO_LOC_MEM: LLC Code Read and Code Prefetch misses satisfied by local memory.	TOR_INSERTS.IA_MISS_CRD_PREF_LOCAL + TOR_INSERTS.IA_MISS_CRD_LOCAL
LLC_CRD_MISS_TO_REM_MEM: LLC Code Read and Code Read Prefetch misses satisfied by a remote cache or remote memory.	TOR_INSERTS.IA_MISS_CRD_PREF_REMOTE + TOR_INSERTS.IA_MISS_CRD_REMOTE
LLC_DRD_MISS_PCT:	LLC_LOOKUP.DATA_READ_MISS / LLC_LOOKUP.DATA_READ_ALL
LLC_DRD_MISS_TO_LOC_MEM: LLC Data Read and Data Prefetch misses satisfied by local memory.	TOR_INSERTS.IA_MISS_DRD_LOCAL
LLC_DRD_MISS_TO_REM_MEM: LLC Data Read and Data Prefetch misses satisfied by a remote cache or remote memory.	TOR_INSERTS.IA_MISS_DRD_REMOTE
LLC_DRD_PREFETCH_HITS:	TOR_INSERTS.IA_HIT_DRD_PREF
LLC_DRD_PREFETCH_MISSES:	TOR_INSERTS.IA_MISS_DRD_PREF
LLC_IA_CRD_HITS:	TOR_INSERTS.IA_HIT_CRD
LLC_MPI: LLC Misses Per Instruction (code, read, RFO and prefetches)	LLC_LOOKUP.MISS_ALL / INST_RETIRED.ALL (on Core)
LLC_PCIE_DATA_BYTES: LLC write miss (disk/network reads) bandwidth in MB	TOR_INSERTS.IO_ITOM * 64
LLC_RFO_MISS_PCT: LLC RFO Miss Ratio	TOR_INSERTS.IA_MISS_RFO / TOR_INSERTS.IA_RFO
LLC_RFO_MISS_TO_LOC_MEM: LLC RFO and RFO Prefetch misses satisfied by local memory.	TOR_INSERTS.IA_MISS_RFO_LOCAL
LLC_RFO_MISS_TO_REM_MEM: LLC RFO and RFO Prefetch misses satisfied by a remote cache or remote memory.	TOR_INSERTS.IA_MISS_RFO_REMOTE
LLC_RFO_PREFETCH_HITS:	TOR_INSERTS.IA_HIT_RFO_PREF
LLC_RFO_PREFETCH_MISSES:	TOR_INSERTS.IA_MISS_RFO_PREF
MEM_WB_BYTES: Data written back to memory in Number of Bytes	LLC_VICTIMS.M_STATE * 64
MMIO_READ_BW: IO Read Bandwidth in MB - Disk or Network Reads	TOR_INSERTS.IA_MISS_UCRDF * 64 / 1000000

Symbol Name: Definition	Equation
MMIO_WRITE_BW: IO Write Bandwidth in MB - Disk or Network Writes	TOR_INSERTS.IA_MISS_WIL* 64 / 1000000
PCIE_FULL_WRITES: Number of full PCI writes	TOR_INSERTS.IO_ITOM
PCI_PARTIAL_WRITES: Number of partial PCI writes	TOR_INSERTS.IO_RFO
PCI_READS: Number of PCI reads	TOR_INSERTS.IO_PCIRDCUR
PCT_RD_REQUESTS: Percentage of HA traffic that is from Read Requests	REQUESTS.READS / (REQUESTS.READS + REQUESTS.WRITES)
PCT_WR_REQUESTS: Percentage of HA traffic that is from Write Requests	REQUESTS.WRITES / (REQUESTS.READS + REQUESTS.WRITES)
STREAMED_FULL_STORES:	TOR_INSERTS.IA_WCILF
STREAMED_FULL_STORES.MISS_LOCAL_TO_DDR:	TOR_INSERTS.IA_MISS_LOCAL_WCILF_DDR
STREAMED_FULL_STORES.MISS_LOCAL_TO_PMM:	TOR_INSERTS.IA_MISS_LOCAL_WCILF_PMM
STREAMED_FULL_STORES.MISS_REMOTE_TO_DDR:	TOR_INSERTS.IA_MISS_REMOTE_WCILF_DDR
STREAMED_FULL_STORES.MISS_REMOTE_TO_PMM:	TOR_INSERTS.IA_MISS_REMOTE_WCILF_PMM
STREAMED_FULL_STORES.MISS_TO_DDR:	TOR_INSERTS.IA_MISS_WCILF_DDR
STREAMED_FULL_STORES.MISS_TO_PMM:	TOR_INSERTS.IA_MISS_WCILF_PMM
STREAMED_PART_STORES:	TOR_INSERTS.IA_WCIL
STREAMED_PART_STORES.MISS_LOCAL_TO_DDR:	TOR_INSERTS.IA_MISS_LOCAL_WCIL_DDR
STREAMED_PART_STORES.MISS_LOCAL_TO_PMM:	TOR_INSERTS.IA_MISS_LOCAL_WCIL_PMM
STREAMED_PART_STORES.MISS_REMOTE_TO_DDR:	TOR_INSERTS.IA_MISS_REMOTE_WCIL_DDR
STREAMED_PART_STORES.MISS_REMOTE_TO_PMM:	TOR_INSERTS.IA_MISS_REMOTE_WCIL_PMM
STREAMED_PART_STORES.MISS_TO_DDR:	TOR_INSERTS.IA_MISS_WCIL_DDR
STREAMED_PART_STORES.MISS_TO_PMM:	TOR_INSERTS.IA_MISS_WCIL_PMM

2.3.7 CHA Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the CHA Box.

BYPASS_CHA_IMC

- **Title:** CHA to iMC Bypass
- **Category:** HA BYPASS Events
- **Event Code:** 0x57
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of times when the CHA was able to bypass HA pipe on the way to iMC. This is a latency optimization for situations when there is light

loadings on the memory subsystem. This can be filtered by when the bypass was taken and when it was not.

Table 2-83. Unit Masks for BYPASS_CHA_IMC

Extension	umask [15:8]	Description
TAKEN	bxxxxxxx1	Taken Filter for transactions that succeeded in taking the full bypass.
INTERMEDIATE	bxxxxxx1x	Intermediate bypass Taken Filter for transactions that succeeded in taking the intermediate bypass.
NOT_TAKEN	bxxxxx1xx	Not Taken Filter for transactions that could not take the bypass, and issues a read to memory. Note that transactions that did not take the bypass but did not issue read to memory will not be counted.

CLOCKTICKS

- **Title:** Clockticks of the uncore caching and home agent (CHA)
- **Category:** UCLK Events
- **Event Code:** 0x00
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

CORE_SNP

- **Title:** Core Cross Snoops Issued
- **Category:** ISMQ Events
- **Event Code:** 0x33
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of transactions that trigger a configurable number of cross snoops. Cores are snooped if the transaction looks up the cache and determines that it is necessary based on the operation type and what CoreValid bits are set. For example, if 2 CV bits are set on a data read, the cores must have the data in S state so it is not necessary to snoop them. However, if only 1 CV bit is set the core my have modified the data. If the transaction was an RFO, it would need to invalidate the lines. This event can be filtered based on who triggered the initial snoop(s).

Table 2-84. Unit Masks for CORE_SNP

Extension	umask [15:8]	Description
REMOTE_ONE	b00010001	Single Snoop Target from Remote
EXT_ONE	b00100001	Single External Snoops
REMOTE_GTONE	b00100010	Multiple Snoop Targets from Remote
EXT_GTONE	b00100010	Multiple External Snoops
CORE_ONE	b01000001	Single Core Requests
CORE_GTONE	b01000010	Multiple Core Requests
EVICT_ONE	b10000001	Single Eviction
EVICT_GTONE	b10000010	Multiple Eviction
ANY_ONE	b11110001	Any Single Snoop
ANY_GTONE	b11110010	Any Cycle with Multiple Snoops

COUNTER0_OCCUPANCY

- **Title:** Counter 0 Occupancy
- **Category:** OCCUPANCY Events
- **Event Code:** 0x1f
- **Max. Inc/Cyc:.** 20

Register Restrictions: 0-3

- **Definition:** Since occupancy counts can only be captured in the Cbo's 0 counter, this event allows a user to capture occupancy related information by filtering the Cb0 occupancy count captured in Counter 0. The filtering available is found in the control register - threshold, invert and edge detect. For example, setting threshold to 1 can effectively monitor how many cycles the monitored queue has an entry.

DIRECT_GO

- **Title:** Direct GO
- **Category:** DIRECT GO Events
- **Event Code:** 0x6e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-85. Unit Masks for DIRECT_GO

Extension	umask [15:8]	Description
HA_TOR_DEALLOC	bxxxxxxx1	
HA_SUPPRESS_NO_D2C	bxxxxxx1x	
HA_SUPPRESS_DRD	bxxxxx1xx	

DIRECT_GO_OPC

- **Title:** Direct GO
- **Category:** DIRECT GO Events
- **Event Code:** 0x6d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-86. Unit Masks for DIRECT_GO_OPC

Extension	umask [15:8]	Description
EXTCMP	bxxxxxxx1	
PULL	bxxxxxx1x	
GO	bxxxxx1xx	
GO_PULL	bxxxx1xxx	
FAST_GO	bxxx1xxxx	
FAST_GO_PULL	bxx1xxxxx	
NOP	bx1xxxxxx	
IDLE_DUE_SUPPRESS	b1xxxxxxx	

DIR_LOOKUP

- **Title:** Multi-socket cacheline directory state lookups
- **Category:** HA DIRECTORY Events
- **Event Code:** 0x53
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of transactions that looked up the directory. Can be filtered by requests that had to snoop and those that did not have to.
- **NOTE:** Only valid for parts that implement the Directory

Table 2-87. Unit Masks for DIR_LOOKUP

Extension	umask [15:8]	Description
SNP	bxxxxxxx1	Snoop Needed Filters for transactions that had to send one or more snoops because the directory was not clean.
NO_SNP	bxxxxxx1x	Snoop Not Needed Filters for transactions that did not have to send any snoops because the directory was clean.

DIR_UPDATE

- **Title:** Multi-socket cacheline directory state updates
- **Category:** HA DIRECTORY Events
- **Event Code:** 0x54
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of directory updates that were required. These result in writes to the memory controller. This can be filtered by directory sets and directory clears.
- **NOTE:** Only valid for parts that implement the Directory

Table 2-88. Unit Masks for DIR_UPDATE

Extension	umask [15:8]	Description
HA	bxxxxxxx1	Directory Updated memory write from HA pipe Counts only directory update Memory writes issued from the HA pipe. Note that any directory update which are part of EWB or IWB are not counted.
TOR	bxxxxxx1x	Directory Updated memory write from TOR pipe Counts only directory update Memory writes issued from the TOR pipe which are the result of remote transaction hitting the SF/LLC and returning data C2C. Note that any directory update which are part of EWB or IWB are not counted.

HITME_HIT

- **Title:** Counts Number of Hits in HitMe Cache
- **Category:** HA HitME Events
- **Event Code:** 0x5f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-89. Unit Masks for HITME_HIT

Extension	umask [15:8]	Description
EX_RDS	bxxxxxxx1	Remote socket read requests that hit in E state. Op is RdCode, RdData, RdDataMigratory, RdCur, RdInv*, Inv*
SHARED_OWNREQ	bxxxxx1xx	Remote socket ownership read requests that hit in S state. Shared hit and op is RdInvOwn, RdInv, Inv*
WBMTOE	bxxxx1xxx	Remote socket WBMtoE requests
WBMTOI_OR_S	bxxx1xxxx	Remote socket writeback to I or S requests op is WbMtoI, WbPushMtoI, WbFlush, or WbMtoS

HITME_LOOKUP

- **Title:** Counts Number of times HitMe Cache is accessed
- **Category:** HA HitME Events
- **Event Code:** 0x5e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-90. Unit Masks for HITME_LOOKUP

Extension	umask [15:8]	Description
READ	bxxxxxxx1	Remote socket read requests op is RdCode, RdData, RdDataMigratory, RdCur, RdInvOwn, RdInv, Inv*
WRITE	bxxxxxx1x	Remote socket write (i.e. writeback) requests op is WbMtoE, WbMtoI, WbPushMtoI, WbFlush, or WbMtoS

HITME_MISS

- **Title:** Counts Number of Misses in HitMe Cache
- **Category:** HA HitME Events
- **Event Code:** 0x60
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-91. Unit Masks for HITME_MISS

Extension	umask [15:8]	Description
SHARED_RDINVOWN	bxx1xxxxx	Remote socket RdInvOwn requests to shared line SF/LLC HitS/F and op is RdInvOwn
NOTSHARED_RDINVOWN	bx1xxxxxx	Remote socket RdInvOwn requests that are not to shared line No SF/LLC HitS/F and op is RdInvOwn
READ_OR_INV	b1xxxxxxx	Remote socket read or invalidate requests op is RdCode, RdData, RdDataMigratory, RdCur, RdInv, Inv*

HITME_UPDATE

- **Title:** Counts the number of Allocate/Update to HitMe Cache
- **Category:** HA HitME Pipe Events
- **Event Code:** 0x61
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-92. Unit Masks for HITME_UPDATE

Extension	umask [15:8]	Description
DEALLOCATE_RSPFWDI_LOC	bxxxxxxx1	op is RspIFwd or RspIFwdWb for a local request Received RspFwdI* for a local request, but converted HitME SF entry
RSPFWDI_REM	bxxxxxx1x	op is RspIFwd or RspIFwdWb for a remote request Updated HitME RspFwdI* or local HitM/E received for a remote request
SHARED	bxxxxx1xx	Update HitMe Cache to SHARed
RDINVOWN	bxxxx1xxx	Update HitMe Cache on RdInvOwn even if not RspFwdI*
DEALLOCATE	bxxx1xxxx	Deallocate HtiME Reads without RspFwdI*

IMC_READS_COUNT

- **Title:** HA to iMC Reads Issued
- **Category:** MC Credit and Traffic Events
- **Event Code:** 0x59
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Count of the number of reads issued to any of the memory controller channels. This can be filtered by the priority of the reads.
- **NOTE:** To match the number of reads seen at the IMC, it's necessary to account for any bypasses. IMC_READS_COUNT.* + BYPASS_CHA_IMC.TAKEN == CAS_COUNT.RD

Table 2-93. Unit Masks for IMC_READS_COUNT

Extension	umask [15:8]	Description
NORMAL	bxxxxxxx1	Normal
PRIORITY	bxxxxxx1x	ISOCH

IMC_WRITES_COUNT

- **Title:** CHA to iMC Full Line Writes Issued
- **Category:** MC Credit and Traffic Events
- **Event Code:** 0x5b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the total number of full line writes issued from the HA into the memory controller.
- **NOTE:** Directory bits are stored in memory. Remote socket RFOs will result in a directory update which, in turn, will cause a write command.

Table 2-94. Unit Masks for IMC_WRITES_COUNT

Extension	umask [15:8]	Description
FULL	bxxxxxxx1	Full Line Non-ISOCH
PARTIAL	bxxxxxx1x	Partial Non-ISOCH
FULL_PRIORITY	bxxxxx1xx	ISOCH Full Line
PARTIAL_PRIORITY	bxxxx1xxx	ISOCH Partial

LLC_LOOKUP

- **Title:** Cache Lookups
- **Category:** CACHE Events
- **Event Code:** 0x34
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times the LLC was accessed - this includes code, data, prefetches and hints coming from L2. This has numerous filters available. Note the non-standard filtering equation. This event will count requests that lookup the cache multiple times with multiple increments. One must ALWAYS select a state or states (in the umask field) to match. Otherwise, the event will count nothing.
- **NOTE:** One of the bits in the umask, those corresponding to each cacheline state (for example, bit0 = invalid (miss), bit4 = shared), and so forth must always be set for this event. To monitor any lookup, set the field to 0xFF.

Table 2-95. Unit Masks for LLC_LOOKUP (Sheet 1 of 3)

Extension	umask [15:8]	xtra [57:32]	Description
COREPREF_OR_DMND_LOCAL_F	bxxxxxxxx	bxxxxxx1xxxxxx	Local request Filter Any local transaction to the LLC, including prefetches from the Core
FLUSH_OR_INV_F	bxxxxxxxx	bxxxxxxxxxx1xx	Flush or Invalidate Filter
REMOTE_SNOOP_F	bxxxxxxxx	bxx1xxxxxxxxxx	Remote snoop request Filter Snoop transactions to the LLC from remote agent
ANY_F	bxxxxxxxx	bxxxxxxx1xxxxx	All Request Filter Any local or remote transaction to the LLC, including prefetch.
LLCPREF_LOCAL_F	bxxxxxxxx	bxxxxx1xxxxxx	Local LLC prefetch requests (from LLC) Filter Any local LLC prefetch to the LLC
CODE_READ_F	bxxxxxxxx	bxxxxxxxx1xxxx	CRd Request Filter Local or remote CRd transactions to the LLC. This includes CRd prefetch.
RFO_F	bxxxxxxxx	bxxxxxxxxx1xxx	RFO Request Filter Local or remote RFO transactions to the LLC. This includes RFO prefetch.
DATA_READ_F	bxxxxxxxx	bxxxxxxxxxxxx1	Data Read Request Filter Read transactions.
LOCAL_F	bxxxxxxxx	bx1xxxxxxxxxx	Transactions homed locally Filter Transaction whose address resides in the local MC.
REMOTE_F	bxxxxxxxx	b1xxxxxxxxxx	Transactions homed remotely Filter Transaction whose address resides in a remote MC
OTHER_REQ_F	bxxxxxxxx	bxxxxxxxxxxx1x	Write Request Filter Writeback transactions to the LLC This includes all write transactions -- both Cachable and UC.

Table 2-95. Unit Masks for LLC_LOOKUP (Sheet 2 of 3)

Extension	umask [15:8]	xtra [57:32]	Description
PREF_OR_DMND_REMOTE_F	bxxxxxxxx	bxxx1xxxxxxxx	Remote non-snoop request Filter Non-snoop transactions to the LLC from remote agent
READ_MISS_REM_HOM	b00000001	0x13D9	Remotely HOMed Read Misses
READ_MISS_LOC_HOM	b00000001	0x0BD9	Locally HOMed Read Misses
RFO_MISS	b00000001	0x1BC8	RFO Misses
MISS_ALL	b00000001	0x1FE0	All Misses
READ_OR_SNOOP_REMOTE_MISS_REM_HOM	b00000001	0x1619	Remotely requested Read or Snoop Misses that are Remotely HOMed
DATA_READ_MISS	b00000001	0x1BC1	Data Read Misses
CODE_READ_MISS	b00000001	0x1BD0	Code Read Misses
READ_MISS	b00000001	0x1BD9	Read Misses
I	bxxxxxxx1	bxxxxxxxxxxxx	I State Miss
SF_S	bxxxxxx1x	bxxxxxxxxxxxx	SnoopFilter - S State SF Hit Shared State
SF_E	bxxxxx1xx	bxxxxxxxxxxxx	SnoopFilter - E State SF Hit Exclusive State
SF_H	bxxxx1xxx	bxxxxxxxxxxxx	SnoopFilter - H State SF Hit HitMe State
READ_SF_HIT	b00001110	0x1BD9	Reads that Hit the Snoop Filter
S	bxxx1xxxx	bxxxxxxxxxxxx	S State Hit Shared State
E	bxx1xxxxx	bxxxxxxxxxxxx	E State Hit Exclusive State
M	bx1xxxxxx	bxxxxxxxxxxxx	M State Hit Modified State
F	b1xxxxxxx	bxxxxxxxxxxxx	F State Hit Forward State
REM_HOM	b11111111	0x15DF	Remotely HOMed
READ	b11111111	0x1BD9	Reads
WRITES_AND_OTHER	b11111111	0x1A42	Writes and Other Flows Modified and Non-Modified Writebacks (from IA, UPI, or Memory), ITOM, Streaming Writes, and other IDI Opcodes that are hard to classify
LOC_HOM	b11111111	0x0BDF	Locally HOMed
CODE_READ	b11111111	0x1BD0	Code Reads
DATA_READ_LOCAL	b11111111	0x19C1	Locally Requested Data Reads
REMOTE_SNP	b11111111	0x1C19	Remote Snoops
READ_REMOTE_LOC_HOM	b11111111	0x0A19	Remotely Requested Reads that are Locally HOMed
FLUSH_INV	b11111111	0x1A44	Flush Invalidates
DATA_READ_REMOTE	b11111111	0x1A01	Remotely Requested Data Reads
ALL	b11111111	0x1FFF	All
RFO_LOCAL	b11111111	0x19C8	Locall Requested RFOs
FLUSH_INV_LOCAL	b11111111	0x1844	Locally Requested Flush Invalidates
RFO	b11111111	0x1BC8	RFOs
CODE_READ_REMOTE	b11111111	0x1A10	Remotely Requested Code Reads

Table 2-95. Unit Masks for LLC_LOOKUP (Sheet 3 of 3)

Extension	umask [15:8]	xtra [57:32]	Description
FLUSH_INV_REMOTE	b11111111	0x1A04	Remotely Requested Flush Invalidates
RFO_REMOTE	b11111111	0x1A08	Remote Requested RFOs
ALL_REMOTE	b11111111	0x1E20	All transactions from Remote Agents
READ_LOCAL_LOC_HOM	b11111111	0x09D9	Locally Requested Reads that are Locally HOMed
LLCPREF_LOCAL	b11111111	0x189D	Locally Requested LLC Prefetches
CODE_READ_LOCAL	b11111111	0x19D0	Locally Requested Code Reads
READ_LOCAL_REM_HOM	b11111111	0x11D9	Locally Requested Reads that are Remotely HOMed
DATA_READ	b11111111	0x1BC1	Data Reads

LLC_VICTIMS

- **Title:** Lines Victimized
- **Category:** CACHE Events
- **Event Code:** 0x37
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of lines that were victimized on a fill. This can be filtered by the state that the line was in.
- **NOTE:** Does not include evict cleans

Table 2-96. Unit Masks for LLC_VICTIMS

Extension	umask [15:8]	xtra [57:32]	Description
LOCAL_ONLY	bxxxxxxxx	bxx1xxxxx	Local Only
REMOTE_ONLY	bxxxxxxxx	b1xxxxxxx	Remote Only
REMOTE_M	b00000001	b10000000	Remote - Lines in M State
M_STATE	bxxxxxxx1	bxxxxxxxx	Lines in M state
LOCAL_M	b00000001	b00100000	Local - Lines in M State
E_STATE	bxxxxxx1x	bxxxxxxxx	Lines in E state
REMOTE_E	b00000010	b10000000	Remote - Lines in E State
LOCAL_E	b00000010	b00100000	Local - Lines in E State
LOCAL_S	b00000100	b00100000	Local - Lines in S State
REMOTE_S	b00000100	b10000000	Remote - Lines in S State
S_STATE	bxxxxx1xx	bxxxxxxxx	Lines in S State
ALL	b00001111	b00000000	All Lines Victimized
LOCAL_ALL	b00001111	b00100000	Local - All Lines
REMOTE_ALL	b00001111	b10000000	Remote - All Lines

MISC

- **Title:** Cbo Misc
- **Category:** MISC Events
- **Event Code:** 0x39
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Miscellaneous events in the Cbo.

Table 2-97. Unit Masks for MISC

Extension	umask [15:8]	Description
RSPI_WAS_FSE	bxxxxxxx1	Silent Snoop Eviction Counts the number of times when a Snoop hit in FSE states and triggered a silent eviction. This is useful because this information is lost in the PRE encodings.
WC_ALIASING	bxxxxxx1x	Write Combining Aliasing Counts the number of times that a USWC write (WCIL(F)) transaction hit in the LLC in M state, triggering a WBMtoI followed by the USWC write. This occurs when there is WC aliasing.
RFO_HIT_S	bxxxx1xxx	RFO HitS Number of times that an RFO hit in S state. This is useful for determining if it might be good for a workload to use RspIWB instead of RspSWB.
CV0_PREF_VIC	bxxx1xxxx	CV0 Prefetch Victim
CV0_PREF_MISS	bxx1xxxxx	CV0 Prefetch Miss

OSB

- **Title:** OSB Snoop Broadcast
- **Category:** HA OSB Events
- **Event Code:** 0x55
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Count of OSB snoop broadcasts. Counts by 1 per request causing OSB snoops to be broadcast. Does not count all the snoops generated by OSB.

Table 2-98. Unit Masks for OSB

Extension	umask [15:8]	Description
LOCAL_INVITOE	bxxxxxxx1	Local InvItoE
LOCAL_READ	bxxxxxx1x	Local Rd
REMOTE_READ	bxxxxx1xx	Remote Rd
REMOTE_READINVITOE	bxxxx1xxx	Remote Rd InvItoE
RFO_HITS_SNP_BCAST	bxxx1xxxx	RFO HitS Snoop Broadcast
OFF_PWRHEURISTIC	bxx1xxxxx	Off

PIPE_REJECT

- **Title:** Pipe Rejects
- **Category:** PIPE REJECT Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** More Miscellaneous events in the Cbo.

Table 2-99. Unit Masks for PIPE_REJECT (Sheet 1 of 3)

Extension	umask [15:8]	xtra [57:32]	Description
TORID_MATCH_GO_P	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx1x xxx	
ADEGRCREDIT	bxxxxxxxx	bxxxxxxxxx xx1xxxxxx xxxxxxxxx xxx	
IDX_INPIPE	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxx1xxxx xxx	
SF_WAYS_RES	bxxxxxxxx	bxxxxxxxxx xxxxxxx1x xxxxxxxxx xxx	
ALLRSFWAYS_RES	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx x1xxxxxxx xxx	
HACREDIT	bxxxxxxxx	bxxxxxxxxx 1xxxxxxxx xxxxxxxxx xxx	
BLEGRCREDIT	bxxxxxxxx	bxxxxxxxxx xxx1xxxxx xxxxxxxxx xxx	
VN_BL_NCS	bxxxxxxxx	bxxx1xxxx xxxxxxxxx xxxxxxxxx xxx	
AKEGRCREDIT	bxxxxxxxx	bxxxxxxxxx x1xxxxxxx xxxxxxxxx xxx	
PMM_MEMMODE_TOR_MATCH	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxx1 xxx	
SETMATCHENTRYWSCT	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxx1xxxxx xxx	
TOPA_MATCH	bxxxxxxxx	bxxxxxxxxx xxxx1xxxx xxxxxxxxx xxx	
ONE_FSF_VIC	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx 1xx	
VN_BL_RSP	bxxxxxxxx	bxxxxxx1xx xxxxxxxxx xxxxxxxxx xxx	
VN_BL_NCB	bxxxxxxxx	bxxxx1xxxx xxxxxxxxx xxxxxxxxx xxx	

Table 2-99. Unit Masks for PIPE_REJECT (Sheet 2 of 3)

Extension	umask [15:8]	xtra [57:32]	Description
IRQ_SETMATCH_VICP	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xx1	
PMM_MEMMODE_TORMATCH_MULTI	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xx1xxxxxx xxx	
VN_BL_WB	bxxxxxxxx	bxxxxx1xxx xxxxxxxxx xxxxxxxxx xxx	
IPQ_SETMATCH_VICP	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxx1xx xxx	
VN_AD_REQ	bxxxxxxxx	bxxxxxxxx1 xxxxxxxxx xxxxxxxxx xxx	
ONE_RSP_CON	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxx1xxxx xxx	
ISMQ_SETMATCH_VICP	bxxxxxxxx	bxxxxxxxxx xxxxxxxx1 xxxxxxxxx xxx	
WAY_MATCH	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxx1xxx xxx	
VN_AD_RSP	bxxxxxxxx	bxxxxxxx1x xxxxxxxxx xxxxxxxxx xxx	
IVEGRCREDIT	bxxxxxxxx	bxxxxxxxxx xxx1xxxxx xxxxxxxxx xxx	
RRQ_SETMATCH_VICP	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx 1xxxxxxxx xxx	
LLC_WAYS_RES	bxxxxxxxx	bxxxxxxxxx xxxxxx1xx xxxxxxxxx xxx	
NOTALLOWSNOOP	bxxxxxxxx	bxxxxxxxxx xxxxx1xxx xxxxxxxxx xxx	
FSF_VICP	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx x1x	
RMW_SETMATCH	bxxxxxxx1	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	

Table 2-99. Unit Masks for PIPE_REJECT (Sheet 3 of 3)

Extension	umask [15:8]	xtra [57:32]	Description
GOTRACK_PAMATCH	bxxxxxx1x	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	
GOTRACK_ALLOWSNP	bxxxxx1xx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	
GOTRACK_WAYMATCH	bxxxx1xxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	
GOTRACK_ALLWAYRSV	bxxx1xxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	
IRQ_PMM	bxx1xxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	
PRQ_PMM	bx1xxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	
PTL_INPIPE	b1xxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxxxx xxx	

PMM_MEMMODE_NM_INVITOX

- **Title:**
- **Category:** HA PM MEMMODE Events
- **Event Code:** 0x65
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-100. Unit Masks for PMM_MEMMODE_NM_INVITOX

Extension	umask [15:8]	Description
LOCAL	bxxxxxxx1	
REMOTE	bxxxxxx1x	
SETCONFLICT	bxxxxx1xx	

PMM_MEMMODE_NM_SETCONFLICTS

- **Title:** PMM Memory Mode related events
- **Category:** HA PM MEMMODE Events
- **Event Code:** 0x64
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-101. Unit Masks for PMM_MEMMODE_NM_SETCONFLICTS

Extension	umask [15:8]	Description
SF	bxxxxxxx1	Counts the number of times CHA saw NM Set conflict in SF/LLC NM evictions due to another read to the same near memory set in the SF.
LLC	bxxxxxx1x	Counts the number of times CHA saw NM Set conflict in SF/LLC NM evictions due to another read to the same near memory set in the LLC.
TOR	bxxxxx1xx	Counts the number of times CHA saw NM Set conflict in TOR No Reject in the CHA due to a pending read to the same near memory set in the TOR.

PMM_MEMMODE_NM_SETCONFLICTS2

- **Title:**
- **Category:** HA PM MEMMODE Events
- **Event Code:** 0x70
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-102. Unit Masks for PMM_MEMMODE_NM_SETCONFLICTS2

Extension	umask [15:8]	Description
IODC	bxxxxxxx1	
MEMWR	bxxxxxx1x	
MEMWRNI	bxxxxx1xx	

PMM_QOS

- **Title:**
- **Category:** HA PMM QOS Events
- **Event Code:** 0x66
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-103. Unit Masks for PMM_QOS

Extension	umask [15:8]	Description
SLOW_INSERT	bxxxxxxx1	
DDR4_FAST_INSERT	bxxxxxx1x	
THROTTLE	bxxxxx1xx	
REJ_IRQ	bxxxx1xxx	
THROTTLE_PRQ	bxxx1xxxx	
THROTTLE_IRQ	bxx1xxxxx	
SLOWTORQ_SKIP	bx1xxxxxx	

PMM_QOS_OCCUPANCY

- **Title:**
- **Category:** HA PMM QOS Events
- **Event Code:** 0x67
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-104. Unit Masks for PMM_QOS_OCCUPANCY

Extension	umask [15:8]	Description
DDR_SLOW_FIFO	bxxxxxxx1	count # of SLOW TOR Request inserted to ha_pmm_tor_req_fifo
DDR_FAST_FIFO	bxxxxxx1x	count # of FAST TOR Request inserted to ha_tor_req_fifo

READ_NO_CREDITS

- **Title:** CHA iMC CHNx READ Credits Empty
- **Category:** MC Credit and Traffic Events
- **Event Code:** 0x58
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of times when there are no credits available for sending reads from the CHA into the iMC. In order to send reads into the memory controller, the HA must first acquire a credit for the iMC's AD Ingress queue.

Table 2-105. Unit Masks for READ_NO_CREDITS (Sheet 1 of 2)

Extension	umask [15:8]	xtra [57:32]	Description
MC11	bxxxxxxxx	bxxxx1xxx	MC11 Filter for memory controller 11 only.
MC13	bxxxxxxxx	bxx1xxxxx	MC13 Filter for memory controller 13 only.
MC9	bxxxxxxxx	bxxxxxx1x	MC9 Filter for memory controller 9 only.
MC10	bxxxxxxxx	bxxxxx1xx	MC10 Filter for memory controller 10 only.
MC8	bxxxxxxxx	bxxxxxxx1	MC8 Filter for memory controller 8 only.
MC12	bxxxxxxxx	bxxx1xxxx	MC12 Filter for memory controller 12 only.
MC0	bxxxxxxx1	bxxxxxxxx	MC0 Filter for memory controller 0 only.
MC1	bxxxxxx1x	bxxxxxxxx	MC1 Filter for memory controller 1 only.
MC2	bxxxxx1xx	bxxxxxxxx	MC2 Filter for memory controller 2 only.
MC3	bxxxx1xxx	bxxxxxxxx	MC3 Filter for memory controller 3 only.
MC4	bxxx1xxxx	bxxxxxxxx	MC4 Filter for memory controller 4 only.
MC5	bxx1xxxxx	bxxxxxxxx	MC5 Filter for memory controller 5 only.

Table 2-105. Unit Masks for READ_NO_CREDITS (Sheet 2 of 2)

Extension	umask [15:8]	xtra [57:32]	Description
MC6	bx1xxxxxx	bxxxxxxxx	MC6 Filter for memory controller 6 only.
MC7	b1xxxxxxx	bxxxxxxxx	MC7 Filter for memory controller 7 only.

REQUESTS

- **Title:** HA Read and Write Requests
- **Category:** HA REQUEST Events
- **Event Code:** 0x50
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the total number of read requests made into the Home Agent. Reads include all read opcodes (including RFO). Writes include all writes (streaming, evictions, HitM, etc).

Table 2-106. Unit Masks for REQUESTS

Extension	umask [15:8]	Description
READS_LOCAL	bxxxxxxx1	Reads Local Local read requests that miss the SF/LLC and are sent to the CHA's Home Agent
READS_REMOTE	bxxxxxx1x	Reads Remote Remote read requests sent to the CHA's Home Agent
READS	b00000011	Reads Local read requests that miss the SF/LLC and remote read requests sent to the CHA's Home Agent
WRITES_LOCAL	bxxxxx1xx	Writes Local Local write requests that miss the SF/LLC and are sent to the CHA's Home Agent
WRITES_REMOTE	bxxxx1xxx	Writes Remote Remote write requests sent to the CHA's Home Agent
WRITES	b00001100	Writes Local write requests that miss the SF/LLC and remote write requests sent to the CHA's Home Agent
INVITOE_LOCAL	bxxx1xxxx	InvalItoE Local Local InvItoE requests (exclusive ownership of a cache line without receiving data) that miss the SF/LLC and are sent to the CHA's home agent
INVITOE_REMOTE	bxx1xxxxx	InvalItoE Remote Remote InvItoE requests (exclusive ownership of a cache line without receiving data) sent to the CHA's home agent
INVITOE	b00110000	InvalItoE InvItoE requests (exclusive ownership of a cache line without receiving data) sent to the CHA's home agent

RxC_INSERTS

- **Title:** Ingress (from CMS) Allocations
- **Category:** INGRESS Events
- **Event Code:** 0x13
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts number of allocations per cycle into the specified Ingress queue.
- **NOTE:** IRQ_REJECTED should not be Ored with the other umasks.

Table 2-107. Unit Masks for RxC_INSERTS

Extension	umask [15:8]	Description
IRQ	bxxxxxxx1	IRQ
IRQ_REJ	bxxxxxx1x	IRQ Rejected
IPQ	bxxxxx1xx	IPQ
PRQ	bxxx1xxxx	PRQ
PRQ_REJ	bxx1xxxxx	PRQ Rejected
RRQ	bx1xxxxxx	RRQ
WBQ	b1xxxxxxx	WBQ

RxC_IPQ0_REJECT

- **Title:** IPQ Requests (from CMS) Rejected - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x22
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-108. Unit Masks for RxC_IPQ0_REJECT

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_IPQ1_REJECT

- **Title:** IPQ Requests (from CMS) Rejected - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x23
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-109. Unit Masks for RxC_IPQ1_REJECT

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the IPQ0 Reject counter was true
HA	bxxxxxx1x	HA
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC OR SF Way Way conflict with another request that caused the reject
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

RxC_IRQ0_REJECT

- **Title:** IRQ Requests (from CMS) Rejected - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-110. Unit Masks for RxC_IRQ0_REJECT

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_IRQ1_REJECT

- **Title:** IRQ Requests (from CMS) Rejected - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x19
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-111. Unit Masks for RxC_IRQ1_REJECT

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the IRQ0 Reject counter was true
HA	bxxxxxx1x	HA
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC or SF Way Way conflict with another request that caused the reject
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

RxC_ISMQ0_REJECT

- **Title:** ISMQ Rejects - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x24
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the ISMQ had to retry. Transaction pass through the ISMQ as responses for requests that already exist in the Cbo. Some examples include: when data is returned or when snoop responses come back from the cores.

Table 2-112. Unit Masks for RxC_ISMQ0_REJECT

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_ISMQ0_RETRY

- **Title:** ISMQ Retries - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x2c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the ISMQ had to retry. Transaction pass through the ISMQ as responses for requests that already exist in the Cbo. Some examples include: when data is returned or when snoop responses come back from the cores.

Table 2-113. Unit Masks for RxC_ISMQ0_RETRY

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_ISMQ1_REJECT

- **Title:** ISMQ Rejects - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x25
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the ISMQ had to retry. Transaction pass through the ISMQ as responses for requests that already exist in the Cbo. Some examples include: when data is returned or when snoop responses come back from the cores.

Table 2-114. Unit Masks for RxC_ISMQ1_REJECT

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the ISMQ0 Reject counter was true
HA	bxxxxxx1x	HA

RxC_ISMQ1_RETRY

- **Title:** ISMQ Retries - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x2d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the ISMQ had to retry. Transaction pass through the ISMQ as responses for requests that already exist in the Cbo. Some examples include: when data is returned or when snoop responses come back from the cores.

Table 2-115. Unit Masks for RxC_ISMQ1_RETRY

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the ISMQ0 Reject counter was true
HA	bxxxxxx1x	HA

RxC_OCCUPANCY

- **Title:** Ingress (from CMS) Occupancy
- **Category:** INGRESS Events
- **Event Code:** 0x11
- **Max. Inc/Cyc:.** 20

Register Restrictions: 0

- **Definition:** Counts number of entries in the specified Ingress queue in each cycle.
- **NOTE:** IRQ_REJECTED should not be Ored with the other umasks.

Table 2-116. Unit Masks for RxC_OCCUPANCY

Extension	umask [15:8]	Description
IRQ	b00000001	IRQ
IPQ	b00000100	IPQ
RRQ	b01000000	RRQ
WBQ	b10000000	WBQ

RxC_OTHER0_RETRY

- **Title:** Other Retries - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x2e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Retry Queue Inserts of Transactions that were already in another Retry Q (sub-events encode the reason for the next reject)

Table 2-117. Unit Masks for RxC_OTHER0_RETRY

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_OTHER1_RETRY

- **Title:** Other Retries - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x2f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Retry Queue Inserts of Transactions that were already in another Retry Q (sub-events encode the reason for the next reject)

Table 2-118. Unit Masks for RxC_OTHER1_RETRY

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the Other0 Reject counter was true
HA	bxxxxxx1x	HA
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC OR SF Way Way conflict with another request that caused the reject
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

RxC_PRQ0_REJECT

- **Title:** PRQ Requests (from CMS) Rejected - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x20
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-119. Unit Masks for RxC_PRQ0_REJECT

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_PRQ1_REJECT

- **Title:** PRQ Requests (from CMS) Rejected - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x21
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-120. Unit Masks for RxC_PRQ1_REJECT

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the PRQ0 Reject counter was true
HA	bxxxxxx1x	HA
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC OR SF Way Way conflict with another request that caused the reject
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

RxC_REQ_Q0_RETRY

- **Title:** Request Queue Retries - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x2a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** "REQUESTQ" includes: IRQ, PRQ, IPQ, RRQ, WBQ (everything except for ISMQ)

Table 2-121. Unit Masks for RxC_REQ_Q0_RETRY

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_REQ_Q1_RETRY

- **Title:** Request Queue Retries - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x2b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** "REQUESTQ" includes: IRQ, PRQ, IPQ, RRQ, WBQ (everything except for ISMQ)

Table 2-122. Unit Masks for RxC_REQ_Q1_RETRY

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the WBQ0 Reject counter was true
HA	bxxxxxx1x	HA
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC OR SF Way Way conflict with another request that caused the reject

Table 2-122. Unit Masks for RxC_REQ_Q1_RETRY

Extension	umask [15:8]	Description
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

RxC_RRQ0_REJECT

- **Title:** RRQ Rejects - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x26
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the RRQ (Remote Response Queue) had to retry.

Table 2-123. Unit Masks for RxC_RRQ0_REJECT

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_RRQ1_REJECT

- **Title:** RRQ Rejects - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x27
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the RRQ (Remote Response Queue) had to retry.

Table 2-124. Unit Masks for RxC_RRQ1_REJECT (Sheet 1 of 2)

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the RRQ0 Reject counter was true
HA	bxxxxxx1x	HA

Table 2-124. Unit Masks for RxC_RRQ1_REJECT (Sheet 2 of 2)

Extension	umask [15:8]	Description
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC OR SF Way Way conflict with another request that caused the reject
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

RxC_WBQ0_REJECT

- **Title:** WBQ Rejects - Set 0
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x28
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the WBQ (Writeback Queue) had to retry.

Table 2-125. Unit Masks for RxC_WBQ0_REJECT

Extension	umask [15:8]	Description
AD_REQ_VN0	bxxxxxxx1	AD REQ on VN0 No AD VN0 credit for generating a request
AD_RSP_VN0	bxxxxxx1x	AD RSP on VN0 No AD VN0 credit for generating a response
BL_RSP_VN0	bxxxxx1xx	BL RSP on VN0 No BL VN0 credit for generating a response
BL_WB_VN0	bxxxx1xxx	BL WB on VN0 No BL VN0 credit for generating a writeback
BL_NCB_VN0	bxxx1xxxx	BL NCB on VN0 No BL VN0 credit for NCB
BL_NCS_VN0	bxx1xxxxx	BL NCS on VN0 No BL VN0 credit for NCS
AK_NON_UPI	bx1xxxxxx	Non UPI AK Request Can't inject AK ring message
IV_NON_UPI	b1xxxxxxx	Non UPI IV Request Can't inject IV ring message

RxC_WBQ1_REJECT

- **Title:** WBQ Rejects - Set 1
- **Category:** INGRESS_RETRY Events
- **Event Code:** 0x29
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a transaction flowing through the WBQ (Writeback Queue) had to retry.

Table 2-126. Unit Masks for RxC_WBQ1_REJECT

Extension	umask [15:8]	Description
ANY0	bxxxxxxx1	ANY0 Any condition listed in the WBQ0 Reject counter was true
HA	bxxxxxx1x	HA
LLC_VICTIM	bxxxxx1xx	LLC Victim
SF_VICTIM	bxxxx1xxx	SF Victim Requests did not generate Snoop filter victim
VICTIM	bxxx1xxxx	Victim
LLC_OR_SF_WAY	bxx1xxxxx	LLC OR SF Way Way conflict with another request that caused the reject
ALLOW_SNP	bx1xxxxxx	Allow Snoop
PA_MATCH	b1xxxxxxx	PhyAddr Match Address match with an outstanding request that was rejected.

SF_EVICTION

- **Title:** Snoop Filter Capacity Evictions
- **Category:** CACHE Events
- **Event Code:** 0x3d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of times a snoop filter entry was evicted, due to lack of space, and replaced with a new entry.
- **NOTE:** For cache lines this CHA has some responsibility for managing, the snoop filter tracks their state in the Cores. Does not count clean evictions

Table 2-127. Unit Masks for SF_EVICTION

Extension	umask [15:8]	Description
M_STATE	bxxxxxxx1	M state
E_STATE	bxxxxxx1x	E state
S_STATE	bxxxxx1xx	S state

SNOOPS_SENT

- **Title:** Snoops Sent
- **Category:** HA REQUEST Events
- **Event Code:** 0x51
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of snoops issued by the HA.

Table 2-128. Unit Masks for SNOOPS_SENT

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All
LOCAL	bxxxxx1xx	Snoops sent for Local Requests Counts the number of broadcast or directed snoops issued by the HA responding to local requests
REMOTE	bxxxx1xxx	Snoops sent for Remote Requests Counts the number of broadcast or directed snoops issued by the HA responding to remote requests
BCST_LOCAL	bxxx1xxxx	Broadcast snoops for Local Requests Counts the number of broadcast snoops issued by the HA responding to local requests
BCST_REMOTE	bxx1xxxxx	Broadcast snoops for Remote Requests Counts the number of broadcast snoops issued by the HA responding to remote requests
DIRECT_LOCAL	bx1xxxxxx	Directed snoops for Local Requests Counts the number of directed snoops issued by the HA responding to local requests
DIRECT_REMOTE	b1xxxxxxx	Directed snoops for Remote Requests Counts the number of directed snoops issued by the HA responding to remote requests

SNOOP_RESP

- **Title:** Snoop Responses Received
- **Category:** HA SNOOP RESPONSE Events
- **Event Code:** 0x5c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the total number of RspI snoop responses received. Whenever a snoops are issued, one or more snoop responses will be returned depending on the topology of the system. In systems larger than 2s, when multiple snoops are returned this will count all the snoops that are received. For example, if 3 snoops were issued and returned RspI, RspS, and RspSFwd; then each of these sub-events would increment by 1.

Table 2-129. Unit Masks for SNOOP_RESP (Sheet 1 of 2)

Extension	umask [15:8]	Description
RSPI	bxxxxxxx1	RspI Filters for snoops responses of RspI. RspI is returned when the remote cache does not have the data, or when the remote cache silently evicts data (such as when an RFO hits non-modified data).
RSPS	bxxxxxx1x	RspS Filters for snoop responses of RspS. RspS is returned when a remote cache has data but is not forwarding it. It is a way to let the requesting socket know that it cannot allocate the data in E state. No data is sent with S RspS.
RSPIFWD	bxxxxx1xx	RspIFwd Filters for snoop responses of RspIFwd. This is returned when a remote caching agent forwards data and the requesting agent is able to acquire the data in E or M states. This is commonly returned with RFO transactions. It can be either a HitM or a HitFE.

Table 2-129. Unit Masks for SNOOP_RESP (Sheet 2 of 2)

Extension	umask [15:8]	Description
RSPSFWD	bxxxx1xxx	RspSFwd Filters for a snoop response of RspSFwd. This is returned when a remote caching agent forwards data but holds on to its currentl copy. This is common for data and code reads that hit in a remote socket in E or F state.
RSPWB	bxxx1xxxx	Rsp*WB Filters for a snoop response of RspIWB or RspSWB. This is returned when a non-RFO request hits in M state. Data and Code Reads can return either RspIWB or RspSWB depending on how the system has been configured. InvItoE transactions will also return RspIWB because they must acquire ownership.
RSPFWDWB	bxx1xxxxx	Rsp*Fwd*WB Filters for a snoop response of Rsp*Fwd*WB. This snoop response is only used in 4s systems. It is used when a snoop HITM's in a remote caching agent and it directly forwards data to a requestor, and simultaneously returns data to the home to be written back to memory.
RSPCNFLCT	bx1xxxxxx	RSPCNFLCT* Filters for snoops responses of RspConflict. This is returned when a snoop finds an existing outstanding transaction in a remote caching agent when it CAMs that caching agent. This triggers conflict resolution hardware. This covers both RspCnflct and RspCnflctWbI.
RSPFWD	b1xxxxxxx	RspFwd Filters for a snoop response of RspFwd to a CA request. This snoop response is only possible for RdCur when a snoop HITM/E in a remote caching agent and it directly forwards data to a requestor without changing the requestor's cache line state.

SNOOP_RESP_LOCAL

- **Title:** Snoop Responses Received Local
- **Category:** HA SNOOP RESPONSE Events
- **Event Code:** 0x5d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of snoop responses received for a Local request

Table 2-130. Unit Masks for SNOOP_RESP_LOCAL (Sheet 1 of 2)

Extension	umask [15:8]	Description
RSPI	bxxxxxxx1	RspI Filters for snoops responses of RspI to local CA requests. RspI is returned when the remote cache does not have the data, or when the remote cache silently evicts data (such as when an RFO hits non-modified data).
RSPS	bxxxxxx1x	RspS Filters for snoop responses of RspS to local CA requests. RspS is returned when a remote cache has data but is not forwarding it. It is a way to let the requesting socket know that it cannot allocate the data in E state. No data is sent with S RspS.
RSPIFWD	bxxxxx1xx	RspIFwd Filters for snoop responses of RspIFwd to local CA requests. This is returned when a remote caching agent forwards data and the requesting agent is able to acquire the data in E or M states. This is commonly returned with RFO transactions. It can be either a HitM or a HitFE.

Table 2-130. Unit Masks for SNOOP_RESP_LOCAL (Sheet 2 of 2)

Extension	umask [15:8]	Description
RSPSFWD	bxxxx1xxx	RspSFwd Filters for a snoop response of RspSFwd to local CA requests. This is returned when a remote caching agent forwards data but holds on to its currentl copy. This is common for data and code reads that hit in a remote socket in E or F state.
RSPWB	bxxx1xxxx	Rsp*WB Filters for a snoop response of RspIWB or RspSWB to local CA requests. This is returned when a non-RFO request hits in M state. Data and Code Reads can return either RspIWB or RspSWB depending on how the system has been configured. InvItoE transactions will also return RspIWB because they must acquire ownership.
RSPFWDWB	bxx1xxxxx	Rsp*FWD*WB Filters for a snoop response of Rsp*Fwd*WB to local CA requests. This snoop response is only used in 4s systems. It is used when a snoop HITM's in a remote caching agent and it directly forwards data to a requestor, and simultaneously returns data to the home to be written back to memory.
RSPCNFLCT	bx1xxxxxx	RspCnflct Filters for snoops responses of RspConflict to local CA requests. This is returned when a snoop finds an existing outstanding transaction in a remote caching agent when it CAMs that caching agent. This triggers conflict resolution hardware. This covers both RspCnflct and RspCnflctWbI.
RSPFWD	b1xxxxxxx	RspFwd Filters for a snoop response of RspFwd to local CA requests. This snoop response is only possible for RdCur when a snoop HITM/E in a remote caching agent and it directly forwards data to a requestor without changing the requestor's cache line state.

SNOOP_RSP_MISC

- **Title:** Misc Snoop Responses Received
- **Category:** CBO SNOOP RESPONSE Events
- **Event Code:** 0x6b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-131. Unit Masks for SNOOP_RSP_MISC

Extension	umask [15:8]	Description
MTOI_RSPIFWDM	bxxxxxxx1	MtoI RspIFwdM
MTOI_RSPDATAM	bxxxxxx1x	MtoI RspIDataM
RSPIFWDMPTL_HITSF	bxxxxx1xx	RspIFwdPtl Hit SF
RSPIFWDMPTL_HITLLC	bxxxx1xxx	RspIFwdPtl Hit LLC
PULLDATAPTL_HITSF	bxxx1xxxx	Pull Data Partial - Hit SF
PULLDATAPTL_HITLLC	bxx1xxxxx	Pull Data Partial - Hit LLC

TOR_INSERTS

- **Title:** TOR Inserts
- **Category:** TOR Events
- **Event Code:** 0x35
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of entries successfuly inserted into the TOR that match qualifications specified by the subevent. Does not include addressless requests such as locks and interrupts.
- **NOTE:** HW does not strictly OR each subevent. The equation is roughly (IRQ|EVICT|PRQ|IPQ|RRQ|WBQ) & (HIT|MISS). Meaning it is necessary to set one of the queue bits before one can measure .HIT or .MISS.

Table 2-132. Unit Masks for TOR_INSERTS (Sheet 1 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
MATCH_OPC	bxxxxxxxx	bxxxxXXXX XXXXXXXx1 xxxxxxxxx	Match the Opcode in b[29:19] of the extended umask field
HIT	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxx1	Just Hits
MMCFG	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx x1xxxxx	MMCFG Access
NOT_NEARMEM	bxxxxxxxx	bxx1xxxxxx xxxxxxxxx xxxxxxx	Just NotNearMem
MISS	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxx1x	Just Misses
ISOC	bxxxxxxxx	b1xxxxxxxx xxxxxxxxx xxxxxxx	Just ISOC
NONCOH	bxxxxxxxx	bx1xxxxxxx xxxxxxxxx xxxxxxx	Just NonCoherent
LOCAL_TGT	bxxxxxxxx	bxxxxxxxxx xxxxxxxx1 xxxxxxx	Just Local Targets
PREMORPH_OPC	bxxxxxxxx	bxxxxXXXX XXXXXXX1x xxxxxxxxx	Match the PreMorphed Opcode in b[29:19] of the extended umask field
PMM	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxx1xxx	PMM Access
DDR	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxx1xx	DDR Access
HBM	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xx1xxxx	HBM Access
REMOTE_TGT	bxxxxxxxx	bxxxxxxxxx xxxxxxxx1x xxxxxxx	Just Remote Targets
NEARMEM	bxxxxxxxx	bxxx1xxxxx xxxxxxxxx xxxxxxx	Just NearMem

Table 2-132. Unit Masks for TOR_INSERTS (Sheet 2 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
MMIO	bxxxxxxxx	bxxxxxxxxx xxxxxxxxxx 1xxxxxx	MMIO Access
IA_MISS_DRD_LOCAL_DDR	b00000001	0xC81686	DRds issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed locally
IA_ITOMCACHENEAR	b00000001	0xCD47FF	ItoMCacheNears issued by iA Cores
IA_MISS_REMOTE_WCIL_PMM	b00000001	0xC86F0A	WCiLs issued by iA Cores targeting PMM that missed the LLC - HOMed remotely
IA_MISS_DRD_REMOTE_DDR	b00000001	0xC81706	DRds issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed remotely
IA_LLCPREFDATA	b00000001	0xCCD7FF	LLCPrefData issued by iA Cores
IA_DRD_PREF	b00000001	0xC897FF	DRd_Prefs issued by iA Cores
IA_MISS_RFO_PREF_LOCAL	b00000001	0xC886FE	RFO_Prefs issued by iA Cores that Missed the LLC - HOMed locally
IA_MISS_WCILF_DDR	b00000001	0xC86786	WCiLFs issued by iA Cores targeting DDR that missed the LLC
IA_MISS_DRD_OPT_PREF	b00000001	0xC8A7FE	DRd_Opt_Prefs issued by iA Cores that missed the LLC
IA_HIT_LLCPREFDATA	b00000001	0xCCD7FD	LLCPrefData issued by iA Cores that hit the LLC
IA_MISS_REMOTE_WCIL_DDR	b00000001	0xC86F06	WCiLs issued by iA Cores targeting DDR that missed the LLC - HOMed remotely
IA_LLCPREFCODE	b00000001	0xCCCFFF	LLCPrefCode issued by iA Cores
IA_MISS_DRD_REMOTE	b00000001	0xC8177E	DRds issued by iA Cores that Missed the LLC - HOMed remotely
IA_DRDPTE	b00000001	0xC837FF	DRdPte issued by iA Cores due to a page walk
IA_MISS_CRD_REMOTE	b00000001	0xC80F7E	CRd issued by iA Cores that Missed the LLC - HOMed remotely
IA_MISS_WCIL_PMM	b00000001	0xC86F8A	WCiLs issued by iA Cores targeting PMM that missed the LLC
IA_MISS_CRD_PREF	b00000001	0xC88FFE	CRd_Prefs issued by iA Cores that Missed the LLC
IA_MISS_DRD_PREF	b00000001	0xC897FE	DRd_Prefs issued by iA Cores that Missed the LLC
IA_HIT_DRD_OPT_PREF	b00000001	0xC8A7FD	DRd_Opt_Prefs issued by iA Cores that hit the LLC
IA_MISS_DRD_PMM	b00000001	0xC8178A	DRds issued by iA Cores targeting PMM Mem that Missed the LLC
IA_MISS	b00000001	0xC001FE	All requests from iA Cores that Missed the LLC
IRQ_IA	bxxxxxxx1	bxxxxxxxxx xxxxxxxxxx xxxxxxx	IRQ - iA From an iA Core
IA_MISS_DRD_PREF_REMOTE_PMM	b00000001	0xC8970A	DRd_Prefs issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed remotely
IA_MISS_RFO_PREF	b00000001	0xC887FE	RFO_Prefs issued by iA Cores that Missed the LLC
IA_MISS_DRD_PREF_PMM	b00000001	0xC8978A	DRd_Prefs issued by iA Cores targeting PMM Mem that Missed the LLC
LOC_IA	b00000001	0xC000FF	All from Local iA All locally initiated requests from iA Cores
IA_HIT_DRD_PREF	b00000001	0xC897FD	DRd_Prefs issued by iA Cores that Hit the LLC
IA_MISS_WCIL	b00000001	0xC86FFE	WCiLs issued by iA Cores that Missed the LLC
IA_MISS_DRD_OPT	b00000001	0xC827FE	DRd_Opt issued by iA Cores that missed the LLC
IA_DRD	b00000001	0xC817FF	DRds issued by iA Cores

Table 2-132. Unit Masks for TOR_INSERTS (Sheet 3 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
IA_MISS_LOCAL_WCIL_DDR	b00000001	0xC86E86	WCiLs issued by iA Cores targeting DDR that missed the LLC - HOMed locally
IA_MISS_RFO_REMOTE	b00000001	0xC8077E	RFOs issued by iA Cores that Missed the LLC - HOMed remotely
IA_MISS_WCIL_DDR	b00000001	0xC86F86	WCiLs issued by iA Cores targeting DDR that missed the LLC
IA_MISS_LLCPREFRFO	b00000001	0xCCC7FE	LLCPrefRFO issued by iA Cores that missed the LLC
IA_MISS_DRDPTE	b00000001	0xC837FE	DRdPte issued by iA Cores due to a page walk that missed the LLC
IA_MISS_CRD_PREF_LOCAL	b00000001	0xC88EFE	CRd_Prefs issued by iA Cores that Missed the LLC - HOMed locally
IA_ITOM	b00000001	0xCC47FF	ItoMs issued by iA Cores
IA_HIT_DRD	b00000001	0xC817FD	DRds issued by iA Cores that Hit the LLC
IA_HIT	b00000001	0xC001FD	All requests from iA Cores that Hit the LLC
IA_MISS_WIL	b00000001	0xC87FDE	WiLs issued by iA Cores that Missed LLC
IA_HIT_LLCPREFRFO	b00000001	0xCCC7FD	LLCPrefRFO issued by iA Cores that hit the LLC
IA_CLFLUSH	b00000001	0xC8C7FF	CLFlushes issued by iA Cores
IA_LLCPREFRFO	b00000001	0xCCC7FF	LLCPrefRFO issued by iA Cores
IA_MISS_DRD_PREF_DDR	b00000001	0xC89786	DRd_Prefs issued by iA Cores targeting DDR Mem that Missed the LLC
IA_DRD_OPT	b00000001	0xC827FF	DRd_Opts issued by iA Cores
IA_HIT_RFO	b00000001	0xC807FD	RFOs issued by iA Cores that Hit the LLC
IA_MISS_UCRDF	b00000001	0xC877DE	UCRdFs issued by iA Cores that Missed LLC
IA_MISS_LLCPREFCODE	b00000001	0xCCCFFE	LLCPrefCode issued by iA Cores that missed the LLC
IA_HIT_ITOM	b00000001	0xCC47FD	ItoMs issued by iA Cores that Hit LLC
IA	b00000001	0xC001FF	All requests from iA Cores
IA_WCILF	b00000001	0xC867FF	WCiLF issued by iA Cores
IA_MISS_RFO	b00000001	0xC807FE	RFOs issued by iA Cores that Missed the LLC
IA_MISS_WCILF_PMM	b00000001	0xC8678A	WCiLFs issued by iA Cores targeting PMM that missed the LLC
IA_MISS_RFO_PREF_REMOTE	b00000001	0xC8877E	RFO_Prefs issued by iA Cores that Missed the LLC - HOMed remotely
IA_RFO_PREF	b00000001	0xC887FF	RFO_Prefs issued by iA Cores
IA_MISS_REMOTE_WCILF_DDR	b00000001	0xC86706	WCiLFs issued by iA Cores targeting DDR that missed the LLC - HOMed remotely
IA_MISS_DRD_PREF_LOCAL_DDR	b00000001	0xC89686	DRd_Prefs issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed locally
IA_MISS_DRD_PREF_LOCAL_PMM	b00000001	0xC8968A	DRd_Prefs issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed locally
IA_WCIL	b00000001	0xC86FFF	WCiLs issued by iA Cores
IA_MISS_DRD_REMOTE_PMM	b00000001	0xC8170A	DRds issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed remotely
IA_DRD_OPT_PREF	b00000001	0xC8A7FF	DRd_Opt_Prefs issued by iA Cores
IA_HIT_DRD_OPT	b00000001	0xC827FD	DRd_Opts issued by iA Cores that hit the LLC
IA_HIT_DRDPTE	b00000001	0xC837FD	DRdPte issued by iA Cores due to a page walk that hit the LLC
IA_HIT_RFO_PREF	b00000001	0xC887FD	RFO_Prefs issued by iA Cores that Hit the LLC

Table 2-132. Unit Masks for TOR_INSERTS (Sheet 4 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
IA_MISS_DRD_PREF_REMOTE_DDR	b00000001	0xC89706	DRd_Prefs issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed remotely
IA_SPECITOM	b00000001	0xCC57FF	SpecItoMs issued by iA Cores
IA_MISS_DRD_LOCAL	b00000001	0xC816FE	DRds issued by iA Cores that Missed the LLC - HOMed locally
IA_WBMTOI	b00000001	0xCC27FF	WbMtoIs issued by iA Cores
IA_MISS_CRD_LOCAL	b00000001	0xC80EFE	CRd issued by iA Cores that Missed the LLC - HOMed locally
IA_HIT_CRD_PREF	b00000001	0xC88FFD	CRd_Prefs issued by iA Cores that hit the LLC
IA_MISS_RFO_LOCAL	b00000001	0xC806FE	RFOs issued by iA Cores that Missed the LLC - HOMed locally
IA_MISS_DRD_LOCAL_PMM	b00000001	0xC8168A	DRds issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed locally
IA_RFO	b00000001	0xC807FF	RFOs issued by iA Cores
IA_MISS_WCILF	b00000001	0xC867FE	WCiLF issued by iA Cores that Missed the LLC
IA_HIT_CRD	b00000001	0xC80FFD	CRds issued by iA Cores that Hit the LLC
IA_MISS_LOCAL_WCILF_DDR	b00000001	0xC86686	WCiLFs issued by iA Cores targeting DDR that missed the LLC - HOMed locally
IA_CLFLUSHOPT	b00000001	0xC8D7FF	CLFlushOpts issued by iA Cores
IA_MISS_DRD	b00000001	0xC817FE	DRds issued by iA Cores that Missed the LLC
IA_MISS_CRD_PREF_REMOTE	b00000001	0xC88F7E	CRd_Prefs issued by iA Cores that Missed the LLC - HOMed remotely
IA_MISS_DRD_DDR	b00000001	0xC81786	DRds issued by iA Cores targeting DDR Mem that Missed the LLC
IA_MISS_CRD	b00000001	0xC80FFE	CRds issued by iA Cores that Missed the LLC
IA_HIT_LLCPREFCODE	b00000001	0xCCCFFD	LLCPrefCode issued by iA Cores that hit the LLC
IA_MISS_LOCAL_WCILF_PMM	b00000001	0xC8668A	WCiLFs issued by iA Cores targeting PMM that missed the LLC - HOMed locally
IA_MISS_REMOTE_WCILF_PMM	b00000001	0xC8670A	WCiLFs issued by iA Cores targeting PMM that missed the LLC - HOMed remotely
IA_MISS_LOCAL_WCIL_PMM	b00000001	0xC86E8A	WCiLs issued by iA Cores targeting PMM that missed the LLC - HOMed locally
IA_MISS_ITOM	b00000001	0xCC47FE	ItoMs issued by iA Cores that Missed LLC
IA_CRD	b00000001	0xC80FFF	CRDs issued by iA Cores
IA_MISS_LLCPREFDATA	b00000001	0xCCD7FE	LLCPrefData issued by iA Cores that missed the LLC
EVICT	bxxxxxx1x	bxxxxxxxxx xxxxxxxxx xxxxxxx	SF/LLC Evictions TOR allocation occurred as a result of SF/LLC evictions (came from the ISMQ)
IO_MISS_PCIRDCUR	b00000100	0xC8F3FE	PCIRdCurs issued by IO Devices that missed the LLC
IO_HIT_ITOMCACHENEAR	b00000100	0xCD43FD	ItoMCacheNears, indicating a partial write request, from IO Devices that hit the LLC
IO_MISS_ITOM	b00000100	0xCC43FE	ItoMs issued by IO Devices that missed the LLC
PRQ_IOSF	bxxxxx1xx	bxxxxxxxxx xxxxxxxxx xxxxxxx	PRQ - IOSF From a PCIe Device
IO_RFO	b00000100	0xC803FF	RFOs issued by IO Devices
LOC_IO	b00000100	0xC000FF	All from Local IO All locally generated IO traffic

Table 2-132. Unit Masks for TOR_INSERTS (Sheet 5 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
IO_WBMTOI	b00000100	0xCC23FF	WbMtoIs issued by IO Devices
IO_HIT_PCIRDCUR	b00000100	0xC8F3FD	PCIRdCurs issued by IO Devices that hit the LLC
IO_HIT_RFO	b00000100	0xC803FD	RFOs issued by IO Devices that hit the LLC
IO_MISS_RFO	b00000100	0xC803FE	RFOs issued by IO Devices that missed the LLC
IO	b00000100	0xC001FF	All requests from IO Devices
IO_MISS_ITOMCACHENEAR	b00000100	0xCD43FE	ItoMCacheNears, indicating a partial write request, from IO Devices that missed the LLC
IO_HIT_ITOM	b00000100	0xCC43FD	ItoMs issued by IO Devices that Hit the LLC
IO_HIT	b00000100	0xC001FD	All requests from IO Devices that hit the LLC
IO_ITOM	b00000100	0xCC43FF	ItoMs issued by IO Devices
IO_CLFLUSH	b00000100	0xC8C3FF	CLFlushes issued by IO Devices
IO_PCIRDCUR	b00000100	0xC8F3FF	PCIRdCurs issued by IO Devices
IO_MISS	b00000100	0xC001FE	All requests from IO Devices that missed the LLC
IO_ITOMCACHENEAR	b00000100	0xCD43FF	ItoMCacheNears, indicating a partial write request, from IO Devices
LOC_ALL	b00000101	0xC000FF	All from Local iA and IO All locally initiated requests
IPQ	bxxxx1xxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	IPQ
IRQ_NON_IA	bxxx1xxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	IRQ - Non iA
PRQ_NON_IOSF	bxx1xxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	PRQ - Non IOSF
RRQ	bx1xxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	RRQ
WBQ	b1xxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	WBQ
ALL	b11111111	0xC001FF	All

TOR_OCCUPANCY

- **Title:** TOR Occupancy
- **Category:** TOR Events
- **Event Code:** 0x36
- **Max. Inc/Cyc:.** 20

Register Restrictions: 0

- **Definition:** For each cycle, this event accumulates the number of valid entries in the TOR that match qualifications specified by the subevent. Does not include address-less requests such as locks and interrupts.
- **NOTE:** HW does not strictly OR each subevent. The equation is roughly (IRQ|EVICT|PRQ|IPQ|RRQ|WBQ) & (HIT|MISS). Meaing it is necessary to set one of the queue bits before one can measure .HIT or .MISS. Also note this event is subect to CHA Filter1 which allows a user to opcode match against TOR entries, distinguish those requests satsified locally vs. those that came from a remote node, and so forth.

Table 2-133. Unit Masks for TOR_OCCUPANCY (Sheet 1 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
DDR	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxx1xx	DDR Access
PMM	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxx1xxx	PMM Access
PREMORPH_OPC	bxxxxxxxx	bxxxxXXXX XXXXXXX1x xxxxxxxxx	Match the PreMorphed Opcode in b[29:19] of the extended umask field
NONCOH	bxxxxxxxx	bx1xxxxxx xxxxxxxxx xxxxxxx	Just NonCoherent
LOCAL_TGT	bxxxxxxxx	bxxxxxxxxx xxxxxxxx1 xxxxxxx	Just Local Targets
REMOTE_TGT	bxxxxxxxx	bxxxxxxxxx xxxxxxx1x xxxxxxx	Just Remote Targets
MMIO	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx 1xxxxxx	MMIO Access
NEARMEM	bxxxxxxxx	bxxx1xxxxx xxxxxxxxx xxxxxxx	Just NearMem
HBM	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xx1xxxx	HBM Access
MMCFG	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx x1xxxxx	MMCFG Access
NOT_NEARMEM	bxxxxxxxx	bxx1xxxxxx xxxxxxxxx xxxxxxx	Just NotNearMem
HIT	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxx1	Just Hits
MATCH_OPC	bxxxxxxxx	bxxxxXXXX XXXXXXx1 xxxxxxxxx	Match the Opcode in b[29:19] of the extended umask field
ISOC	bxxxxxxxx	b1xxxxxxx xxxxxxxxx xxxxxxx	Just ISOC
MISS	bxxxxxxxx	bxxxxxxxxx xxxxxxxxx xxxxx1x	Just Misses
IA_SPECITOM	b00000001	0xCC57FF	SpecItoMs issued by iA Cores
IA_WBMTOI	b00000001	0xCC27FF	WbMtoIs issued by iA Cores
IA_MISS_DRD_LOCAL	b00000001	0xC816FE	DRds issued by iA Cores that Missed the LLC - HOMed locally
IA_MISS_CRD_LOCAL	b00000001	0xC80EFE	CRd issued by iA Cores that Missed the LLC - HOMed locally
IA_MISS_DRD_PREF_LOCAL_PMM	b00000001	0xC8968A	DRd_Prefs issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed locally
IA_WCIL	b00000001	0xC86FFF	WCiLs issued by iA Cores

Table 2-133. Unit Masks for TOR_OCCUPANCY (Sheet 2 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
IA_MISS_DRD_REMOTE_PMM	b00000001	0xC8170A	DRds issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed remotely
IA_HIT_DRD_OPT	b00000001	0xC827FD	DRd_Opts issued by iA Cores that hit the LLC
IA_DRD_OPT_PREF	b00000001	0xC8A7FF	DRd_Opt_Prefs issued by iA Cores
IA_HIT_DRDPTE	b00000001	0xC837FD	DRdPte issued by iA Cores due to a page walk that hit the LLC
IA_HIT_RFO_PREF	b00000001	0xC887FD	RFO_Prefs issued by iA Cores that Hit the LLC
IA_MISS_DRD_PREF_REMOTE_DDR	b00000001	0xC89706	DRd_Prefs issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed remotely
IA_HIT_ITOM	b00000001	0xCC47FD	ItoMs issued by iA Cores that Hit LLC
IA	b00000001	0xC001FF	All requests from iA Cores
IA_WCILF	b00000001	0xC867FF	WCiLF issued by iA Cores
IA_MISS_RFO	b00000001	0xC807FE	RFOs issued by iA Cores that Missed the LLC
IA_MISS_WCILF_PMM	b00000001	0xC8678A	WCiLFs issued by iA Cores targeting PMM that missed the LLC
IA_MISS_RFO_PREF_REMOTE	b00000001	0xC8877E	RFO_Prefs issued by iA Cores that Missed the LLC - HOMed remotely
IA_RFO_PREF	b00000001	0xC887FF	RFO_Prefs issued by iA Cores
IA_MISS_REMOTE_WCILF_DDR	b00000001	0xC86706	WCiLFs issued by iA Cores targeting DDR that missed the LLC - HOMed remotely
IA_MISS_DRD_PREF_LOCAL_DDR	b00000001	0xC89686	DRd_Prefs issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed locally
IA_MISS_DRD_PREF_DDR	b00000001	0xC89786	DRd_Prefs issued by iA Cores targeting DDR Mem that Missed the LLC
IA_DRD_OPT	b00000001	0xC827FF	DRd_Opts issued by iA Cores
IA_HIT_RFO	b00000001	0xC807FD	RFOs issued by iA Cores that Hit the LLC
IA_MISS_UCRDF	b00000001	0xC877DE	UCRdFs issued by iA Cores that Missed LLC
IA_MISS_LLCPREFCODE	b00000001	0xCCCFFE	LLCPrefCode issued by iA Cores that missed the LLC
IA_MISS_LOCAL_WCILF_PMM	b00000001	0xC8668A	WCiLFs issued by iA Cores targeting PMM that missed the LLC - HOMed locally
IA_MISS_REMOTE_WCILF_PMM	b00000001	0xC8670A	WCiLFs issued by iA Cores targeting PMM that missed the LLC - HOMed remotely
IA_MISS_LOCAL_WCIL_PMM	b00000001	0xC86E8A	WCiLs issued by iA Cores targeting PMM that missed the LLC - HOMed locally
IA_MISS_ITOM	b00000001	0xCC47FE	ItoMs issued by iA Cores that Missed LLC
IA_CRD	b00000001	0xC80FFF	CRDs issued by iA Cores
IA_MISS_LLCPREFDATA	b00000001	0xCCD7FE	LLCPrefData issued by iA Cores that missed the LLC
IA_MISS_WCILF	b00000001	0xC867FE	WCiLF issued by iA Cores that Missed the LLC
IA_HIT_CRD	b00000001	0xC80FFD	CRds issued by iA Cores that Hit the LLC
IA_MISS_LOCAL_WCILF_DDR	b00000001	0xC86686	WCiLFs issued by iA Cores targeting DDR that missed the LLC - HOMed locally
IA_CLFLUSHOPT	b00000001	0xC8D7FF	CLFlushOpts issued by iA Cores
IA_MISS_DRD	b00000001	0xC817FE	DRds issued by iA Cores that Missed the LLC
IA_MISS_CRD_PREF_REMOTE	b00000001	0xC88F7E	CRd_Prefs issued by iA Cores that Missed the LLC - HOMed remotely
IA_MISS_DRD_DDR	b00000001	0xC81786	DRds issued by iA Cores targeting DDR Mem that Missed the LLC

Table 2-133. Unit Masks for TOR_OCCUPANCY (Sheet 3 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
IA_MISS_CRD	b00000001	0xC80FFE	CRds issued by iA Cores that Missed the LLC
IA_HIT_LLCPREFCODE	b00000001	0xCCCFFD	LLCPrefCode issued by iA Cores that hit the LLC
IA_MISS_DRD_LOCAL_PMM	b00000001	0xC8168A	DRds issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed locally
IA_RFO	b00000001	0xC807FF	RFOs issued by iA Cores
IA_HIT_CRD_PREF	b00000001	0xC88FFD	CRd_Prefs issued by iA Cores that hit the LLC
IA_MISS_RFO_LOCAL	b00000001	0xC806FE	RFOs issued by iA Cores that Missed the LLC - HOMed locally
IA_MISS_CRD_PREF	b00000001	0xC88FFE	CRd_Prefs issued by iA Cores that Missed the LLC
IA_MISS_DRD_PREF	b00000001	0xC897FE	DRd_Prefs issued by iA Cores that Missed the LLC
IA_HIT_DRD_OPT_PREF	b00000001	0xC8A7FD	DRd_Opt_Prefs issued by iA Cores that hit the LLC
IA_MISS_DRD_PMM	b00000001	0xC8178A	DRds issued by iA Cores targeting PMM Mem that Missed the LLC
IA_MISS	b00000001	0xC001FE	All requests from iA Cores that Missed the LLC
IRQ_IA	bxxxxxxx1	bxxxxxxxxx xxxxxxxxxx xxxxxxx	IRQ - iA From an iA Core
IA_LLCPREFCODE	b00000001	0xCCCFFF	LLCPrefCode issued by iA Cores
IA_MISS_DRD_REMOTE	b00000001	0xC8177E	DRds issued by iA Cores that Missed the LLC - HOMed remotely
IA_DRDPTE	b00000001	0xC837FF	DRdPte issued by iA Cores due to a page walk
IA_MISS_WCIL_PMM	b00000001	0xC86F8A	WCiLs issued by iA Cores targeting PMM that missed the LLC
IA_MISS_CRD_REMOTE	b00000001	0xC80F7E	CRd issued by iA Cores that Missed the LLC - HOMed remotely
IA_DRD_PREF	b00000001	0xC897FF	DRd_Prefs issued by iA Cores
IA_MISS_RFO_PREF_LOCAL	b00000001	0xC886FE	RFO_Prefs issued by iA Cores that Missed the LLC - HOMed locally
IA_MISS_DRD_OPT_PREF	b00000001	0xC8A7FE	DRd_Opt_Prefs issued by iA Cores that missed the LLC
IA_MISS_WCILF_DDR	b00000001	0xC86786	WCiLFs issued by iA Cores targeting DDR that missed the LLC
IA_HIT_LLCPREFDATA	b00000001	0xCCD7FD	LLCPrefData issued by iA Cores that hit the LLC
IA_MISS_REMOTE_WCIL_DDR	b00000001	0xC86F06	WCiLs issued by iA Cores targeting DDR that missed the LLC - HOMed remotely
IA_MISS_DRD_LOCAL_DDR	b00000001	0xC81686	DRds issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed locally
IA_ITOMCACHENEAR	b00000001	0xCD47FF	ItoMCacheNears issued by iA Cores
IA_MISS_REMOTE_WCIL_PMM	b00000001	0xC86F0A	WCiLs issued by iA Cores targeting PMM that missed the LLC - HOMed remotely
IA_MISS_DRD_REMOTE_DDR	b00000001	0xC81706	DRds issued by iA Cores targeting DDR Mem that Missed the LLC - HOMed remotely
IA_LLCPREFDATA	b00000001	0xCCD7FF	LLCPrefData issued by iA Cores
IA_HIT_LLCPREFRFO	b00000001	0xCCC7FD	LLCPrefRFO issued by iA Cores that hit the LLC
IA_CLFLUSH	b00000001	0xC8C7FF	CLFlushes issued by iA Cores
IA_LLCPREFRFO	b00000001	0xCCC7FF	LLCPrefRFO issued by iA Cores
IA_MISS_CRD_PREF_LOCAL	b00000001	0xC88EFE	CRd_Prefs issued by iA Cores that Missed the LLC - HOMed locally
IA_ITOM	b00000001	0xCC47FF	ItoMs issued by iA Cores

Table 2-133. Unit Masks for TOR_OCCUPANCY (Sheet 4 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
IA_HIT_DRD	b00000001	0xC817FD	DRds issued by iA Cores that Hit the LLC
IA_HIT	b00000001	0xC001FD	All requests from iA Cores that Hit the LLC
IA_MISS_WIL	b00000001	0xC87FDE	WiLs issued by iA Cores that Missed LLC
IA_MISS_DRD_OPT	b00000001	0xC827FE	DRd_Opt issued by iA Cores that missed the LLC
IA_DRD	b00000001	0xC817FF	DRds issued by iA Cores
IA_MISS_LOCAL_WCIL_DDR	b00000001	0xC86E86	WCiLs issued by iA Cores targeting DDR that missed the LLC - HOMed locally
IA_MISS_RFO_REMOTE	b00000001	0xC8077E	RFOs issued by iA Cores that Missed the LLC - HOMed remotely
IA_MISS_WCIL_DDR	b00000001	0xC86F86	WCiLs issued by iA Cores targeting DDR that missed the LLC
IA_MISS_LLCPREFRFO	b00000001	0xCCC7FE	LLCPrefRFO issued by iA Cores that missed the LLC
IA_MISS_DRDPTE	b00000001	0xC837FE	DRdPte issued by iA Cores due to a page walk that missed the LLC
IA_MISS_DRD_PREF_REMOTE_PMM	b00000001	0xC8970A	DRd_Prefs issued by iA Cores targeting PMM Mem that Missed the LLC - HOMed remotely
IA_MISS_RFO_PREF	b00000001	0xC887FE	RFO_Prefs issued by iA Cores that Missed the LLC
IA_MISS_DRD_PREF_PMM	b00000001	0xC8978A	DRd_Prefs issued by iA Cores targeting PMM Mem that Missed the LLC
IA_HIT_DRD_PREF	b00000001	0xC897FD	DRd_Prefs issued by iA Cores that Hit the LLC
LOC_IA	b00000001	0xC000FF	All from Local iA All locally initiated requests from iA Cores
IA_MISS_WCIL	b00000001	0xC86FFE	WCiLs issued by iA Cores that Missed the LLC
EVICT	bxxxxxx1x	bxxxxxxxxx xxxxxxxxx xxxxxxx	SF/LLC Evictions TOR allocation occurred as a result of SF/LLC evictions (came from the ISMQ)
IO_PCIRDCUR	b00000100	0xC8F3FF	PCIRdCurs issued by IO Devices
IO_CLFLUSH	b00000100	0xC8C3FF	CLFlushes issued by IO Devices
IO_HIT	b00000100	0xC001FD	All requests from IO Devices that hit the LLC
IO_ITOM	b00000100	0xCC43FF	ItoMs issued by IO Devices
IO_ITOMCACHENEAR	b00000100	0xCD43FF	ItoMCacheNears, indicating a partial write request, from IO Devices
IO_MISS	b00000100	0xC001FE	All requests from IO Devices that missed the LLC
IO_RFO	b00000100	0xC803FF	RFOs issued by IO Devices
LOC_IO	b00000100	0xC000FF	All from Local IO All locally generated IO traffic
IO_HIT_ITOMCACHENEAR	b00000100	0xCD43FD	ItoMCacheNears, indicating a partial write request, from IO Devices that hit the LLC
IO_MISS_ITOM	b00000100	0xCC43FE	ItoMs issued by IO Devices that missed the LLC
IO_MISS_PCIRDCUR	b00000100	0xC8F3FE	PCIRdCurs issued by IO Devices that missed the LLC
IO_MISS_RFO	b00000100	0xC803FE	RFOs issued by IO Devices that missed the LLC
IO	b00000100	0xC001FF	All requests from IO Devices
IO_MISS_ITOMCACHENEAR	b00000100	0xCD43FE	ItoMCacheNears, indicating a partial write request, from IO Devices that missed the LLC
IO_HIT_ITOM	b00000100	0xCC43FD	ItoMs issued by IO Devices that Hit the LLC
IO_HIT_RFO	b00000100	0xC803FD	RFOs issued by IO Devices that hit the LLC

Table 2-133. Unit Masks for TOR_OCCUPANCY (Sheet 5 of 5)

Extension	umask [15:8]	xtra [57:32]	Description
PRQ	bxxxxx1xx	bxxxxxxxxx xxxxxxxxx xxxxxxx	PRQ - IOSF From a PCIe Device
IO_WBMTOI	b00000100	0xCC23FF	WbMtoIs issued by IO Devices
IO_HIT_PCIRDCUR	b00000100	0xC8F3FD	PCIRdCurs issued by IO Devices that hit the LLC
LOC_ALL	b00000101	0xC000FF	All from Local iA and IO All locally initiated requests
IPQ	bxxxx1xxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	IPQ
IRQ_NON_IA	bxxx1xxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	IRQ - Non iA
PRQ_NON_IOSF	bxx1xxxxx	bxxxxxxxxx xxxxxxxxx xxxxxxx	PRQ - Non IOSF

WB_PUSH_MTOI

- **Title:** WbPushMtoI
- **Category:** HA WBPUSHMTOI Events
- **Event Code:** 0x56
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of times when the CHA was received WbPushMtoI

Table 2-134. Unit Masks for WB_PUSH_MTOI

Extension	umask [15:8]	Description
LLC	bxxxxxxx1	Pushed to LLC Counts the number of times when the CHA was able to push WbPushMToI to LLC
MEM	bxxxxxx1x	Pushed to Memory Counts the number of times when the CHA was unable to push WbPushMToI to LLC (hence pushed it to MEM)

WRITE_NO_CREDITS

- **Title:** CHA iMC CHNx WRITE Credits Empty
- **Category:** MC Credit and Traffic Events
- **Event Code:** 0x5a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of times when there are no credits available for sending WRITEs from the CHA into the iMC. In order to send WRITEs into the memory controller, the HA must first acquire a credit for the iMC's BL Ingress queue.

Table 2-135. Unit Masks for WRITE_NO_CREDITS

Extension	umask [15:8]	xtra [57:32]	Description
MC11	bxxxxxxxx	bxxxx1xxx	MC11 Filter for memory controller 11 only.
MC13	bxxxxxxxx	bxx1xxxxx	MC13 Filter for memory controller 13 only.
MC9	bxxxxxxxx	bxxxxxx1x	MC9 Filter for memory controller 9 only.
MC10	bxxxxxxxx	bxxxxx1xx	MC10 Filter for memory controller 10 only.
MC8	bxxxxxxxx	bxxxxxxx1	MC8 Filter for memory controller 8 only.
MC12	bxxxxxxxx	bxxx1xxxx	MC12 Filter for memory controller 12 only.
MC0	bxxxxxxx1	bxxxxxxxx	MC0 Filter for memory controller 0 only.
MC1	bxxxxxx1x	bxxxxxxxx	MC1 Filter for memory controller 1 only.
MC2	bxxxxx1xx	bxxxxxxxx	MC2 Filter for memory controller 2 only.
MC3	bxxxx1xxx	bxxxxxxxx	MC3 Filter for memory controller 3 only.
MC4	bxxx1xxxx	bxxxxxxxx	MC4 Filter for memory controller 4 only.
MC5	bxx1xxxxx	bxxxxxxxx	MC5 Filter for memory controller 5 only.
MC6	bx1xxxxxx	bxxxxxxxx	MC6 Filter for memory controller 6 only.
MC7	b1xxxxxxx	bxxxxxxxx	MC7 Filter for memory controller 7 only.

XPT_PREF

- **Title:** XPT Prefetches
- **Category:** XPT Events
- **Event Code:** 0x6f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-136. Unit Masks for XPT_PREF (Sheet 1 of 2)

Extension	umask [15:8]	Description
SENT0	bxxxxxxx1	Sent (on 0?) Number of XPT prefetches sent
DROP0_NOCRD	bxxxxx1xx	Dropped (on 0?) - No Credits Number of XPT prefetches dropped due to lack of XPT AD egress credits
DROP0_CONFLICT	bxxxx1xxx	Dropped (on 0?) - Conflict Number of XPT prefetches dropped due to AD CMS write port contention
SENT1	bxxx1xxxx	Sent (on 1?) Number of XPT prefetches sent

Table 2-136. Unit Masks for XPT_PREF (Sheet 2 of 2)

Extension	umask [15:8]	Description
DROP1_NOCRD	bx1xxxxxx	Dropped (on 1?) - No Credits Number of XPT prefetches dropped due to lack of XPT AD egress credits
DROP1_CONFLICT	b1xxxxxxx	Dropped (on 1?) - Conflict Number of XPT prefetches dropped due to AD CMS write port contention

2.4 Memory Controller (IMC) Performance Monitoring

The 3rd Gen Intel Xeon Processor Scalable Family integrated Memory Controller provides the interface to DRAM and communicates to the rest of the Uncore through the Mesh2Mem block.

The memory controller also provides a variety of RAS features, such as ECC, memory access retry, memory scrubbing, thermal throttling, mirroring, and rank sparing.

2.4.1 Functional Overview

The memory controller communicates to DRAM, translating read and write commands into specific memory commands and schedules them with respect to memory timing. The other main function of the memory controller is advanced ECC support.

The 3rd Gen Intel Xeon Processor Scalable Family supports up to 8 channels of DDR4 with 2 channels per
memory controller.

Each memory controller supports channel speeds up to 3200 MT/s.

A selection of IMC functionality that performance monitoring provides some insight into:

- Supports up to 16 ranks per channel with 16 independent banks per rank.
- ECC support (correct any error within a x4 device)
- Open or closed page policy
- ISOCH
- Demand and Patrol Scrubbing support
- Support for LR-DIMMs (load reduced) for a buffered memory solution demanding higher capacity memory subsytems.

2.4.2 IMC Performance Monitoring Overview

The IMC supports event monitoring through four 48-bit wide counters (MC_CHy_PCI_PMON_CTR{3:0}) and one fixed counter (MC_CHy_PCI_PMON_FIXED_CTR) for each DRAM channel (of which there are 2 in 3rd Gen Intel Xeon Processor Scalable Family) the MC is attached to. Each of these counters can be programmed (MC_CHy_PCI_PMON_CTL{3:0}) to capture any MC event. The MC counters will increment by a maximum of 8b per cycle.

2.4.3 Additional IMC Performance Monitoring

Following is a counter that always tracks the number of DRAM clocks (dclks - half of DDR speed) in the IMC. The dclk never changes frequency (on a given system), and therefore is a good measure of wall clock (unlike the Uncore clock which can change frequency based on system load). This clock is generally a bit slower than the uclk (~800MHz to ~1.066 GHz) and therefore has less fidelity.

Figure 2-6. PMON Control Register for DCLK

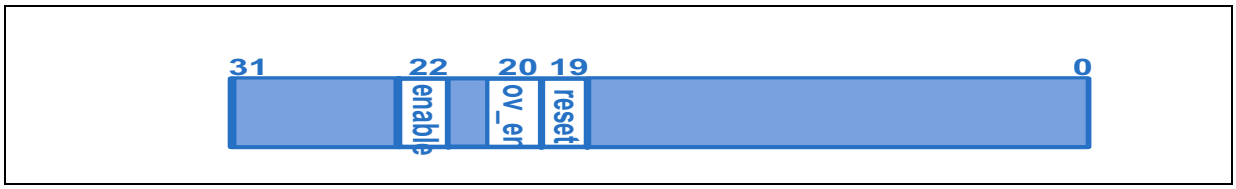


Table 2-137. MC_CHy_PCI_PMON_FIXED_CTL Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	31:23	RV	0	Ignored
en	22	RW-V	0	Local Counter Enable.
ig	21	RV	0	Ignored
ov_en	20	RW-V	0	When this bit is asserted and the corresponding counter overflows, a PMI exception is sent to the UBox.
rst	19	WO	0	When set to 1, the corresponding counter will be cleared to 0.
ig	18:0	RV	0	Ignored

Table 2-138. MC_CHy_PCI_PMON_CTR{FIXED,3-0} Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:48	RV	0	Ignored
event_count	47:0	RW-V	0	48-bit performance event counter

There are a few free-running counters, providing information highly valuable to a wide array of customers, in each iMC that collect counts for cumulative Read / Write Bandwidth across all channels.

'Free Running' counters cannot be changed by SW operating in a normal environment. SW cannot write them, cannot stop them and cannot reset the values.

Note: Counting will be suspended when the MC is powered down.

There is one register per stack to track the number of DDR cycles as measured by MC

And one extra register each to track Read and Write Bandwidth:

- **Read bandwidth** - counts bytes of data transmitted from the DIMM to MC.

Table 2-139. MC_MMIO_PMON_FRCTR_DCLK Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:48	RV	0	Ignored
event_count	47:0	RO-V	0	48-bit running count of DDR clocks captured in MC

Table 2-140. MC_MMIO_PMON_FRCTR_RD_BW Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:48	RV	0	Ignored
event_count	47:0	RO-V	0	48-bit running count of data bytes read from attached DIMM

- **Write bandwidth** - counts bytes of data transmitted from MC to DIMM.

Table 2-141. MC_MMIO_PMON_FRCTR_WR_BW Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:48	RV	0	Ignored
event_count	47:0	RO-V	0	48-bit running count of data bytes written to attached DIMM

2.4.4 IMC Performance Monitoring Events

A sampling of events available for monitoring in the IMC:

- **Translated commands**: Various Read and Write CAS commands
- **Memory commands**: CAS, Precharge, Refresh, Preemptions, and so forth.
- Page hits and page misses.
- **Page Closing** Events
- **Control of power consumption**: Thermal Throttling by Rank, Time spent in CKE ON mode, and so forth.

and many more.

Internal IMC Queues:

RPQ - Read Pending Queue for DDR.

WPQ - Write Pending Queue for DDR.

PMM RPQ - Read Pending Queue for PMM.

PMM WPQ - Write Pending Queue for PMM.

SCOREBOARD - To track 2LM traffic.

2.4.5 iMC Box Events Ordered By Code

The following table summarizes the directly measured iMC Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
CLOCKTICKS	0x00	0-3	1	DRAM Clockticks
ACT_COUNT	0x01	0-3	1	DRAM Activate Count
PRE_COUNT	0x02	0-3	1	DRAM Precharge commands.
CAS_COUNT	0x04	0-3	1	DRAM RD_CAS and WR_CAS Commands.
RPQ_INSERTS	0x10	0-3	1	Read Pending Queue Allocations
RPQ_CYCLES_NE	0x11	0-3	1	Read Pending Queue Not Empty
RPQ_CYCLES_FULL_PCH0	0x12	0-3	1	Read Pending Queue Full Cycles
RPQ_CYCLES_FULL_PCH1	0x15	0-3	1	Read Pending Queue Full Cycles
WPQ_CYCLES_FULL_PCH1	0x16	0-3	1	Write Pending Queue Full Cycles
RDB_INSERTS	0x17	0-3	1	Read Data Buffer Inserts
RDB_NOT_EMPTY	0x18	0-3	1	Read Data Buffer Not Empty
RDB_FULL	0x19	0-3	1	Read Data Buffer Full
RDB_OCCUPANCY	0x1a	0-3	1	Read Data Buffer Occupancy
WPQ_INSERTS	0x20	0-3	1	Write Pending Queue Allocations
WPQ_CYCLES_NE	0x21	0-3	1	Write Pending Queue Not Empty
WPQ_CYCLES_FULL_PCH0	0x22	0-3	1	Write Pending Queue Full Cycles
WPQ_READ_HIT	0x23	0-3	1	Write Pending Queue CAM Match
WPQ_WRITE_HIT	0x24	0-3	1	Write Pending Queue CAM Match
PARITY_ERRORS	0x2c		0	
POWER_SELF_REFRESH	0x43	0-3	0	Clock-Enabled Self-Refresh
DRAM_PRE_ALL	0x44	0-3	1	DRAM Precharge All Commands
DRAM_REFRESH	0x45	0-3	1	Number of DRAM Refreshes Issued
POWER_THROTTLE_CYCLES	0x46	0-3	1	Throttle Cycles for Rank 0
POWER_CKE_CYCLES	0x47	0-3	16	CKE_ON_CYCLES by Rank
RPQ_OCCUPANCY_PCH0	0x80	0-3	22	Read Pending Queue Occupancy
RPQ_OCCUPANCY_PCH1	0x81	0-3	22	Read Pending Queue Occupancy
WPQ_OCCUPANCY_PCH0	0x82	0-3	40	Write Pending Queue Occupancy
WPQ_OCCUPANCY_PCH1	0x83	0-3	40	Write Pending Queue Occupancy
POWER_CHANNEL_PPD	0x85	0-3	4	Channel PPD Cycles
POWER_CRIT_THROTTLE_CYCLES	0x86	0-3	1	Throttle Cycles for Rank 0
PCLS	0xa0		0	
SB_CYCLES_NE	0xd0	0-3	1	Scoreboard Cycles Not-Empty
SB_CYCLES_FULL	0xd1	0-3	1	Scoreboard Cycles Full
SB_ACCESSES	0xd2	0-3	1	Scoreboard Accesses
TAGCHK	0xd3	0-3	1	2LM Tag Check
SB_REJECT	0xd4	0-3	1	Number of Scoreboard Requests Rejected
SB_OCCUPANCY	0xd5	0-3	128	Scoreboard Occupancy
SB_INSERTS	0xd6	0-3	1	Scoreboard Inserts
SB_STRV_ALLOC	0xd7	0-3	1	

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
SB_STRV_OCC	0xd8	0-3	63	
SB_CANARY	0xd9	0-3	1	
SB_PREF_INSERTS	0xda	0-3	1	Scoreboard Prefetch Inserts
SB_PREF_OCCUPANCY	0xdb	0-3	1	Scoreboard Prefetch Occupancy
SB_TAGGED	0xdd		0	
SB_STRV_DEALLOC	0xde	0-3	1	
PMM_RPQ_OCCUPANCY	0xe0		0	PMM Read Pending Queue Occupancy
PMM_RPQ_CYCLES_NE	0xe1		0	PMM Read Queue Cycles Not Empty
PMM_RPQ_CYCLES_FULL	0xe2		0	PMM Read Queue Cycles Full
PMM_RPQ_INSERTS	0xe3		0	PMM Read Queue Inserts
PMM_WPQ_OCCUPANCY	0xe4		0	PMM Write Pending Queue Occupancy
PMM_WPQ_CYCLES_NE	0xe5		0	PMM Write Queue Cycles Not Empty
PMM_WPQ_CYCLES_FULL	0xe6		0	PMM Write Queue Cycles Full
PMM_WPQ_INSERTS	0xe7		0	PMM Write Queue Inserts
PMM_WPQ_FLUSH	0xe8		0	
PMM_WPQ_FLUSH_CYC	0xe9		0	
PMM_CMD1	0xea		0	PMM Commands
PMM_CMD2	0xeb		0	PMM Commands - Part 2

2.4.6 iMC Box Common Metrics (Derived Events)

The following table summarizes metrics commonly calculated from iMC Box events.

Symbol Name: Definition	Equation
MEM_BW_READS: Memory bandwidth consumed by reads. Expressed in bytes.	(CAS_COUNT.RD * 64)
MEM_BW_TOTAL: Total memory bandwidth. Expressed in bytes.	MEM_BW_READS + MEM_BW_WRITES
MEM_BW_WRITES: Memory bandwidth consumed by writes Expressed in bytes.	(CAS_COUNT.WR * 64)
PCT_CYCLES_CRITICAL_THROTTLE: The percentage of cycles all DRAM ranks in critical thermal throttling	POWER_CRITICAL_THROTTLE_CYCLES / MC_Chy_PCI_PMON_CTR_FIXED
PCT_CYCLES_DRAM_RANKx_IN_THR: The percentage of cycles DRAM rank (x) spent in thermal throttling.	POWER_THROTTLE_CYCLES.RANKx / MC_Chy_PCI_PMON_CTR_FIXED
PCT_CYCLES_PPD: The percentage of cycles all DRAM ranks in PPD mode	POWER_CHANNEL_PPD / MC_Chy_PCI_PMON_CTR_FIXED
PCT_CYCLES_SELF_REFRESH: The percentage of cycles Memory is in self refresh power mode	POWER_SELF_REFRESH / MC_Chy_PCI_PMON_CTR_FIXED
PCT_RD_REQUESTS: Percentage of read requests from total requests.	RPQ_INSERTS / (RPQ_INSERTS + WPQ_INSERTS)

Symbol Name: Definition	Equation
PCT_REQUESTS_PAGE_EMPTY: Percentage of memory requests that resulted in Page Empty	PRE_COUNT.PGT - CAS_COUNT.ALL
PCT_REQUESTS_PAGE_HIT: Percentage of memory requests that resulted in Page Misses - Precharge followed by Activate	(PRE_COUNT.RD + PRE_COUNT.WR) / CAS_COUNT.ALL
PCT_WR_REQUESTS: Percentage of write requests from total requests.	WPQ_INSERTS / (RPQ_INSERTS + WPQ_INSERTS)

2.4.7 iMC Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the iMC Box.

ACT_COUNT

- **Title:** DRAM Activate Count
- **Category:** ACT Events
- **Event Code:** 0x01
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of DRAM Activate commands sent on this channel. Activate commands are issued to open up a page on the DRAM devices so that it can be read or written to with a CAS. One can calculate the number of Page Misses by subtracting the number of Page Miss precharges from the number of Activates.

Table 2-142. Unit Masks for ACT_COUNT

Extension	umask [15:8]	Description
BYP	bxxxx1xxx	Activate due to Bypass
ALL	b00001011	All Activates

CAS_COUNT

- **Title:** DRAM RD_CAS and WR_CAS Commands.
- **Category:** CAS Events
- **Event Code:** 0x04
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** DRAM RD_CAS and WR_CAS Commands

Table 2-143. Unit Masks for CAS_COUNT

Extension	umask [15:8]	Description
RD_REG	bxxxxxxx1	DRAM RD_CAS commands w/out auto-pre Counts the total number or DRAM Read CAS commands issued on this channel. This includes both regular RD CAS commands as well as those with implicit Precharge. We do not filter based on major mode, as RD_CAS is not issued during WMM (with the exception of underfills).
RD_PRE_REG	bxxxxxx1x	DRAM RD_CAS commands w/auto-pre Counts the total number or DRAM Read CAS commands issued on this channel. This includes both regular RD CAS commands as well as those with explicit Precharge. AutoPre is only used in systems that are using closed page policy. We do not filter based on major mode, as RD_CAS is not issued during WMM (with the exception of underfills).
RD_UNDERFILL	bxxxxx1xx	Underfill Read Issued
RD_PRE_UNDERFILL	bxxxx1xxx	
RD	b00001111	All DRAM Reads Counts the total number of DRAM Read CAS commands, w/ and w/o auto-pre, issued on this channel. This includes underfills.
WR_NONPRE	bxxx1xxxx	DRAM WR_CAS commands w/o auto-pre
WR_PRE	bxx1xxxxx	DRAM WR_CAS commands w/ auto-pre
WR	b00110000	All DRAM WR_CAS (both Modes) Counts the total number of DRAM Write CAS commands issued, w/ and w/o auto-pre, on this channel.
ALL	b00111111	All DRAM Read and Write actions Counts the total number of DRAM CAS commands issued on this channel.

CLOCKTICKS

- **Title:** DRAM Clockticks
- **Category:** DCLK Events
- **Event Code:** 0x00
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

DRAM_PRE_ALL

- **Title:** DRAM Precharge All Commands
- **Category:** DRAM_PRE_ALL Events
- **Event Code:** 0x44
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times that the precharge all command was sent.

DRAM_REFRESH

- **Title:** Number of DRAM Refreshes Issued
- **Category:** DRAM_REFRESH Events
- **Event Code:** 0x45
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of refreshes issued.

Table 2-144. Unit Masks for DRAM_REFRESH

Extension	umask [15:8]	Description
OPPORTUNISTIC	bxxxxxxx1	
PANIC	bxxxxxx1x	
HIGH	bxxxxx1xx	

PARITY_ERRORS

- **Title:**
- **Category:** Error Events
- **Event Code:** 0x2c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PCLS

- **Title:**
- **Category:** Debug Events
- **Event Code:** 0xa0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-145. Unit Masks for PCLS

Extension	umask [15:8]	Description
RD	bxxxxxxx1	
WR	bxxxxxx1x	
TOTAL	bxxxxx1xx	

PMM_CMD1

- **Title:** PMM Commands
- **Category:** PMM CMD Events
- **Event Code:** 0xea
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-146. Unit Masks for PMM_CMD1

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All Counts all commands issued to PMM
RD	bxxxxxx1x	Reads - RPQ Counts read requests issued to the PMM RPQ
WR	bxxxxx1xx	Writes Counts write commands issued to PMM

Table 2-146. Unit Masks for PMM_CMD1

Extension	umask [15:8]	Description
UFILL_RD	bxxxx1xxx	Underfill reads Counts underfill read commands, due to a partial write, issued to PMM
RPQ_GNTS	bxxx1xxxx	RPQ GNTs
WPQ_GNTS	bxx1xxxxx	Underfill GNTs
MISC_GNT	bx1xxxxxx	Misc GNTs
MISC	b1xxxxxxx	Misc Commands (error, flow ACKs)

PMM_CMD2

- **Title:** PMM Commands - Part 2
- **Category:** PMM CMD Events
- **Event Code:** 0xeb
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-147. Unit Masks for PMM_CMD2

Extension	umask [15:8]	Description
OPP_RD	bxxxxxxx1	Opportunistic Reads
NODATA_EXP	bxxxxxx1x	Expected No data packet (ERID matched NDP encoding)
NODATA_UNEXP	bxxxxx1xx	Unexpected No data packet (ERID matched a Read, but data was a NDP)
REQS_SLOT0	bxxxx1xxx	Read Requests - Slot 0
REQS_SLOT1	bxxx1xxxx	Read Requests - Slot 1
PMM_ECC_ERROR	bxx1xxxxx	ECC Errors
PMM_ERID_ERROR	bx1xxxxxx	ERID detectable parity error
PMM_ERID_STARVED	b1xxxxxxx	

PMM_RPQ_CYCLES_FULL

- **Title:** PMM Read Queue Cycles Full
- **Category:** PMM RPQ Events
- **Event Code:** 0xe2
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PMM_RPQ_CYCLES_NE

- **Title:** PMM Read Queue Cycles Not Empty
- **Category:** PMM RPQ Events
- **Event Code:** 0xe1
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PMM_RPQ_INSERTS

- **Title:** PMM Read Queue Inserts
- **Category:** PMM RPQ Events
- **Event Code:** 0xe3
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of read requests allocated in the PMM Read Pending Queue. This includes both ISOCH and non-ISOCH requests.

PMM_RPQ_OCCUPANCY

- **Title:** PMM Read Pending Queue Occupancy
- **Category:** PMM RPQ Events
- **Event Code:** 0xe0
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Accumulates the per cycle occupancy of the PMM Read Pending Queue.
- **NOTE:** Dividing by Inserts provides the average latency entries were resident in the queue. The HA must acquire a credit from the iMC to ensure the request will be accepted and queued. The credit must be received by the HA before sending the request. The read queue entry is deallocated once the data has been transferred from the IXP DIMM

Table 2-148. Unit Masks for PMM_RPQ_OCCUPANCY

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	
NO_GNT	bxxxxxx1x	
GNT_WAIT	bxxxxx1xx	

PMM_WPQ_CYCLES_FULL

- **Title:** PMM Write Queue Cycles Full
- **Category:** PMM WPQ Events
- **Event Code:** 0xe6
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PMM_WPQ_CYCLES_NE

- **Title:** PMM Write Queue Cycles Not Empty
- **Category:** PMM WPQ Events
- **Event Code:** 0xe5
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PMM_WPQ_FLUSH

- **Title:**
- **Category:** PMM WPQ Events
- **Event Code:** 0xe8
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PMM_WPQ_FLUSH_CYC

- **Title:**
- **Category:** PMM WPQ Events
- **Event Code:** 0xe9
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PMM_WPQ_INSERTS

- **Title:** PMM Write Queue Inserts
- **Category:** PMM WPQ Events
- **Event Code:** 0xe7
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts number of write requests allocated in the PMM Write Pending Queue.

PMM_WPQ_OCCUPANCY

- **Title:** PMM Write Pending Queue Occupancy
- **Category:** PMM WPQ Events
- **Event Code:** 0xe4
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Accumulates the per cycle occupancy of the PMM Write Pending Queue.
- **NOTE:** Dividing by Inserts provides the average latency entries were resident in the queue. The HA must acquire a credit from the iMC to ensure the request will be accepted and queued. The credit must be received by the HA before sending the request. The write queue entry is deallocated once the iMC has ensured the data was transferred to the IXP DIMM.

Table 2-149. Unit Masks for PMM_WPQ_OCCUPANCY

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	
CAS	bxxxxxx1x	
PWR	bxxxxx1xx	

POWER_CHANNEL_PPD

- **Title:** Channel PPD Cycles
- **Category:** POWER Events
- **Event Code:** 0x85
- **Max. Inc/Cyc:.** 4

Register Restrictions: 0-3

- **Definition:** Number of cycles when all the ranks in the channel are in PPD mode. If IBT=off is enabled, then this can be used to count those cycles. If it is not enabled, then this can count the number of cycles when that could have been taken advantage of.
- **NOTE:** IBT = Input Buffer Termination = On

POWER_CKE_CYCLES

- **Title:** CKE_ON_CYCLES by Rank
- **Category:** POWER Events
- **Event Code:** 0x47
- **Max. Inc/Cyc:.** 16

Register Restrictions: 0-3

- **Definition:** Number of cycles spent in CKE ON mode. The filter allows you to select a rank to monitor. If multiple ranks are in CKE ON mode at one time, the counter will ONLY increment by one rather than doing accumulation. Multiple counters will need to be used to track multiple ranks simultaneously. There is no distinction between the different CKE modes (APD, PPDS, PPDF). This can be determined based on the system programming. These events should commonly be used with Invert to get the number of cycles in power saving mode. Edge Detect is also useful here. Make sure that you do NOT use Invert with Edge Detect (this just confuses the system and is not necessary).

Table 2-150. Unit Masks for POWER_CKE_CYCLES

Extension	umask [15:8]	Description
LOW_0	b00000001	DIMM ID
LOW_1	b00000010	DIMM ID
LOW_2	b00000100	DIMM ID
LOW_3	b00001000	DIMM ID

POWER_CRIT_THROTTLE_CYCLES

- **Title:** Throttle Cycles for Rank 0
- **Category:** POWER Events
- **Event Code:** 0x86
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles while the iMC is being throttled by either thermal constraints or by the PCU throttling. It is not possible to distinguish between the two. This can be filtered by rank. If multiple ranks are selected and are being throttled at the same time, the counter will only increment by 1.

Table 2-151. Unit Masks for POWER_CRIT_THROTTLE_CYCLES

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Thermal throttling is performed per DIMM. We support 3 DIMMs per channel. This ID allows us to filter by ID.
SLOT1	bxxxxxx1x	

POWER_SELF_REFRESH

- **Title:** Clock-Enabled Self-Refresh
- **Category:** POWER Events
- **Event Code:** 0x43
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the iMC is in self-refresh and the iMC still has a clock. This happens in some package C-states. For example, the PCU may ask the iMC to enter self-refresh even though some of the cores are still processing. One use of this is for Monroe technology. Self-refresh is required during package C3 and C6, but there is no clock in the iMC at this time, so it is not possible to count these cases.

POWER_THROTTLE_CYCLES

- **Title:** Throttle Cycles for Rank 0
- **Category:** POWER Events
- **Event Code:** 0x46
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles while the iMC is being throttled by either thermal constraints or by the PCU throttling. It is not possible to distinguish between the two. This can be filtered by rank. If multiple ranks are selected and are being throttled at the same time, the counter will only increment by 1.

Table 2-152. Unit Masks for POWER_THROTTLE_CYCLES

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Thermal throttling is performed per DIMM. We support 3 DIMMs per channel. This ID allows us to filter by ID.
SLOT1	bxxxxxx1x	

PRE_COUNT

- **Title:** DRAM Precharge commands.
- **Category:** PRE Events
- **Event Code:** 0x02
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of DRAM Precharge commands sent on this channel.

Table 2-153. Unit Masks for PRE_COUNT

Extension	umask [15:8]	Description
RD	bxxxxx1xx	Precharge due to read Precharge from read bank scheduler
WR	bxxxx1xxx	Precharge due to write Precharge from write bank scheduler
PAGE_MISS	bxxxx11xx	Precharge due to page miss Pages Misses are due to precharges from bank scheduler (rd/wr requests)
PGT	bxxx1xxxx	Precharge due to page table Prechages from Page Table
ALL	b00011100	

RDB_FULL

- **Title:** Read Data Buffer Full
- **Category:** RDB Events
- **Event Code:** 0x19
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

RDB_INSERTS

- **Title:** Read Data Buffer Inserts
- **Category:** RDB Events
- **Event Code:** 0x17
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

RDB_NOT_EMPTY

- **Title:** Read Data Buffer Not Empty
- **Category:** RDB Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

RDB_OCCUPANCY

- **Title:** Read Data Buffer Occupancy
- **Category:** RDB Events
- **Event Code:** 0x1a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

RPQ_CYCLES_FULL_PCH0

- **Title:** Read Pending Queue Full Cycles
- **Category:** RPQ Events
- **Event Code:** 0x12
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the Read Pending Queue is full. When the RPQ is full, the HA will not be able to issue any additional read requests into the iMC. This count should be similar count in the HA which tracks the number of cycles that the HA has no RPQ credits, just somewhat smaller to account for the credit return overhead. We generally do not expect to see RPQ become full except for potentially during Write Major Mode or while running with slow DRAM. This event only tracks non-ISOC queue entries.

RPQ_CYCLES_FULL_PCH1

- **Title:** Read Pending Queue Full Cycles
- **Category:** RPQ Events
- **Event Code:** 0x15
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the Read Pending Queue is full. When the RPQ is full, the HA will not be able to issue any additional read requests into the iMC. This count should be similar count in the HA which tracks the number of cycles that the HA has no RPQ credits, just somewhat smaller to account for the credit return overhead. We generally do not expect to see RPQ become full except for potentially during Write Major Mode or while running with slow DRAM. This event only tracks non-ISOC queue entries.

RPQ_CYCLES_NE

- **Title:** Read Pending Queue Not Empty
- **Category:** RPQ Events
- **Event Code:** 0x11
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles that the Read Pending Queue is not empty. This can then be used to calculate the average occupancy (in conjunction with the Read Pending Queue Occupancy count). The RPQ is used to schedule reads out to the memory controller and to track the requests. Requests allocate into the RPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the HA to the iMC. They deallocate after the CAS command has been issued to memory. This filter is to be used in conjunction with the occupancy filter so that one can correctly track the average occupancies for schedulable entries and scheduled requests.

Table 2-154. Unit Masks for RPQ_CYCLES_NE

Extension	umask [15:8]	Description
PCH0	bxxxxxxx1	
PCH1	bxxxxxx1x	

RPQ_INSERTS

- **Title:** Read Pending Queue Allocations
- **Category:** RPQ Events
- **Event Code:** 0x10
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of allocations into the Read Pending Queue. This queue is used to schedule reads out to the memory controller and to track the requests. Requests allocate into the RPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the HA to the iMC. They deallocate after the CAS command has been issued to memory. This includes both ISOCH and non-ISOCH requests.

Table 2-155. Unit Masks for RPQ_INSERTS

Extension	umask [15:8]	Description
PCH0	bxxxxxxx1	
PCH1	bxxxxxx1x	

RPQ_OCCUPANCY_PCH0

- **Title:** Read Pending Queue Occupancy
- **Category:** RPQ Events
- **Event Code:** 0x80
- **Max. Inc/Cyc:.** 22

Register Restrictions: 0-3

- **Definition:** Accumulates the occupancies of the Read Pending Queue each cycle. This can then be used to calculate both the average occupancy (in conjunction with the number of cycles not empty) and the average latency (in conjunction with the number of allocations). The RPQ is used to schedule reads out to the memory controller and to track the requests. Requests allocate into the RPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the HA to the iMC. They deallocate after the CAS command has been issued to memory.

RPQ_OCCUPANCY_PCH1

- **Title:** Read Pending Queue Occupancy
- **Category:** RPQ Events
- **Event Code:** 0x81
- **Max. Inc/Cyc:.** 22

Register Restrictions: 0-3

- **Definition:** Accumulates the occupancies of the Read Pending Queue each cycle. This can then be used to calculate both the average occupancy (in conjunction with the number of cycles not empty) and the average latency (in conjunction with the number of allocations). The RPQ is used to schedule reads out to the memory controller and to track the requests. Requests allocate into the RPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the HA to the iMC. They deallocate after the CAS command has been issued to memory.

SB_ACCESSES

- **Title:** Scoreboard Accesses
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd2
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-156. Unit Masks for SB_ACCESSES

Extension	umask [15:8]	Description
RD_ACCEPTS	bxxxxxxx1	Reads Accepted
RD_REJECTS	bxxxxxx1x	Reads Rejected
WR_ACCEPTS	bxxxxx1xx	Writes Accepted
ACCEPTS	b00000101	Scoreboard Accesses Accepted
WR_REJECTS	bxxxx1xxx	Writes Rejected
REJECTS	b00001010	Scoreboard Accesses Rejected
NM_RD_CMPS	bxxx1xxxx	Near Mem read completions
NM_WR_CMPS	bxx1xxxxx	Near Mem write completions
FM_RD_CMPS	bx1xxxxxx	Far Mem read completions
FM_WR_CMPS	b1xxxxxxx	Far Mem write completions

SB_CANARY

- **Title:**
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd9
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-157. Unit Masks for SB_CANARY

Extension	umask [15:8]	Description
ALLOC	bxxxxxxx1	Alloc
DEALLOC	bxxxxxx1x	Dealloc
VLD	bxxxxx1xx	Valid
NM_RD_STARVED	bxxxx1xxx	Near Mem Reads Starved
NM_WR_STARVED	bxxx1xxxx	Near Mem Writes Starved
FM_RD_STARVED	bxx1xxxxx	Far Mem Reads Starved
FM_WR_STARVED	bx1xxxxxx	Far Mem Writes Starved
FM_TGR_WR_STARVED	b1xxxxxxx	Far Mem TGR Writes Starved

SB_CYCLES_FULL

- **Title:** Scoreboard Cycles Full
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd1
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

SB_CYCLES_NE

- **Title:** Scoreboard Cycles Not-Empty
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd0
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

SB_INSERTS

- **Title:** Scoreboard Inserts
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd6
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-158. Unit Masks for SB_INSERTS

Extension	umask [15:8]	Description
RDS	bxxxxxxx1	Reads
WRS	bxxxxxx1x	Writes
PMM_RDS	bxxxxx1xx	Persistent Mem reads
PMM_WRS	bxxxx1xxx	Persistent Mem writes
BLOCK_RDS	bxxx1xxxx	Block region reads
BLOCK_WRS	bxx1xxxxx	Block region writes

SB_OCCUPANCY

- **Title:** Scoreboard Occupancy
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd5
- **Max. Inc/Cyc:.** 128

Register Restrictions: 0-3

- **Definition:**

Table 2-159. Unit Masks for SB_OCCUPANCY

Extension	umask [15:8]	Description
RDS	bxxxxxxx1	Reads
PMM_RDS	bxxxxx1xx	Persistent Mem reads

Table 2-159. Unit Masks for SB_OCCUPANCY

Extension	umask [15:8]	Description
PMM_WRS	bxxxx1xxx	Persistent Mem writes
BLOCK_RDS	bxx1xxxxx	Block region reads
BLOCK_WRS	bx1xxxxxx	Block region writes

SB_PREF_INSERTS

- **Title:** Scoreboard Prefetch Inserts
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xda
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-160. Unit Masks for SB_PREF_INSERTS

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All
DDR	bxxxxxx1x	DDR4
PMM	bxxxxx1xx	Persistent Mem

SB_PREF_OCCUPANCY

- **Title:** Scoreboard Prefetch Occupancy
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xdb
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-161. Unit Masks for SB_PREF_OCCUPANCY

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All
DDR	bxxxxxx1x	DDR4
PMM	bxxxxx1xx	Persistent Mem

SB_REJECT

- **Title:** Number of Scoreboard Requests Rejected
- **Category:** PMM MEMMODE COHERENCY Events
- **Event Code:** 0xd4
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-162. Unit Masks for SB_REJECT

Extension	umask [15:8]	Description
NM_SET_CNFLT	bxxxxxxx1	NM requests rejected due to set conflict
FM_ADDR_CNFLT	bxxxxxx1x	FM requests rejected due to full address conflict
PATROL_SET_CNFLT	bxxxxx1xx	Patrol requests rejected due to set conflict
CANARY	bxxxx1xxx	
DDR_EARLY_CMP	bxx1xxxxx	

SB_STRV_ALLOC

- **Title:**
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd7
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-163. Unit Masks for SB_STRV_ALLOC

Extension	umask [15:8]	Description
NM_RD	bxxxxxxx1	Near Mem Read - Set
FM_RD	bxxxxxx1x	Far Mem Read - Set
NM_WR	bxxxxx1xx	Near Mem Write - Set
FM_WR	bxxxx1xxx	Far Mem Write - Set
FM_TGR	bxxx1xxxx	Far Mem TGR

SB_STRV_DEALLOC

- **Title:**
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xde
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-164. Unit Masks for SB_STRV_DEALLOC

Extension	umask [15:8]	Description
NM_RD	bxxxxxxx1	Near Mem Read - Set
FM_RD	bxxxxxx1x	Far Mem Read - Set
NM_WR	bxxxxx1xx	Near Mem Write - Set
FM_WR	bxxxx1xxx	Far Mem Write - Set
FM_TGR	bxxx1xxxx	Far Mem TGR

SB_STRV_OCC

- **Title:**
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xd8
- **Max. Inc/Cyc:.** 63

Register Restrictions: 0-3

- **Definition:**

Table 2-165. Unit Masks for SB_STRV_OCC

Extension	umask [15:8]	Description
NM_RD	bxxxxxxx1	Near Mem Read
FM_RD	bxxxxxx1x	Far Mem Read
NM_WR	bxxxxx1xx	Near Mem Write
FM_WR	bxxxx1xxx	Far Mem Write
FM_TGR	bxxx1xxxx	Far Mem TGR

SB_TAGGED

- **Title:**
- **Category:** PMM MEMMODE SCOREBOARD Events
- **Event Code:** 0xdd
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-166. Unit Masks for SB_TAGGED

Extension	umask [15:8]	Description
NEW	bxxxxxxx1	
RD_HIT	bxxxxxx1x	
RD_MISS	bxxxxx1xx	
DDR4_CMP	bxxxx1xxx	
PMM0_CMP	bxxx1xxxx	
PMM1_CMP	bxx1xxxxx	
PMM2_CMP	bx1xxxxxx	
OCC	b1xxxxxxx	

TAGCHK

- **Title:** 2LM Tag Check
- **Category:** TAG CHECK Events
- **Event Code:** 0xd3
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-167. Unit Masks for TAGCHK

Extension	umask [15:8]	Description
HIT	bxxxxxxx1	Hit in Near Memory Cache
MISS_CLEAN	bxxxxxx1x	Miss, no data in this line
MISS_DIRTY	bxxxxx1xx	Miss, existing data may be evicted to Far Memory
NM_RD_HIT	bxxxx1xxx	Read Hit in Near Memory Cache
NM_WR_HIT	bxxx1xxxx	Write Hit in Near Memory Cache

WPQ_CYCLES_FULL_PCH0

- **Title:** Write Pending Queue Full Cycles
- **Category:** WPQ Events
- **Event Code:** 0x22
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the Write Pending Queue is full. When the WPQ is full, the HA will not be able to issue any additional write requests into the iMC. This count should be similar count in the CHA which tracks the number of cycles that the CHA has no WPQ credits, just somewhat smaller to account for the credit return overhead.

WPQ_CYCLES_FULL_PCH1

- **Title:** Write Pending Queue Full Cycles
- **Category:** WPQ Events
- **Event Code:** 0x16
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the Write Pending Queue is full. When the WPQ is full, the HA will not be able to issue any additional write requests into the iMC. This count should be similar count in the CHA which tracks the number of cycles that the CHA has no WPQ credits, just somewhat smaller to account for the credit return overhead.

WPQ_CYCLES_NE

- **Title:** Write Pending Queue Not Empty
- **Category:** WPQ Events
- **Event Code:** 0x21
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles that the Write Pending Queue is not empty. This can then be used to calculate the average queue occupancy (in conjunction with the WPQ Occupancy Accumulation count). The WPQ is used to schedule write out to the memory controller and to track the writes. Requests allocate into the WPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the CHA to the iMC. They deallocate after being issued to DRAM. Write requests themselves are able to complete (from the perspective of the rest of the system) as soon they have "posted" to the iMC. This is not to be confused with actually performing the write to DRAM. Therefore, the average latency for this queue is actually not useful for deconstruction intermediate write latencies.

Table 2-168. Unit Masks for WPQ_CYCLES_NE

Extension	umask [15:8]	Description
PCH0	bxxxxxxx1	
PCH1	bxxxxxx1x	

WPQ_INSERTS

- **Title:** Write Pending Queue Allocations
- **Category:** WPQ Events
- **Event Code:** 0x20
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of allocations into the Write Pending Queue. This can then be used to calculate the average queuing latency (in conjunction with the WPQ occupancy count). The WPQ is used to schedule write out to the memory controller and to track the writes. Requests allocate into the WPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the CHA to the iMC. They deallocate after being issued to DRAM. Write requests themselves are able to complete (from the perspective of the rest of the system) as soon they have "posted" to the iMC.

Table 2-169. Unit Masks for WPQ_INSERTS

Extension	umask [15:8]	Description
PCH0	bxxxxxxx1	
PCH1	bxxxxxx1x	

WPQ_OCCUPANCY_PCH0

- **Title:** Write Pending Queue Occupancy
- **Category:** WPQ Events
- **Event Code:** 0x82
- **Max. Inc/Cyc:.** 40

Register Restrictions: 0-3

- **Definition:** Accumulates the occupancies of the Write Pending Queue each cycle. This can then be used to calculate both the average queue occupancy (in conjunction with the number of cycles not empty) and the average latency (in conjunction with the number of allocations). The WPQ is used to schedule write out to the memory controller and to track the writes. Requests allocate into the WPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the HA to the iMC. They deallocate after being issued to DRAM. Write requests themselves are able to complete (from the perspective of the rest of the system) as soon they have "posted" to the iMC. This is not to be confused with actually performing the write to DRAM. Therefore, the average latency for this queue is actually not useful for deconstruction intermediate write latencies. So, we provide filtering based on if the request has posted or not. By using the "not posted" filter, we can track how long writes spent in the iMC before completions were sent to the HA. The "posted" filter, on the other hand, provides information about how much queueing is actually happenning in the iMC for writes before they are actually issued to memory. High average occupancies will generally coincide with high write major mode counts.

WPQ_OCCUPANCY_PCH1

- **Title:** Write Pending Queue Occupancy
- **Category:** WPQ Events
- **Event Code:** 0x83
- **Max. Inc/Cyc:.** 40

Register Restrictions: 0-3

- **Definition:** Accumulates the occupancies of the Write Pending Queue each cycle. This can then be used to calculate both the average queue occupancy (in conjunction with the number of cycles not empty) and the average latency (in conjunction with the number of allocations). The WPQ is used to schedule write out to the memory controller and to track the writes. Requests allocate into the WPQ soon after they enter the memory controller, and need credits for an entry in this buffer before being sent from the HA to the iMC. They deallocate after being issued to DRAM. Write requests themselves are able to complete (from the perspective of the rest of the system) as soon they have "posted" to the iMC. This is not to be confused with actually performing the write to DRAM. Therefore, the average latency for this queue is actually not useful for deconstruction intermediate write latencies. So, we provide filtering based on if the request has posted or not. By using the "not posted" filter, we can track how long writes spent in the iMC before completions were sent to the HA. The "posted" filter, on the other hand, provides information about how much queueing is actually happenning in the iMC for writes before they are actually issued to memory. High average occupancies will generally coincide with high write major mode counts.

WPQ_READ_HIT

- **Title:** Write Pending Queue CAM Match
- **Category:** WPQ Events
- **Event Code:** 0x23
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times a request hits in the WPQ (write-pending queue). The iMC allows writes and reads to pass up other writes to different addresses. Before a read or a write is issued, it will first CAM the WPQ to see if there is a write pending to that address. When reads hit, they are able to directly pull their data from the WPQ instead of going to memory. Writes that hit will overwrite the existing data. Partial writes that hit will not need to do underfill reads and will simply update their relevant sections.

Table 2-170. Unit Masks for WPQ_READ_HIT

Extension	umask [15:8]	Description
PCH0	bxxxxxxx1	
PCH1	bxxxxxx1x	

WPQ_WRITE_HIT

- **Title:** Write Pending Queue CAM Match
- **Category:** WPQ Events
- **Event Code:** 0x24
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times a request hits in the WPQ (write-pending queue). The iMC allows writes and reads to pass up other writes to different addresses. Before a read or a write is issued, it will first CAM the WPQ to see if there is a write pending to that address. When reads hit, they are able to directly pull their data from the WPQ instead of going to memory. Writes that hit will overwrite the

existing data. Partial writes that hit will not need to do underfill reads and will simply update their relevant sections.

Table 2-171. Unit Masks for WPQ_WRITE_HIT

Extension	umask [15:8]	Description
PCH0	bxxxxxxx1	
PCH1	bxxxxxx1x	

2.5 IIO Performance Monitoring

IIO stacks are responsible for managing traffic between the PCIe domain and the Mesh domain. The IIO PMON block is situated near the IIO stack's traffic controller capturing traffic controller as well as PCIe root port information. The traffic controller is responsible for translating traffic coming in from the Mesh (through M2PCIe) and processed by IRP into the PCIe domain to IO agents such as CBDMA, DMA and PCIe.

The following table should help map the events and the free running BW counters found in each M2IOSF block to the component that M2IOSF block is servicing.

Table 2-172. 3rd Gen Intel Xeon Processor Scalable Family M2IOSF Mapping

M2IOSF 0		M2IOSF1		M2IOSF2		M2IOSF2		M2IOSF3		M2IOSF4	
PORT0	PORT1	PORT0	PORT1	PORT0	PORT1	PORT0	PORT1	PORT0	PORT1	PORT0	PORT1
PCIe4 x16	NONE	PCIe4 x16	NONE	PCIe4 x16	NONE	PCIe4 x16	NONE	N/A	N/A	CBDMA/ DMI/ PCIe3	NONE
PART 0-3	PART 4-7	PART 0-3	PART 4-7	PART 0-3	PART 4-7	PART 0-3	PART 4-7	N/A	N/A	PART 0-3	PART 4-7

2.5.1 IIO Performance Monitoring Overview

Each IIO Box, which sits near the IIO stack's Traffic Controller, supports event monitoring through four 48b wide counters (IIO{5-0}_MSR_PMON_CTR/CTL{3:0}). Each of these counters can be programmed to count any IIO event. The IIO counters can increment by a maximum of 7b per cycle.

2.5.2 Additional IIO Performance Monitoring

2.5.2.1 IIO PMON Counter Control - Difference from Baseline

IIO performance monitoring control registers provide a small amount of additional functionality. The following table defines those cases.

Figure 2-7. IIO Counter Control Register for 3rd Gen Intel Xeon Processor Scalable Family Server

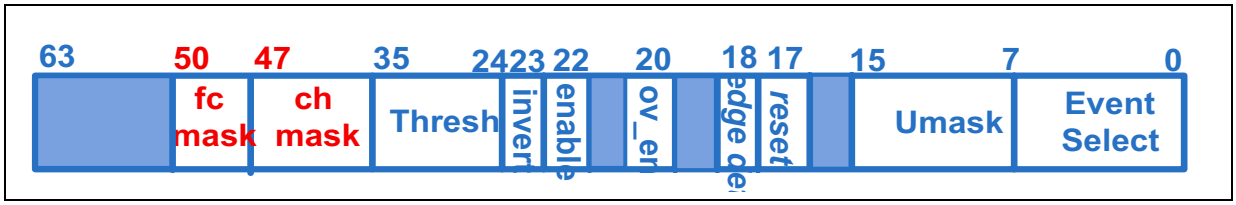


Table 2-173. IIOn_MSR_PMON_CTL{3-0} Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
rsv	63:47	RV	0	Reserved. SW must write to 0 else behavior is undefined.
fc_mask	50:48	RW-V	0	FC Mask - applicable to certain events (Filter - fc) 0 - Posted Requests 1 - Non-posted Requests 2 - Completions
ch_mask	47:36	RW-V	0	Channel Mask Filter - applicable to certain events (Filter - chnl)
thresh	35:24	RW-V	0	Threshold used in counter comparison.

There are a number of free-running counters, providing information highly valuable to a wide array of customers, in each IIO Stack that collect counts for Input/Output Bandwidth for each Port.

'Free Running' counters cannot be changed by SW operating in a normal environment. SW cannot write them, cannot stop them and cannot reset the values.

Note: Counting will be suspended when the IIO stack is powered down.

There is one register per stack to track the number of IIO cycles

Table 2-174. IIO_MSR_PMON_FRCTR_IOCLK Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:48	RV	0	Ignored
event_count	47:0	RO-V	0	48-bit running count of IO clocks

And two extra registers per port to track

- **Inbound (PCIe -> CPU) bandwidth** - counts 32 bytes of data, associated with writes and completions, transmitted from the IO stack to the traffic controller

Table 2-175. IIO_MSR_PMON_FRCTR_BW_IN_P{0-7} Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:36	RV	0	Ignored
event_count	47:0	RO-V	0	36-bit running count of data bytes transmitted from link for this port.

- **Output (CPU -> PCIe) bandwidth** - counts 32 bytes of data, associated with writes and completions, transmitted from the traffic controller to the IO stack

Table 2-176. IIO_MSR_PMON_FRCTR_BW_OUT_P{0-7} Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:36	RV	0	Ignored
event_count	47:0	RO-V	0	36-bit running count of data bytes transmitted link for this port.

2.5.3 IIO Performance Monitoring Events

IIO provides events to track information related to all the traffic passing through it's boundaries.

- Bandwidth consumed and Transactions processed broken down by transaction type
- Per Port Utilization
- Link Power States
- Completion Buffer tracking

2.5.4 IIO Box Events Ordered By Code

The following table summarizes the directly measured IIO Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
MASK_MATCH_AND	0x02	0-1	0	AND Mask/match for debug bus
MASK_MATCH_OR	0x03	0-1	0	OR Mask/match for debug bus
IOMMU0	0x40		0	
IOMMU1	0x41		0	
PWT_OCCUPANCY	0x42		0	PWT occupancy
IOMMU3	0x43		0	
NOTHING	0x80	0-3	1	Counting disabled
CLOCKTICKS	0x81	0-3	1	Clockticks of the integrated IO (IIO) traffic controller
SYMBOL_TIMES	0x82		0	Symbol Times on Link
DATA_REQ_OF_CPU	0x83	0-1	1024	Data requested of the CPU
TXN_REQ_OF_CPU	0x84	0-3	1	Number Transactions requested of the CPU
NUM_REQ_OF_CPU	0x85	0-3	1	Number requests PCIe makes of the main die
INBOUND_ARB_REQ	0x86	0-3	1	Incoming arbitration requests
INBOUND_ARB_WON	0x87	0-3	1	Incoming arbitration requests granted
NUM_OUTSTANDING_REQ_OF_CPU	0x88	2-3	512	
NUM_REQ_OF_CPU_BY_TGT	0x8e	0-3	1	Num requests sent by PCIe - by target
NUM_TGT_MATCHED_REQ_OF_CPU	0x8f	0-3	1	ITC address map 1
REQ_FROM_PCIE_PASS_CMPL	0x90	0-3	1	PCIe Request - pass complete
REQ_FROM_PCIE_CL_CMPL	0x91	0-3	1	PCIe Request - cacheline complete
REQ_FROM_PCIE_CMPL	0x92	0-3	1	PCIe Request complete

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
DATA_REQ_BY_CPU	0xc0	2-3	1024	Data requested by the CPU
TXN_REQ_BY_CPU	0xc1	0-3	1	Number Transactions requested by the CPU
COMP_BUF_INSERTS	0xc2	0-3	1	PCIe Completion Buffer Inserts
NUM_REQ_FROM_CPU	0xc2	0-3	1	Number requests sent to PCIe from main die
NUM_OUSTANDING_REQ_FROM_CPU	0xc5	2-3	256	Occupancy of outbound request queue
OUTBOUND_CL_REQS_ISSUED	0xd0	0-3	1	Outbound cacheline requests issued
OUTBOUND_TLP_REQS_ISSUED	0xd1	0-3	1	Outbound TLP (transaction layer packet) requests issued
COMP_BUF_OCCUPANCY	0xd5	2-3		PCIe Completion Buffer Occupancy

2.5.5 IIO Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the IIO Box.

CLOCKTICKS

- **Title:** Clockticks of the integrated IO (IIO) traffic controller
- **Category:** CLOCK Events
- **Event Code:** 0x81
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Increments counter once every Traffic Controller clock, the LSCLK (500MHz)

COMP_BUF_INSERTS

- **Title:** PCIe Completion Buffer Inserts
- **Category:** PCIe Completion Buffer Events
- **Event Code:** 0xc2
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**
- **NOTE:** aka OTC_ENQ.PREALLOC? (May need to redo)

Table 2-177. Unit Masks for COMP_BUF_INSERTS

Extension	umask [15:8]	xtra [50:36]	Description
CMPD.PART4	bxxxxxx11	b100 bxxxxxxx1xxxx	Part 4 x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CMPD.PART2	bxxxxxx11	b100 bxxxxxxxxx1xx	Part 2 x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CMPD.PART6	bxxxxxx11	b100 bxxxxx1xxxxxx	Part 6 x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6

Table 2-177. Unit Masks for COMP_BUF_INSERTS

Extension	umask [15:8]	xtra [50:36]	Description
CMPD.PART0	bxxxxxx11	b100 bxxxxxxxxx xx1	Part 0 x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CMPD.PART7	bxxxxxx11	b100 bxxxx1xxxx xxx	Part 7 x4 card is plugged in to slot 7
CMPD.PART5	bxxxxxx11	b100 bxxxxxx1xx xxx	Part 5 x4 card is plugged in to slot 5
CMPD.PART3	bxxxxxx11	b100 bxxxxxxxx1 xxx	Part 3 x4 card is plugged in to slot 3
CMPD.PART1	bxxxxxx11	b100 bxxxxxxxxx x1x	Part 1 x4 card is plugged in to slot 1
CMPD.ALL	bxxxxxx11	b100 bxxxx1111 1111	All Ports

COMP_BUF_OCCUPANCY

- **Title:** PCIe Completion Buffer Occupancy
- **Category:** PCIe Completion Buffer Events
- **Event Code:** 0xd5
- **Max. Inc/Cyc:.**

Register Restrictions: 2-3

- **Definition:**

Table 2-178. Unit Masks for COMP_BUF_OCCUPANCY (Sheet 1 of 2)

Extension	umask [15:8]	xtra [50:36]	Description
CMPD.PART0	bxxxxxxx1	b100 bxxxxxxxxx xxx	Part 0 x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CMPD.PART1	bxxxxxx1x	b100 bxxxxxxxxx xxx	Part 1 x4 card is plugged in to slot 1
CMPD.PART2	bxxxxx1xx	b100 bxxxxxxxxx xxx	Part 2 x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CMPD.PART3	bxxxx1xxx	b100 bxxxxxxxxx xxx	Part 3 x4 card is plugged in to slot 3
CMPD.PART4	bxxx1xxxx	b100 bxxxxxxxxx xxx	Part 4 x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CMPD.PART5	bxx1xxxxx	b100 bxxxxxxxxx xxx	Part 5 x4 card is plugged in to slot 5
CMPD.PART6	bx1xxxxxx	b100 bxxxxxxxxx xxx	Part 6 x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6

Table 2-178. Unit Masks for COMP_BUF_OCCUPANCY (Sheet 2 of 2)

Extension	umask [15:8]	xtra [50:36]	Description
CMPD.PART7	b1xxxxxxx	b100 bxxxxxxxxxxxx	Part 7 x4 card is plugged in to slot 7
CMPD.ALL	b11111111	b100 bxxxxxxxxxxxx	All Ports

DATA_REQ_BY_CPU

- **Title:** Data requested by the CPU
- **Category:** Payload Events
- **Event Code:** 0xc0
- **Max. Inc/Cyc:.** 1024

Register Restrictions: 2-3

- **Definition:** Number of DWs (4 bytes) requested by the main die. Includes all requests initiated by the main die, including reads and writes.
- **NOTE:** Unlike free running counters, Mem Read and Peer read subevents count requests not completions. Peer R/W subevents do not include confined P2P traffic. Peer R/W subevents are also counted as Mem R/W subevents.

Table 2-179. Unit Masks for DATA_REQ_BY_CPU (Sheet 1 of 4)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_WRITE.IOMMU0	bxxxxxxx1	b111 bxxx1xxxxxxx	Core writing to Card's MMIO space IOMMU - Type 0
MEM_WRITE.PART1	bxxxxxxx1	b111 bxxxxxxxxx1x	Core writing to Card's MMIO space x4 card is plugged in to slot 1
MEM_WRITE.PART6	bxxxxxxx1	b111 bxxxxx1xxxxx	Core writing to Card's MMIO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MEM_WRITE.PART7	bxxxxxxx1	b111 bxxxx1xxxxxx	Core writing to Card's MMIO space x4 card is plugged in to slot 7
MEM_WRITE.PART5	bxxxxxxx1	b111 bxxxxx1xxxxx	Core writing to Card's MMIO space x4 card is plugged in to slot 5
MEM_WRITE.PART2	bxxxxxxx1	b111 bxxxxxxxx1xx	Core writing to Card's MMIO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MEM_WRITE.IOMMU1	bxxxxxxx1	b111 bxx1xxxxxxxx	Core writing to Card's MMIO space IOMMU - Type 1
MEM_WRITE.PART3	bxxxxxxx1	b111 bxxxxxxx1xxx	Core writing to Card's MMIO space x4 card is plugged in to slot 3
MEM_WRITE.PART0	bxxxxxxx1	b111 bxxxxxxxxxx1	Core writing to Card's MMIO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_WRITE.PART4	bxxxxxxx1	b111 bxxxxxx1xxxx	Core writing to Card's MMIO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_WRITE.PART1	bxxxxxx1x	b111 bxxxxxxxxx1x	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 1
PEER_WRITE.IOMMU0	bxxxxxx1x	b111 bxxx1xxxxxxx	Another card (different IIO stack) writing to this card. IOMMU - Type 0
PEER_WRITE.PART7	bxxxxxx1x	b111 bxxxx1xxxxxx	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 7
PEER_WRITE.PART6	bxxxxxx1x	b111 bxxxxx1xxxxx	Another card (different IIO stack) writing to this card. x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6

Table 2-179. Unit Masks for DATA_REQ_BY_CPU (Sheet 2 of 4)

Extension	umask [15:8]	xtra [50:36]	Description
PEER_WRITE.PART5	bxxxxxx1x	b111 bxxxxxx1xxxxx	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 5
PEER_WRITE.PART2	bxxxxxx1x	b111 bxxxxxxxxx1xx	Another card (different IIO stack) writing to this card. x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_WRITE.IOMMU1	bxxxxxx1x	b111 bxx1xxxxxxxxx	Another card (different IIO stack) writing to this card. IOMMU - Type 1
PEER_WRITE.PART3	bxxxxxx1x	b111 bxxxxxxxx1xxx	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 3
PEER_WRITE.PART4	bxxxxxx1x	b111 bxxxxxxx1xxxx	Another card (different IIO stack) writing to this card. x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_WRITE.PART0	bxxxxxx1x	b111 bxxxxxxxxxxx1	Another card (different IIO stack) writing to this card. x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_READ.IOMMU0	bxxxxx1xx	b111 bxxx1xxxxxxxx	Core reporting completion of Card read from Core DRAM IOMMU - Type 0
MEM_READ.PART3	bxxxxx1xx	b111 bxxxxxxxx1xxx	Core reporting completion of Card read from Core DRAM x4 card is plugged in to slot 3
MEM_READ.PART6	bxxxxx1xx	b111 bxxxxx1xxxxxx	Core reporting completion of Card read from Core DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MEM_READ.PART5	bxxxxx1xx	b111 bxxxxxx1xxxxx	Core reporting completion of Card read from Core DRAM x4 card is plugged in to slot 5
MEM_READ.PART0	bxxxxx1xx	b111 bxxxxxxxxxxx1	Core reporting completion of Card read from Core DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_READ.PART1	bxxxxx1xx	b111 bxxxxxxxxxx1x	Core reporting completion of Card read from Core DRAM x4 card is plugged in to slot 1
MEM_READ.IOMMU1	bxxxxx1xx	b111 bxx1xxxxxxxxx	Core reporting completion of Card read from Core DRAM IOMMU - Type 1
MEM_READ.PART2	bxxxxx1xx	b111 bxxxxxxxxx1xx	Core reporting completion of Card read from Core DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MEM_READ.PART7	bxxxxx1xx	b111 bxxxx1xxxxxx	Core reporting completion of Card read from Core DRAM x4 card is plugged in to slot 7
MEM_READ.PART4	bxxxxx1xx	b111 bxxxxxxx1xxxx	Core reporting completion of Card read from Core DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_READ.IOMMU1	bxxxx1xxx	b111 bxx1xxxxxxxxx	Another card (different IIO stack) reading from this card. IOMMU - Type 1
PEER_READ.PART3	bxxxx1xxx	b111 bxxxxxxxx1xxx	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 3
PEER_READ.PART0	bxxxx1xxx	b111 bxxxxxxxxxxx1	Another card (different IIO stack) reading from this card. x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_READ.PART2	bxxxx1xxx	b111 bxxxxxxxxx1xx	Another card (different IIO stack) reading from this card. x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_READ.PART5	bxxxx1xxx	b111 bxxxxxx1xxxxx	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 5
PEER_READ.PART4	bxxxx1xxx	b111 bxxxxxxx1xxxx	Another card (different IIO stack) reading from this card. x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_READ.PART1	bxxxx1xxx	b111 bxxxxxxxxxx1x	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 1
PEER_READ.IOMMU0	bxxxx1xxx	b111 bxxx1xxxxxxxx	Another card (different IIO stack) reading from this card. IOMMU - Type 0

Table 2-179. Unit Masks for DATA_REQ_BY_CPU (Sheet 3 of 4)

Extension	umask [15:8]	xtra [50:36]	Description
PEER_READ.PART7	bxxxx1xxx	b111 bxxxx1xxxxxxx	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 7
PEER_READ.PART6	bxxxx1xxx	b111 bxxxxx1xxxxxx	Another card (different IIO stack) reading from this card. x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
CFG_WRITE.IOMMU0	bxxx1xxxx	b111 bxxx1xxxxxxxx	Core writing to Card's PCICFG space IOMMU - Type 0
CFG_WRITE.PART1	bxxx1xxxx	b111 bxxxxxxxxxx1x	Core writing to Card's PCICFG space x4 card is plugged in to slot 1
CFG_WRITE.PART6	bxxx1xxxx	b111 bxxxxx1xxxxxx	Core writing to Card's PCICFG space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
CFG_WRITE.PART7	bxxx1xxxx	b111 bxxxx1xxxxxxx	Core writing to Card's PCICFG space x4 card is plugged in to slot 7
CFG_WRITE.PART2	bxxx1xxxx	b111 bxxxxxxxxx1xx	Core writing to Card's PCICFG space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CFG_WRITE.PART5	bxxx1xxxx	b111 bxxxxxx1xxxxx	Core writing to Card's PCICFG space x4 card is plugged in to slot 5
CFG_WRITE.IOMMU1	bxxx1xxxx	b111 bxx1xxxxxxxxx	Core writing to Card's PCICFG space IOMMU - Type 1
CFG_WRITE.PART3	bxxx1xxxx	b111 bxxxxxxx1xxx	Core writing to Card's PCICFG space x4 card is plugged in to slot 3
CFG_WRITE.PART0	bxxx1xxxx	b111 bxxxxxxxxxxx1	Core writing to Card's PCICFG space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CFG_WRITE.PART4	bxxx1xxxx	b111 bxxxxxxx1xxxx	Core writing to Card's PCICFG space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
IO_WRITE.PART3	bxx1xxxxx	b111 bxxxxxxx1xxx	Core writing to Card's IO space x4 card is plugged in to slot 3
IO_WRITE.IOMMU0	bxx1xxxxx	b111 bxxx1xxxxxxxx	Core writing to Card's IO space IOMMU - Type 0
IO_WRITE.PART0	bxx1xxxxx	b111 bxxxxxxxxxxx1	Core writing to Card's IO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
IO_WRITE.PART6	bxx1xxxxx	b111 bxxxxx1xxxxxx	Core writing to Card's IO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
IO_WRITE.PART5	bxx1xxxxx	b111 bxxxxxx1xxxxx	Core writing to Card's IO space x4 card is plugged in to slot 5
IO_WRITE.PART2	bxx1xxxxx	b111 bxxxxxxxxx1xx	Core writing to Card's IO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
IO_WRITE.PART7	bxx1xxxxx	b111 bxxxx1xxxxxxx	Core writing to Card's IO space x4 card is plugged in to slot 7
IO_WRITE.PART1	bxx1xxxxx	b111 bxxxxxxxxxx1x	Core writing to Card's IO space x4 card is plugged in to slot 1
IO_WRITE.IOMMU1	bxx1xxxxx	b111 bxx1xxxxxxxxx	Core writing to Card's IO space IOMMU - Type 1
IO_WRITE.PART4	bxx1xxxxx	b111 bxxxxxxx1xxxx	Core writing to Card's IO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CFG_READ.IOMMU0	bx1xxxxxx	b111 bxxx1xxxxxxxx	Core reading from Card's PCICFG space IOMMU - Type 0
CFG_READ.PART3	bx1xxxxxx	b111 bxxxxxxxx1xxx	Core reading from Card's PCICFG space x4 card is plugged in to slot 3
CFG_READ.PART6	bx1xxxxxx	b111 bxxxxx1xxxxxx	Core reading from Card's PCICFG space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6

Table 2-179. Unit Masks for DATA_REQ_BY_CPU (Sheet 4 of 4)

Extension	umask [15:8]	xtra [50:36]	Description
CFG_READ.PART5	bx1xxxxxx	b111 bxxxxxx1xxxxx	Core reading from Card's PCICFG space x4 card is plugged in to slot 5
CFG_READ.PART0	bx1xxxxxx	b111 bxxxxxxxxxxx1	Core reading from Card's PCICFG space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CFG_READ.PART1	bx1xxxxxx	b111 bxxxxxxxxxx1x	Core reading from Card's PCICFG space x4 card is plugged in to slot 1
CFG_READ.IOMMU1	bx1xxxxxx	b111 bxx1xxxxxxxxx	Core reading from Card's PCICFG space IOMMU - Type 1
CFG_READ.PART2	bx1xxxxxx	b111 bxxxxxxxxx1xx	Core reading from Card's PCICFG space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CFG_READ.PART7	bx1xxxxxx	b111 bxxxx1xxxxxxx	Core reading from Card's PCICFG space x4 card is plugged in to slot 7
CFG_READ.PART4	bx1xxxxxx	b111 bxxxxxxx1xxxx	Core reading from Card's PCICFG space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
IO_READ.PART7	b1xxxxxxx	b111 bxxxx1xxxxxxx	Core reading from Card's IO space x4 card is plugged in to slot 7
IO_READ.PART0	b1xxxxxxx	b111 bxxxxxxxxxxx1	Core reading from Card's IO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
IO_READ.PART4	b1xxxxxxx	b111 bxxxxxxx1xxxx	Core reading from Card's IO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
IO_READ.PART6	b1xxxxxxx	b111 bxxxxx1xxxxxx	Core reading from Card's IO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
IO_READ.PART2	b1xxxxxxx	b111 bxxxxxxxxx1xx	Core reading from Card's IO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
IO_READ.IOMMU0	b1xxxxxxx	b111 bxxx1xxxxxxxx	Core reading from Card's IO space IOMMU - Type 0
IO_READ.IOMMU1	b1xxxxxxx	b111 bxx1xxxxxxxx	Core reading from Card's IO space IOMMU - Type 1
IO_READ.PART5	b1xxxxxxx	b111 bxxxxxx1xxxxx	Core reading from Card's IO space x4 card is plugged in to slot 5
IO_READ.PART1	b1xxxxxxx	b111 bxxxxxxxxxx1x	Core reading from Card's IO space x4 card is plugged in to slot 1
IO_READ.PART3	b1xxxxxxx	b111 bxxxxxxxx1xxx	Core reading from Card's IO space x4 card is plugged in to slot 3

DATA_REQ_OF_CPU

- **Title:** Data requested of the CPU
- **Category:** Payload Events
- **Event Code:** 0x83
- **Max. Inc/Cyc:.** 1024

Register Restrictions: 0-1

- **Definition:** Number of DWs (4 bytes) the card requests of the main die. Includes all requests initiated by the Card, including reads and writes.
- **NOTE:** Multiply value by 4 to convert to number of Bytes. Unlike free running counters, Mem Read and Peer read subevents count requests not completions. Unlike the *_BY_CPU.PEER* events, peer R/W subevents do include confined P2P traffic. Counts are incremented on the request path rather than the completion path. Expect to change back on next prodcut.

Table 2-180. Unit Masks for DATA_REQ_OF_CPU (Sheet 1 of 3)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_WRITE.PART2	bxxxxxxx1	b111 bxxxxxxxxx1xx	Card writing to DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MEM_WRITE.PART5	bxxxxxxx1	b111 bxxxxxx1xxxxx	Card writing to DRAM x4 card is plugged in to slot 5
MEM_WRITE.IOMMU0	bxxxxxxx1	b111 bxxx1xxxxxxxx	Card writing to DRAM IOMMU - Type 0
MEM_WRITE.PART1	bxxxxxxx1	b111 bxxxxxxxxxx1x	Card writing to DRAM x4 card is plugged in to slot 1
MEM_WRITE.IOMMU1	bxxxxxxx1	b111 bxx1xxxxxxxxx	Card writing to DRAM IOMMU - Type 1
MEM_WRITE.PART3	bxxxxxxx1	b111 bxxxxxxxx1xxx	Card writing to DRAM x4 card is plugged in to slot 3
MEM_WRITE.PART0	bxxxxxxx1	b111 bxxxxxxxxxxx1	Card writing to DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_WRITE.PART6	bxxxxxxx1	b111 bxxxxx1xxxxxx	Card writing to DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MEM_WRITE.PART4	bxxxxxxx1	b111 bxxxxxxx1xxxx	Card writing to DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MEM_WRITE.PART7	bxxxxxxx1	b111 bxxxx1xxxxxx	Card writing to DRAM x4 card is plugged in to slot 7
PEER_WRITE.PART7	bxxxxxx1x	b111 bxxxx1xxxxxx	Card writing to another Card (same or different stack) x4 card is plugged in to slot 7
PEER_WRITE.PART5	bxxxxxx1x	b111 bxxxxxx1xxxxx	Card writing to another Card (same or different stack) x4 card is plugged in to slot 5
PEER_WRITE.PART6	bxxxxxx1x	b111 bxxxxx1xxxxxx	Card writing to another Card (same or different stack) x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_WRITE.IOMMU0	bxxxxxx1x	b111 bxxx1xxxxxxxx	Card writing to another Card (same or different stack) IOMMU - Type 0
PEER_WRITE.PART1	bxxxxxx1x	b111 bxxxxxxxxxx1x	Card writing to another Card (same or different stack) x4 card is plugged in to slot 1
PEER_WRITE.PART2	bxxxxxx1x	b111 bxxxxxxxxx1xx	Card writing to another Card (same or different stack) x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_WRITE.IOMMU1	bxxxxxx1x	b111 bxx1xxxxxxxxx	Card writing to another Card (same or different stack) IOMMU - Type 1
PEER_WRITE.PART4	bxxxxxx1x	b111 bxxxxxxx1xxxx	Card writing to another Card (same or different stack) x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_WRITE.PART0	bxxxxxx1x	b111 bxxxxxxxxxxx1	Card writing to another Card (same or different stack) x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_WRITE.PART3	bxxxxxx1x	b111 bxxxxxxxx1xxx	Card writing to another Card (same or different stack) x4 card is plugged in to slot 3
MEM_READ.PART1	bxxxxx1xx	b111 bxxxxxxxxxx1x	Card reading from DRAM x4 card is plugged in to slot 1
MEM_READ.IOMMU0	bxxxxx1xx	b111 bxxx1xxxxxxxx	Card reading from DRAM IOMMU - Type 0
MEM_READ.IOMMU1	bxxxxx1xx	b111 bxx1xxxxxxxxx	Card reading from DRAM IOMMU - Type 1
MEM_READ.PART2	bxxxxx1xx	b111 bxxxxxxxxx1xx	Card reading from DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2

Table 2-180. Unit Masks for DATA_REQ_OF_CPU (Sheet 2 of 3)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_READ.PART3	bxxxxx1xx	b111 bxxxxxxxx1xxx	Card reading from DRAM x4 card is plugged in to slot 3
MEM_READ.PART7	bxxxxx1xx	b111 bxxxx1xxxxxx	Card reading from DRAM x4 card is plugged in to slot 7
MEM_READ.PART6	bxxxxx1xx	b111 bxxxxx1xxxxx	Card reading from DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MEM_READ.PART5	bxxxxx1xx	b111 bxxxxxx1xxxx	Card reading from DRAM x4 card is plugged in to slot 5
MEM_READ.PART4	bxxxxx1xx	b111 bxxxxxxx1xxx	Card reading from DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MEM_READ.PART0	bxxxxx1xx	b111 bxxxxxxxxxx1	Card reading from DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_READ.PART1	bxxxx1xxx	b111 bxxxxxxxxx1x	Card reading from another Card (same or different stack) x4 card is plugged in to slot 1
PEER_READ.IOMMU1	bxxxx1xxx	b111 bxx1xxxxxxxx	Card reading from another Card (same or different stack) IOMMU - Type 1
PEER_READ.PART3	bxxxx1xxx	b111 bxxxxxxxx1xxx	Card reading from another Card (same or different stack) x4 card is plugged in to slot 3
PEER_READ.PART0	bxxxx1xxx	b111 bxxxxxxxxxx1	Card reading from another Card (same or different stack) x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_READ.PART5	bxxxx1xxx	b111 bxxxxxx1xxxx	Card reading from another Card (same or different stack) x4 card is plugged in to slot 5
PEER_READ.IOMMU0	bxxxx1xxx	b111 bxxx1xxxxxxx	Card reading from another Card (same or different stack) IOMMU - Type 0
PEER_READ.PART2	bxxxx1xxx	b111 bxxxxxxxx1xx	Card reading from another Card (same or different stack) x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_READ.PART4	bxxxx1xxx	b111 bxxxxxxx1xxx	Card reading from another Card (same or different stack) x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_READ.PART7	bxxxx1xxx	b111 bxxxx1xxxxxx	Card reading from another Card (same or different stack) x4 card is plugged in to slot 7
PEER_READ.PART6	bxxxx1xxx	b111 bxxxxx1xxxxx	Card reading from another Card (same or different stack) x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
ATOMIC.IOMMU1	bxxx1xxxx	b111 bxx1xxxxxxxx	Atomic requests targeting DRAM IOMMU - Type 1
ATOMIC.PART4	bxxx1xxxx	b111 bxxxxxxx1xxx	Atomic requests targeting DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
ATOMIC.PART3	bxxx1xxxx	b111 bxxxxxxxx1xxx	Atomic requests targeting DRAM x4 card is plugged in to slot 3
ATOMIC.PART5	bxxx1xxxx	b111 bxxxxxx1xxxx	Atomic requests targeting DRAM x4 card is plugged in to slot 5
ATOMIC.IOMMU0	bxxx1xxxx	b111 bxxx1xxxxxxx	Atomic requests targeting DRAM IOMMU - Type 0
ATOMIC.PART2	bxxx1xxxx	b111 bxxxxxxxxx1xx	Atomic requests targeting DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
ATOMIC.PART7	bxxx1xxxx	b111 bxxxx1xxxxxx	Atomic requests targeting DRAM x4 card is plugged in to slot 7
ATOMIC.PART0	bxxx1xxxx	b111 bxxxxxxxxxx1	Atomic requests targeting DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0

Table 2-180. Unit Masks for DATA_REQ_OF_CPU (Sheet 3 of 3)

Extension	umask [15:8]	xtra [50:36]	Description
ATOMIC.PART1	bxxx1xxxx	b111 bxxxxxxxxxx1x	Atomic requests targeting DRAM x4 card is plugged in to slot 1
ATOMIC.PART6	bxxx1xxxx	b111 bxxxxx1xxxxxx	Atomic requests targeting DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MSG.PART1	bx1xxxxxx	b111 bxxxxxxxxxx1x	Messages x4 card is plugged in to slot 1
MSG.PART5	bx1xxxxxx	b111 bxxxxxx1xxxxx	Messages x4 card is plugged in to slot 5
MSG.PART6	bx1xxxxxx	b111 bxxxxx1xxxxxx	Messages x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MSG.PART0	bx1xxxxxx	b111 bxxxxxxxxxxx1	Messages x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MSG.IOMMU0	bx1xxxxxx	b111 bxxx1xxxxxxxx	Messages IOMMU - Type 0
MSG.PART4	bx1xxxxxx	b111 bxxxxxxx1xxxx	Messages x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MSG.PART2	bx1xxxxxx	b111 bxxxxxxxxx1xx	Messages x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MSG.PART7	bx1xxxxxx	b111 bxxxx1xxxxxxx	Messages x4 card is plugged in to slot 7
MSG.IOMMU1	bx1xxxxxx	b111 bxx1xxxxxxxxx	Messages IOMMU - Type 1
MSG.PART3	bx1xxxxxx	b111 bxxxxxxxx1xxx	Messages x4 card is plugged in to slot 3
CMPD.PART3	b1xxxxxxx	b111 bxxxxxxxx1xxx	CmpD - device sending completion to CPU request x4 card is plugged in to slot 3
CMPD.PART6	b1xxxxxxx	b111 bxxxxx1xxxxxx	CmpD - device sending completion to CPU request x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
CMPD.PART0	b1xxxxxxx	b111 bxxxxxxxxxxx1	CmpD - device sending completion to CPU request x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CMPD.PART7	b1xxxxxxx	b111 bxxxx1xxxxxxx	CmpD - device sending completion to CPU request x4 card is plugged in to slot 7
CMPD.PART4	b1xxxxxxx	b111 bxxxxxxx1xxxx	CmpD - device sending completion to CPU request x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CMPD.IOMMU0	b1xxxxxxx	b111 bxxx1xxxxxxxx	CmpD - device sending completion to CPU request IOMMU - Type 0
CMPD.PART2	b1xxxxxxx	b111 bxxxxxxxxx1xx	CmpD - device sending completion to CPU request x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CMPD.IOMMU1	b1xxxxxxx	b111 bxx1xxxxxxxxx	CmpD - device sending completion to CPU request IOMMU - Type 1
CMPD.PART1	b1xxxxxxx	b111 bxxxxxxxxxx1x	CmpD - device sending completion to CPU request x4 card is plugged in to slot 1
CMPD.PART5	b1xxxxxxx	b111 bxxxxxx1xxxxx	CmpD - device sending completion to CPU request x4 card is plugged in to slot 5

INBOUND_ARB_REQ

- **Title:** Incoming arbitration requests
- **Category:** ITC Events
- **Event Code:** 0x86
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** How often different queues (for example, channel / fc) ask to send request into pipeline

Table 2-181. Unit Masks for INBOUND_ARB_REQ

Extension	umask [15:8]	xtra [50:36]	Description
IOMMU_REQ	bxxxxxxx1	b111 bxxxx1111 1111	Issuing to IOMMU
IOMMU_HIT	bxxxxxx1x	b111 bxxxx1111 1111	Processing response from IOMMU
REQ_OWN	bxxxxx1xx	b111 bxxxx1111 1111	Request Ownership Only for posted requests
FINAL_RD_WR	bxxxx1xxx	b111 bxxxx1111 1111	Issuing final read or write of line
WR	bxxx1xxxx	b111 bxxxx1111 1111	Writing line Only for posted requests
DATA	bxx1xxxxx	b111 bxxxx1111 1111	Passing data to be written Only for posted requests

INBOUND_ARB_WON

- **Title:** Incoming arbitration requests granted
- **Category:** ITC Events
- **Event Code:** 0x87
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** How often different queues (for example, channel / fc) are allowed to send request into pipeline

Table 2-182. Unit Masks for INBOUND_ARB_WON

Extension	umask [15:8]	xtra [50:36]	Description
IOMMU_REQ	bxxxxxxx1	b111 bxxxx1111 1111	Issuing to IOMMU
IOMMU_HIT	bxxxxxx1x	b111 bxxxx1111 1111	Processing response from IOMMU
REQ_OWN	bxxxxx1xx	b111 bxxxx1111 1111	Request Ownership Only for posted requests
FINAL_RD_WR	bxxxx1xxx	b111 bxxxx1111 1111	Issuing final read or write of line

Table 2-182. Unit Masks for INBOUND_ARB_WON

Extension	umask [15:8]	xtra [50:36]	Description
WR	bxxx1xxxx	b111 bxxxx1111 1111	Writing line Only for posted requests
DATA	bxx1xxxxx	b111 bxxxx1111 1111	Passing data to be written Only for posted requests

IOMMU0

- **Title:**
- **Category:** IOMMU Events
- **Event Code:** 0x40
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-183. Unit Masks for IOMMU0

Extension	umask [15:8]	Description
FIRST_LOOKUPS	bxxxxxxx1	IOTLB lookups first Some transactions have to look up IOTLB multiple times. Counts the first time a request looks up IOTLB.
ALL_LOOKUPS	bxxxxxx1x	IOTLB lookups all Some transactions have to look up IOTLB multiple times. Counts every time a request looks up IOTLB.
4K_HITS	bxxxxx1xx	IOTLB Hits to a 4K Page Counts if a transaction to a 4K page, on its first lookup, hits the IOTLB.
2M_HITS	bxxxx1xxx	IOTLB Hits to a 2M Page Counts if a transaction to a 2M page, on its first lookup, hits the IOTLB.
1G_HITS	bxxx1xxxx	IOTLB Hits to a 1G Page Counts if a transaction to a 1G page, on its first lookup, hits the IOTLB.
MISSES	bxx1xxxxx	IOTLB Fills (same as IOTLB miss) When a transaction misses IOTLB, it does a page walk to look up memory and bring in the relevant page translation. Counts when this page translation is written to IOTLB.
CTXT_CACHE_LOOKUPS	bx1xxxxxx	Context cache lookups Counts each time a transaction looks up root context cache.
CTXT_CACHE_HITS	b1xxxxxxx	Context cache hits Counts each time a first look up of the transaction hits the RCC.

IOMMU1

- **Title:**
- **Category:** IOMMU Events
- **Event Code:** 0x41
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-184. Unit Masks for IOMMU1

Extension	umask [15:8]	Description
PWT_CACHE_LOOKUPS	bxxxxxxx1	PageWalk cache lookup Counts each time a transaction looks up second level page walk cache.
PWC_4K_HITS	bxxxxxx1x	PWC Hit to a 4K page Counts each time a transaction's first look up hits the SLPWC at the 4K level
PWC_2M_HITS	bxxxxx1xx	PWC Hit to a 2M page Counts each time a transaction's first look up hits the SLPWC at the 2M level
PWC_1G_HITS	bxxxx1xxx	PWC Hit to a 1G page Counts each time a transaction's first look up hits the SLPWC at the 1G level
PWC_512G_HITS	bxxx1xxxx	PWT Hit to a 256T page Counts each time a transaction's first look up hits the SLPWC at the 512G level
PWC_CACHE_FILLS	bxx1xxxxx	PageWalk cache fill When a transaction misses SLPWC, it does a page walk to look up memory and bring in the relevant page translation. When this page translation is written to SLPWC, ObsPwcFillValid_nnnH is asserted.
NUM_MEM_ACCESSES	bx1xxxxxx	IOMMU memory access IOMMU sends out memory fetches when it misses the cache look up which is indicated by this signal. M2IOSF only uses low priority channel
CYC_PWT_FULL	b1xxxxxxx	Cycles PWT full Counts cycles the IOMMU has reached its maximum limit for outstanding page walks.

IOMMU3

- **Title:**
- **Category:** IOMMU Events
- **Event Code:** 0x43
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-185. Unit Masks for IOMMU3

Extension	umask [15:8]	Description
NUM_INVAL_GBL	bxxxxxxx1	Global IOTLB invalidation cycles Indicates that IOMMU is doing global invalidation.
NUM_INVAL_DOMAIN	bxxxxxx1x	Domain-selective IOTLB invalidation cycles Counts number of Domain selective invalidation events
NUM_INVAL_PAGE	bxxxxx1xx	Page-selective IOTLB invalidation cycles Counts number of Page-selective within Domain Invalidation events
NUM_CTXT_CACHE_INVAL_GBL	bxxxx1xxx	Context cache global invalidation cycles Counts number of Context Cache global invalidation events
NUM_CTXT_CACHE_INVAL_DOMAIN	bxxx1xxxx	Domain-selective Context cache invalidation cycles Counts number of Domain selective context cache invalidation events
NUM_CTXT_CACHE_INVAL_DEVICE	bxx1xxxxx	Device-selective Context cache invalidation cycles Counts number of Device selective context cache invalidation events

Table 2-185. Unit Masks for IOMMU3

Extension	umask [15:8]	Description
INT_CACHE_LOOKUPS	bx1xxxxxx	Interrupt Entry cache lookup Counts the number of transaction looks up that interrupt remapping cache.
INT_CACHE_HITS	b1xxxxxxx	Interrupt Entry cache hit Counts each time a transaction's first look up hits the IEC.

MASK_MATCH_AND

- **Title:** AND Mask/match for debug bus
- **Category:** Debug Events
- **Event Code:** 0x02
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-1

- **Definition:** Asserted if all bits specified by mask match

Table 2-186. Unit Masks for MASK_MATCH_AND

Extension	umask [15:8]	Description
BUS0	bxxxxxxx1	Non-PCIe bus
BUS1	bxxxxxx1x	PCIe bus
BUS0_NOT_BUS1	bxxxxx1xx	Non-PCIe bus and !(PCIe bus)
BUS0_BUS1	bxxxx1xxx	Non-PCIe bus and PCIe bus
NOT_BUS0_BUS1	bxxx1xxxx	!(Non-PCIe bus) and PCIe bus
NOT_BUS0_NOT_BUS1	bxx1xxxxx	!(Non-PCIe bus) and !(PCIe bus)

MASK_MATCH_OR

- **Title:** OR Mask/match for debug bus
- **Category:** Debug Events
- **Event Code:** 0x03
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-1

- **Definition:** Asserted if any bits specified by mask match

Table 2-187. Unit Masks for MASK_MATCH_OR

Extension	umask [15:8]	Description
BUS0	bxxxxxxx1	Non-PCIe bus
BUS1	bxxxxxx1x	PCIe bus
BUS0_NOT_BUS1	bxxxxx1xx	Non-PCIe bus and !(PCIe bus)
BUS0_BUS1	bxxxx1xxx	Non-PCIe bus and PCIe bus
NOT_BUS0_BUS1	bxxx1xxxx	!(Non-PCIe bus) and PCIe bus
NOT_BUS0_NOT_BUS1	bxx1xxxxx	!(Non-PCIe bus) and !(PCIe bus)

NOTHING

- **Title:** Counting disabled
- **Category:** CLOCK Events
- **Event Code:** 0x80
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

NUM_OUSTANDING_REQ_FROM_CPU

- **Title:** Occupancy of outbound request queue
- **Category:** OTC Events
- **Event Code:** 0xc5
- **Max. Inc/Cyc:.** 256

Register Restrictions: 2-3

- **Definition:** Counts number of outbound requests/completions IIO is currently processing
- **NOTE:** Only 1 bit each of the fc_mask and ch_mask can be set

Table 2-188. Unit Masks for NUM_OUSTANDING_REQ_FROM_CPU

Extension	umask [15:8]	xtra [50:36]	Description
TO_IO	bxxxx1xxx	b111 bxxxx1111 1111	To device

NUM_OUTSTANDING_REQ_OF_CPU

- **Title:**
- **Category:** ITC Events
- **Event Code:** 0x88
- **Max. Inc/Cyc:.** 512

Register Restrictions: 2-3

- **Definition:**

Table 2-189. Unit Masks for NUM_OUTSTANDING_REQ_OF_CPU

Extension	umask [15:8]	xtra [50:36]	Description
IOMMU_REQ	bxxxxxxx1	b111 bxxxx1111 1111	Issuing to IOMMU
IOMMU_HIT	bxxxxxx1x	b111 bxxxx1111 1111	Processing response from IOMMU
REQ_OWN	bxxxxx1xx	b111 bxxxx1111 1111	Request Ownership Only for posted requests
FINAL_RD_WR	bxxxx1xxx	b111 bxxxx1111 1111	Issuing final read or write of line

Table 2-189. Unit Masks for NUM_OUTSTANDING_REQ_OF_CPU

Extension	umask [15:8]	xtra [50:36]	Description
WR	bxxx1xxxx	b111 bxxxx1111 1111	Writing line Only for posted requests
DATA	bxx1xxxxx	b111 bxxxx1111 1111	Passing data to be written Only for posted requests

NUM_REQ_FROM_CPU

- **Title:** Number requests sent to PCIe from main die
- **Category:** OTC Events
- **Event Code:** 0xc2
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-190. Unit Masks for NUM_REQ_FROM_CPU

Extension	umask [15:8]	xtra [50:36]	Description
IRP	bxxxxxxx1	b111 bxxxx1111 1111	From IRP Captures Posted/Non-posted allocations from IRP. i.e. either non-confined P2P traffic or from the CPU
ITC	bxxxxxx1x	b111 bxxxx1111 1111	From ITC Confined P2P
PREALLOC	bxxxxx1xx	b111 bxxxx1111 1111	Completion allocations

NUM_REQ_OF_CPU

- **Title:** Number requests PCIe makes of the main die
- **Category:** ITC Events
- **Event Code:** 0x85
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts full PCIe requests before they're broken into a series of cache-line size requests as measured by DATA_REQ_OF_CPU and TXN_REQ_OF_CPU.

Table 2-191. Unit Masks for NUM_REQ_OF_CPU

Extension	umask [15:8]	xtra [50:36]	Description
COMMIT.ALL	bxxxxxxx1	b111 bxxxx1111 1111	All
ALL.DROP	bxxxxxx1x	b111 bxxxx1111 1111	Drop request Packet error detected, must be dropped

NUM_REQ_OF_CPU_BY_TGT

- **Title:** Num requests sent by PCIe - by target
- **Category:** ITC Events
- **Event Code:** 0x8e
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-192. Unit Masks for NUM_REQ_OF_CPU_BY_TGT

Extension	umask [15:8]	xtra [50:36]	Description
MSGB	bxxxxxxx1	b111 bxxxx1111 1111	MsgB
MCAST	bxxxxxx1x	b111 bxxxx1111 1111	Multi-cast
UBOX	bxxxxx1xx	b111 bxxxx1111 1111	Ubox
MEM	bxxxx1xxx	b111 bxxxx1111 1111	Memory
REM_P2P	bxxx1xxxx	b111 bxxxx1111 1111	Remote P2P
LOC_P2P	bxx1xxxxx	b111 bxxxx1111 1111	Local P2P
CONFINED_P2P	bx1xxxxxx	b111 bxxxx1111 1111	Confined P2P
ABORT	b1xxxxxxx	b111 bxxxx1111 1111	Abort

NUM_TGT_MATCHED_REQ_OF_CPU

- **Title:** ITC address map 1
- **Category:** ITC Events
- **Event Code:** 0x8f
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

OUTBOUND_CL_REQS_ISSUED

- **Title:** Outbound cacheline requests issued
- **Category:** OTC Events
- **Event Code:** 0xd0
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Each outbound cacheline granular request may need to make multiple passes through the pipeline. Each time a cacheline completes all its passes it advances line

Table 2-193. Unit Masks for OUTBOUND_CL_REQS_ISSUED

Extension	umask [15:8]	xtra [50:36]	Description
TO_IO	bxxxx1xxx	b111 bxxxx1111 1111	64B requests issued to device

OUTBOUND_TLP_REQS_ISSUED

- **Title:** Outbound TLP (transaction layer packet) requests issued
- **Category:** OTC Events
- **Event Code:** 0xd1
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Each time an outbound completes all its passes it advances the pointer

Table 2-194. Unit Masks for OUTBOUND_TLP_REQS_ISSUED

Extension	umask [15:8]	xtra [50:36]	Description
TO_IO	bxxxx1xxx	b111 bxxxx1111 1111	To device

PWT_OCCUPANCY

- **Title:** PWT occupancy
- **Category:** IOMMU Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Indicates how many page walks are outstanding at any point in time.

REQ_FROM_PCIE_CL_CMPL

- **Title:** PCIe Request - cacheline complete
- **Category:** ITC Events
- **Event Code:** 0x91
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Each PCIe request is broken down into a series of cacheline granular requests and each cacheline size request may need to make multiple passes through the pipeline (for example, for posted interrupts or multi-cast). Each time a cacheline completes all its passes (for example, finishes posting writes to all multi-cast targets) it advances line
- **NOTE:** For a normal write (no posted interrupt, no multi-cast) advance line = advance state. For a PCIe request of <= cacheline, advance pointer = advance line

Table 2-195. Unit Masks for REQ_FROM_PCIE_CL_CMPL

Extension	umask [15:8]	xtra [50:36]	Description
REQ_OWN	bxxxxx1xx	b111 bxxxx1111 1111	Request Ownership Only for posted requests
FINAL_RD_WR	bxxxx1xxx	b111 bxxxx1111 1111	Issuing final read or write of line
WR	bxxx1xxxx	b111 bxxxx1111 1111	Writing line Only for posted requests
DATA	bxx1xxxxx	b111 bxxxx1111 1111	Passing data to be written Only for posted requests

REQ_FROM_PCIE_CMPL

- **Title:** PCIe Request complete
- **Category:** ITC Events
- **Event Code:** 0x92
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Each PCIe request is broken down into a series of cacheline granular requests and each cacheline size request may need to make multiple passes through the pipeline (for example, for posted interrupts or multi-cast). Each time a single PCIe request completes all its cacheline granular requests, it advances pointer.
- **NOTE:** For a PCIe request of <= cacheline, advance pointer = advance line

Table 2-196. Unit Masks for REQ_FROM_PCIE_CMPL

Extension	umask [15:8]	xtra [50:36]	Description
IOMMU_REQ	bxxxxxxx1	b111 bxxxx1111 1111	Issuing to IOMMU
IOMMU_HIT	bxxxxxx1x	b111 bxxxx1111 1111	Processing response from IOMMU
REQ_OWN	bxxxxx1xx	b111 bxxxx1111 1111	Request Ownership Only for posted requests
FINAL_RD_WR	bxxxx1xxx	b111 bxxxx1111 1111	Issuing final read or write of line
WR	bxxx1xxxx	b111 bxxxx1111 1111	Writing line Only for posted requests
DATA	bxx1xxxxx	b111 bxxxx1111 1111	Passing data to be written Only for posted requests

REQ_FROM_PCIE_PASS_CMPL

- **Title:** PCIe Request - pass complete
- **Category:** ITC Events
- **Event Code:** 0x90
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Each PCIe request is broken down into a series of cacheline granular requests and each cacheline size request may need to make multiple passes through the pipeline (for example, for posted interrupts or multi-cast). Each time a cacheline completes a single pass (for example, posts a write to single multi-cast target) it advances state
- **NOTE:** For a normal write (no posted interrupt, no multi-cast) advance line = advance state

Table 2-197. Unit Masks for REQ_FROM_PCIE_PASS_CMPL

Extension	umask [15:8]	xtra [50:36]	Description
REQ_OWN	bxxxxx1xx	b111 bxxxx1111 1111	Request Ownership Only for posted requests
FINAL_RD_WR	bxxxx1xxx	b111 bxxxx1111 1111	Issuing final read or write of line
WR	bxxx1xxxx	b111 bxxxx1111 1111	Writing line Only for posted requests
DATA	bxx1xxxxx	b111 bxxxx1111 1111	Passing data to be written Only for posted requests

SYMBOL_TIMES

- **Title:** Symbol Times on Link
- **Category:** Miscellaneous Events
- **Event Code:** 0x82
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Gen1 - increment once every 4nS, Gen2 - increment once every 2nS, Gen3 - increment once every 1nS

TXN_REQ_BY_CPU

- **Title:** Number Transactions requested by the CPU
- **Category:** Transaction Events
- **Event Code:** 0xc1
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Also known as Outbound. Number of requests initiated by the main die, including reads and writes.
- **NOTE:** Unlike free running counters, Mem Read and Peer read subevents count requests not completions. Peer R/W subevents do not include confined P2P traffic. Peer R/W subevents are also counted as Mem R/W subevents.

Table 2-198. Unit Masks for TXN_REQ_BY_CPU (Sheet 1 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_WRITE.IOMMU1	bxxxxxxx1	b111 bxx1xxxxxxxxx	Core writing to Card's MMIO space IOMMU - Type 1
MEM_WRITE.PART3	bxxxxxxx1	b111 bxxxxxxxx1xxx	Core writing to Card's MMIO space x4 card is plugged in to slot 3
MEM_WRITE.PART0	bxxxxxxx1	b111 bxxxxxxxxxxx1	Core writing to Card's MMIO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_WRITE.PART5	bxxxxxxx1	b111 bxxxxxx1xxxxx	Core writing to Card's MMIO space x4 card is plugged in to slot 5
MEM_WRITE.PART2	bxxxxxxx1	b111 bxxxxxxxxx1xx	Core writing to Card's MMIO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MEM_WRITE.PART4	bxxxxxxx1	b111 bxxxxxxx1xxx	Core writing to Card's MMIO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MEM_WRITE.PART1	bxxxxxxx1	b111 bxxxxxxxxx1x	Core writing to Card's MMIO space x4 card is plugged in to slot 1
MEM_WRITE.IOMMU0	bxxxxxxx1	b111 bxxx1xxxxxxxx	Core writing to Card's MMIO space IOMMU - Type 0
MEM_WRITE.PART7	bxxxxxxx1	b111 bxxxx1xxxxxxx	Core writing to Card's MMIO space x4 card is plugged in to slot 7
MEM_WRITE.PART6	bxxxxxxx1	b111 bxxxxx1xxxxxx	Core writing to Card's MMIO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_WRITE.PART5	bxxxxxx1x	b111 bxxxxxx1xxxxx	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 5
PEER_WRITE.PART6	bxxxxxx1x	b111 bxxxxx1xxxxxx	Another card (different IIO stack) writing to this card. x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_WRITE.PART7	bxxxxxx1x	b111 bxxxx1xxxxxxx	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 7
PEER_WRITE.PART2	bxxxxxx1x	b111 bxxxxxxxxx1xx	Another card (different IIO stack) writing to this card. x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_WRITE.PART3	bxxxxxx1x	b111 bxxxxxxxx1xxx	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 3
PEER_WRITE.PART4	bxxxxxx1x	b111 bxxxxxxx1xxx	Another card (different IIO stack) writing to this card. x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_WRITE.PART0	bxxxxxx1x	b111 bxxxxxxxxxx1	Another card (different IIO stack) writing to this card. x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0

Table 2-198. Unit Masks for TXN_REQ_BY_CPU (Sheet 2 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
PEER_WRITE.PART1	bxxxxxx1x	b111 bxxxxxxxxx x1x	Another card (different IIO stack) writing to this card. x4 card is plugged in to slot 1
PEER_WRITE.IOMMU0	bxxxxxx1x	b111 bxx1xxxxxx xxx	Another card (different IIO stack) writing to this card. IOMMU - Type 1
MEM_READ.PART2	bxxxxx1xx	b111 bxxxxxxxxx 1xx	Core reading from Card's MMIO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MEM_READ.PART7	bxxxxx1xx	b111 bxxxx1xxxx xxx	Core reading from Card's MMIO space x4 card is plugged in to slot 7
MEM_READ.PART1	bxxxxx1xx	b111 bxxxxxxxxx x1x	Core reading from Card's MMIO space x4 card is plugged in to slot 1
MEM_READ.IOMMU1	bxxxxx1xx	b111 bxx1xxxxxx xxx	Core reading from Card's MMIO space IOMMU - Type 1
MEM_READ.PART4	bxxxxx1xx	b111 bxxxxxxx1x xxx	Core reading from Card's MMIO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MEM_READ.PART3	bxxxxx1xx	b111 bxxxxxxxx1 xxx	Core reading from Card's MMIO space x4 card is plugged in to slot 3
MEM_READ.IOMMU0	bxxxxx1xx	b111 bxxx1xxxxx xxx	Core reading from Card's MMIO space IOMMU - Type 0
MEM_READ.PART0	bxxxxx1xx	b111 bxxxxxxxxx xx1	Core reading from Card's MMIO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_READ.PART6	bxxxxx1xx	b111 bxxxxx1xxx xxx	Core reading from Card's MMIO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MEM_READ.PART5	bxxxxx1xx	b111 bxxxxxx1xx xxx	Core reading from Card's MMIO space x4 card is plugged in to slot 5
PEER_READ.IOMMU0	bxxxx1xxx	b111 bxxx1xxxxx xxx	Another card (different IIO stack) reading from this card. IOMMU - Type 0
PEER_READ.PART1	bxxxx1xxx	b111 bxxxxxxxxx x1x	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 1
PEER_READ.PART6	bxxxx1xxx	b111 bxxxxx1xxx xxx	Another card (different IIO stack) reading from this card. x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_READ.PART7	bxxxx1xxx	b111 bxxxx1xxxx xxx	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 7
PEER_READ.PART2	bxxxx1xxx	b111 bxxxxxxxxx 1xx	Another card (different IIO stack) reading from this card. x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_READ.PART5	bxxxx1xxx	b111 bxxxxxx1xx xxx	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 5

Table 2-198. Unit Masks for TXN_REQ_BY_CPU (Sheet 3 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
PEER_READ.IOMMU1	bxxxx1xxx	b111 bxx1xxxxxx xxx	Another card (different IIO stack) reading from this card. IOMMU - Type 1
PEER_READ.PART3	bxxxx1xxx	b111 bxxxxxxxx1 xxx	Another card (different IIO stack) reading from this card. x4 card is plugged in to slot 3
PEER_READ.PART0	bxxxx1xxx	b111 bxxxxxxxxx xx1	Another card (different IIO stack) reading from this card. x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_READ.PART4	bxxxx1xxx	b111 bxxxxxxx1x xxx	Another card (different IIO stack) reading from this card. x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CFG_WRITE.IOMMU1	bxxx1xxxx	b111 bxx1xxxxxx xxx	Core writing to Card's PCICFG space IOMMU - Type 1
CFG_WRITE.PART3	bxxx1xxxx	b111 bxxxxxxxx1 xxx	Core writing to Card's PCICFG space x4 card is plugged in to slot 3
CFG_WRITE.PART0	bxxx1xxxx	b111 bxxxxxxxxx xx1	Core writing to Card's PCICFG space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CFG_WRITE.PART2	bxxx1xxxx	b111 bxxxxxxxxx 1xx	Core writing to Card's PCICFG space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CFG_WRITE.PART5	bxxx1xxxx	b111 bxxxxxx1xx xxx	Core writing to Card's PCICFG space x4 card is plugged in to slot 5
CFG_WRITE.PART4	bxxx1xxxx	b111 bxxxxxxx1x xxx	Core writing to Card's PCICFG space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CFG_WRITE.PART1	bxxx1xxxx	b111 bxxxxxxxxx x1x	Core writing to Card's PCICFG space x4 card is plugged in to slot 1
CFG_WRITE.IOMMU0	bxxx1xxxx	b111 bxxx1xxxxx xxx	Core writing to Card's PCICFG space IOMMU - Type 0
CFG_WRITE.PART7	bxxx1xxxx	b111 bxxxx1xxxx xxx	Core writing to Card's PCICFG space x4 card is plugged in to slot 7
CFG_WRITE.PART6	bxxx1xxxx	b111 bxxxxx1xxx xxx	Core writing to Card's PCICFG space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
IO_WRITE.PART1	bxx1xxxxx	b111 bxxxxxxxxx x1x	Core writing to Card's IO space x4 card is plugged in to slot 1
IO_WRITE.IOMMU1	bxx1xxxxx	b111 bxx1xxxxxx xxx	Core writing to Card's IO space IOMMU - Type 1
IO_WRITE.PART2	bxx1xxxxx	b111 bxxxxxxxxx 1xx	Core writing to Card's IO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2

Table 2-198. Unit Masks for TXN_REQ_BY_CPU (Sheet 4 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
IO_WRITE.PART7	bxx1xxxxx	b111 bxxxx1xxxx xxx	Core writing to Card's IO space x4 card is plugged in to slot 7
IO_WRITE.PART4	bxx1xxxxx	b111 bxxxxxxx1x xxx	Core writing to Card's IO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
IO_WRITE.IOMMU0	bxx1xxxxx	b111 bxxx1xxxxx xxx	Core writing to Card's IO space IOMMU - Type 0
IO_WRITE.PART3	bxx1xxxxx	b111 bxxxxxxxx1 xxx	Core writing to Card's IO space x4 card is plugged in to slot 3
IO_WRITE.PART6	bxx1xxxxx	b111 bxxxxx1xxx xxx	Core writing to Card's IO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
IO_WRITE.PART5	bxx1xxxxx	b111 bxxxxxx1xx xxx	Core writing to Card's IO space x4 card is plugged in to slot 5
IO_WRITE.PART0	bxx1xxxxx	b111 bxxxxxxxxx xx1	Core writing to Card's IO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CFG_READ.PART2	bx1xxxxxx	b111 bxxxxxxxxx 1xx	Core reading from Card's PCICFG space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
CFG_READ.PART7	bx1xxxxxx	b111 bxxxx1xxxx xxx	Core reading from Card's PCICFG space x4 card is plugged in to slot 7
CFG_READ.PART1	bx1xxxxxx	b111 bxxxxxxxxx x1x	Core reading from Card's PCICFG space x4 card is plugged in to slot 1
CFG_READ.IOMMU1	bx1xxxxxx	b111 bxx1xxxxxx xxx	Core reading from Card's PCICFG space IOMMU - Type 1
CFG_READ.PART4	bx1xxxxxx	b111 bxxxxxxx1x xxx	Core reading from Card's PCICFG space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CFG_READ.PART3	bx1xxxxxx	b111 bxxxxxxxx1 xxx	Core reading from Card's PCICFG space x4 card is plugged in to slot 3
CFG_READ.IOMMU0	bx1xxxxxx	b111 bxxx1xxxxx xxx	Core reading from Card's PCICFG space IOMMU - Type 0
CFG_READ.PART0	bx1xxxxxx	b111 bxxxxxxxxx xx1	Core reading from Card's PCICFG space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CFG_READ.PART6	bx1xxxxxx	b111 bxxxxx1xxx xxx	Core reading from Card's PCICFG space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
CFG_READ.PART5	bx1xxxxxx	b111 bxxxxxx1xx xxx	Core reading from Card's PCICFG space x4 card is plugged in to slot 5
IO_READ.IOMMU1	b1xxxxxxx	b111 bxx1xxxxxx xxx	Core reading from Card's IO space IOMMU - Type 1

Table 2-198. Unit Masks for TXN_REQ_BY_CPU (Sheet 5 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
IO_READ.PART5	b1xxxxxxx	b111 bxxxxxx1xx xxx	Core reading from Card's IO space x4 card is plugged in to slot 5
IO_READ.PART1	b1xxxxxxx	b111 bxxxxxxxxx x1x	Core reading from Card's IO space x4 card is plugged in to slot 1
IO_READ.PART3	b1xxxxxxx	b111 bxxxxxxxx1 xxx	Core reading from Card's IO space x4 card is plugged in to slot 3
IO_READ.PART0	b1xxxxxxx	b111 bxxxxxxxxx xx1	Core reading from Card's IO space x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
IO_READ.PART7	b1xxxxxxx	b111 bxxxx1xxxx xxx	Core reading from Card's IO space x4 card is plugged in to slot 7
IO_READ.PART4	b1xxxxxxx	b111 bxxxxxxx1x xxx	Core reading from Card's IO space x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
IO_READ.PART2	b1xxxxxxx	b111 bxxxxxxxxx 1xx	Core reading from Card's IO space x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
IO_READ.IOMMU0	b1xxxxxxx	b111 bxxx1xxxxx xxx	Core reading from Card's IO space IOMMU - Type 0
IO_READ.PART6	b1xxxxxxx	b111 bxxxxx1xxx xxx	Core reading from Card's IO space x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6

TXN_REQ_OF_CPU

- **Title:** Number Transactions requested of the CPU
- **Category:** Transaction Events
- **Event Code:** 0x84
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Also known as Inbound. Number of 64B cache line requests initiated by the Card, including reads and writes.
- **NOTE:** Unlike free running counters, Mem Read and Peer read subevents count requests not completions. Unlike the *_BY_CPU.PEER* events, peer R/W subevents do include confined P2P traffic.

Table 2-199. Unit Masks for TXN_REQ_OF_CPU (Sheet 1 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_WRITE.PART1	bxxxxxxx1	b111 bxxxxxxxxx x1x	Card writing to DRAM x4 card is plugged in to slot 1
MEM_WRITE.PART0	bxxxxxxx1	b111 bxxxxxxxxx xx1	Card writing to DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_WRITE.IOMMU1	bxxxxxxx1	b111 bxx1xxxxxx xxx	Card writing to DRAM IOMMU - Type 1

Table 2-199. Unit Masks for TXN_REQ_OF_CPU (Sheet 2 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_WRITE.PART3	bxxxxxxx1	b111 bxxxxxxxx1 xxx	Card writing to DRAM x4 card is plugged in to slot 3
MEM_WRITE.PART2	bxxxxxxx1	b111 bxxxxxxxxx 1xx	Card writing to DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MEM_WRITE.PART5	bxxxxxxx1	b111 bxxxxxx1xx xxx	Card writing to DRAM x4 card is plugged in to slot 5
MEM_WRITE.IOMMU0	bxxxxxxx1	b111 bxxx1xxxxx xxx	Card writing to DRAM IOMMU - Type 0
MEM_WRITE.PART4	bxxxxxxx1	b111 bxxxxxxx1x xxx	Card writing to DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MEM_WRITE.PART7	bxxxxxxx1	b111 bxxxx1xxxx xxx	Card writing to DRAM x4 card is plugged in to slot 7
MEM_WRITE.PART6	bxxxxxxx1	b111 bxxxxx1xxx xxx	Card writing to DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_WRITE.PART5	bxxxxxx1x	b111 bxxxxxx1xx xxx	Card writing to another Card (same or different stack) x4 card is plugged in to slot 5
PEER_WRITE.PART6	bxxxxxx1x	b111 bxxxxx1xxx xxx	Card writing to another Card (same or different stack) x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_WRITE.PART7	bxxxxxx1x	b111 bxxxx1xxxx xxx	Card writing to another Card (same or different stack) x4 card is plugged in to slot 7
PEER_WRITE.IOMMU1	bxxxxxx1x	b111 bxx1xxxxxx xxx	Card writing to another Card (same or different stack) IOMMU - Type 1
PEER_WRITE.PART2	bxxxxxx1x	b111 bxxxxxxxxx 1xx	Card writing to another Card (same or different stack) x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_WRITE.PART0	bxxxxxx1x	b111 bxxxxxxxxx xx1	Card writing to another Card (same or different stack) x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_WRITE.PART4	bxxxxxx1x	b111 bxxxxxxx1x xxx	Card writing to another Card (same or different stack) x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_WRITE.PART3	bxxxxxx1x	b111 bxxxxxxxx1 xxx	Card writing to another Card (same or different stack) x4 card is plugged in to slot 3
PEER_WRITE.IOMMU0	bxxxxxx1x	b111 bxxx1xxxxx xxx	Card writing to another Card (same or different stack) IOMMU - Type 0
PEER_WRITE.PART1	bxxxxxx1x	b111 bxxxxxxxxx x1x	Card writing to another Card (same or different stack) x4 card is plugged in to slot 1
MEM_READ.PART2	bxxxxx1xx	b111 bxxxxxxxxx 1xx	Card reading from DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2

Table 2-199. Unit Masks for TXN_REQ_OF_CPU (Sheet 3 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
MEM_READ.PART3	bxxxxx1xx	b111 bxxxxxxxx1 xxx	Card reading from DRAM x4 card is plugged in to slot 3
MEM_READ.PART7	bxxxxx1xx	b111 bxxxx1xxxx xxx	Card reading from DRAM x4 card is plugged in to slot 7
MEM_READ.IOMMU0	bxxxxx1xx	b111 bxxx1xxxxx xxx	Card reading from DRAM IOMMU - Type 0
MEM_READ.PART1	bxxxxx1xx	b111 bxxxxxxxxx x1x	Card reading from DRAM x4 card is plugged in to slot 1
MEM_READ.IOMMU1	bxxxxx1xx	b111 bxx1xxxxxx xxx	Card reading from DRAM IOMMU - Type 1
MEM_READ.PART4	bxxxxx1xx	b111 bxxxxxxx1x xxx	Card reading from DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MEM_READ.PART0	bxxxxx1xx	b111 bxxxxxxxxx xx1	Card reading from DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MEM_READ.PART5	bxxxxx1xx	b111 bxxxxxx1xx xxx	Card reading from DRAM x4 card is plugged in to slot 5
MEM_READ.PART6	bxxxxx1xx	b111 bxxxxx1xxx xxx	Card reading from DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
PEER_READ.PART5	bxxxx1xxx	b111 bxxxxxx1xx xxx	Card reading from another Card (same or different stack) x4 card is plugged in to slot 5
PEER_READ.IOMMU0	bxxxx1xxx	b111 bxxx1xxxxx xxx	Card reading from another Card (same or different stack) IOMMU - Type 0
PEER_READ.PART2	bxxxx1xxx	b111 bxxxxxxxxx 1xx	Card reading from another Card (same or different stack) x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
PEER_READ.PART1	bxxxx1xxx	b111 bxxxxxxxxx x1x	Card reading from another Card (same or different stack) x4 card is plugged in to slot 1
PEER_READ.PART0	bxxxx1xxx	b111 bxxxxxxxxx xx1	Card reading from another Card (same or different stack) x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
PEER_READ.IOMMU1	bxxxx1xxx	b111 bxx1xxxxxx xxx	Card reading from another Card (same or different stack) IOMMU - Type 1
PEER_READ.PART3	bxxxx1xxx	b111 bxxxxxxxx1 xxx	Card reading from another Card (same or different stack) x4 card is plugged in to slot 3
PEER_READ.PART6	bxxxx1xxx	b111 bxxxxx1xxx xxx	Card reading from another Card (same or different stack) x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6

Table 2-199. Unit Masks for TXN_REQ_OF_CPU (Sheet 4 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
PEER_READ.PART4	bxxxx1xxx	b111 bxxxxxxx1xxxx	Card reading from another Card (same or different stack) x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
PEER_READ.PART7	bxxxx1xxx	b111 bxxxx1xxxxxxx	Card reading from another Card (same or different stack) x4 card is plugged in to slot 7
ATOMIC.PART5	bxxx1xxxx	b111 bxxxxxx1xxxx	Atomic requests targeting DRAM x4 card is plugged in to slot 5
ATOMIC.IOMMU0	bxxx1xxxx	b111 bxxx1xxxxxxx	Atomic requests targeting DRAM IOMMU - Type 0
ATOMIC.PART2	bxxx1xxxx	b111 bxxxxxxxxx1xx	Atomic requests targeting DRAM x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
ATOMIC.IOMMU1	bxxx1xxxx	b111 bxx1xxxxxxxx	Atomic requests targeting DRAM IOMMU - Type 1
ATOMIC.PART4	bxxx1xxxx	b111 bxxxxxxx1xxx	Atomic requests targeting DRAM x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
ATOMIC.PART3	bxxx1xxxx	b111 bxxxxxxxx1xxx	Atomic requests targeting DRAM x4 card is plugged in to slot 3
ATOMIC.PART0	bxxx1xxxx	b111 bxxxxxxxxxxx1	Atomic requests targeting DRAM x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
ATOMIC.PART1	bxxx1xxxx	b111 bxxxxxxxxxx1x	Atomic requests targeting DRAM x4 card is plugged in to slot 1
ATOMIC.PART6	bxxx1xxxx	b111 bxxxxx1xxxxx	Atomic requests targeting DRAM x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
ATOMIC.PART7	bxxx1xxxx	b111 bxxxx1xxxxxx	Atomic requests targeting DRAM x4 card is plugged in to slot 7
MSG.PART0	bx1xxxxxx	b111 bxxxxxxxxxxx1	Messages x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
MSG.IOMMU0	bx1xxxxxx	b111 bxxx1xxxxxxx	Messages IOMMU - Type 0
MSG.PART1	bx1xxxxxx	b111 bxxxxxxxxx1x	Messages x4 card is plugged in to slot 1
MSG.PART6	bx1xxxxxx	b111 bxxxxx1xxxxx	Messages x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
MSG.PART5	bx1xxxxxx	b111 bxxxxxx1xxxx	Messages x4 card is plugged in to slot 5
MSG.IOMMU1	bx1xxxxxx	b111 bxx1xxxxxxxx	Messages IOMMU - Type 1

Table 2-199. Unit Masks for TXN_REQ_OF_CPU (Sheet 5 of 5)

Extension	umask [15:8]	xtra [50:36]	Description
MSG.PART3	bx1xxxxxx	b111 bxxxxxxxx1xxx	Messages x4 card is plugged in to slot 3
MSG.PART4	bx1xxxxxx	b111 bxxxxxxx1xxxx	Messages x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
MSG.PART2	bx1xxxxxx	b111 bxxxxxxxxx1xx	Messages x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2
MSG.PART7	bx1xxxxxx	b111 bxxxx1xxxxxxx	Messages x4 card is plugged in to slot 7
CMPD.PART0	b1xxxxxxx	b111 bxxxxxxxxxxx1	CmpD - device sending completion to CPU request x16 card plugged in to Lane 0/1/2/3, Or x8 card plugged in to Lane 0/1, Or x4 card is plugged in to slot 0
CMPD.PART7	b1xxxxxxx	b111 bxxxx1xxxxxxx	CmpD - device sending completion to CPU request x4 card is plugged in to slot 7
CMPD.PART3	b1xxxxxxx	b111 bxxxxxxxx1xxx	CmpD - device sending completion to CPU request x4 card is plugged in to slot 3
CMPD.PART6	b1xxxxxxx	b111 bxxxxx1xxxxxx	CmpD - device sending completion to CPU request x8 card plugged in to Lane 6/7, Or x4 card is plugged in to slot 6
CMPD.IOMMU1	b1xxxxxxx	b111 bxx1xxxxxxxxx	CmpD - device sending completion to CPU request IOMMU - Type 1
CMPD.PART1	b1xxxxxxx	b111 bxxxxxxxxxx1x	CmpD - device sending completion to CPU request x4 card is plugged in to slot 1
CMPD.PART5	b1xxxxxxx	b111 bxxxxxx1xxxxx	CmpD - device sending completion to CPU request x4 card is plugged in to slot 5
CMPD.PART4	b1xxxxxxx	b111 bxxxxxxx1xxxx	CmpD - device sending completion to CPU request x16 card plugged in to Lane 4/5/6/7, Or x8 card plugged in to Lane 4/5, Or x4 card is plugged in to slot 4
CMPD.IOMMU0	b1xxxxxxx	b111 bxxx1xxxxxxxx	CmpD - device sending completion to CPU request IOMMU - Type 0
CMPD.PART2	b1xxxxxxx	b111 bxxxxxxxxx1xx	CmpD - device sending completion to CPU request x8 card plugged in to Lane 2/3, Or x4 card is plugged in to slot 2

2.6 IRP Performance Monitoring

IRP (IIO Ring Port) is responsible for maintaining coherency for IIO traffic targeting coherent memory.

2.6.1 IRP Performance Monitoring Overview

Each IRP Box supports event monitoring through two 48b wide counters (IRP{5-0}_MSR_PMON_CTR/CTL{1:0}). Each of these counters can be programmed to count any IRP event. The IRP counters can increment by a maximum of 7b per cycle.

2.6.2 IRP Performance Monitoring Events

IRP provides events to track information related to all the traffic passing through it's boundaries.

- Write Cache Occupancy
- Ingress/Egress Traffic - by Ring Type
- Stalls awaiting Credit
- FAF ("Fire and Forget") Queue

2.6.3 IRP Box Events Ordered By Code

The following table summarizes the directly measured IRP Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
CLOCKTICKS	0x01	0-1	1	Clockticks of the IO coherency tracker (IRP)
TxC_BL_DRS_INSERTS	0x02	0-1	1	BL DRS Egress Inserts
TxC_BL_NCB_INSERTS	0x03	0-1	1	BL NCB Egress Inserts
TxC_BL_NCS_INSERTS	0x04	0-1	1	BL NCS Egress Inserts
TxC_BL_DRS_CYCLES_FULL	0x05	0-1	1	BL DRS Egress Cycles Full
TxC_BL_NCB_CYCLES_FULL	0x06	0-1	1	BL NCB Egress Cycles Full
TxC_BL_NCS_CYCLES_FULL	0x07	0-1	1	BL NCS Egress Cycles Full
TxC_BL_DRS_OCCUPANCY	0x08	0-1	63	BL DRS Egress Occupancy
TxC_BL_NCB_OCCUPANCY	0x09	0-1	31	BL NCB Egress Occupancy
TxC_BL_NCS_OCCUPANCY	0x0a	0-1	15	BL NCS Egress Occupancy
TxC_AK_INSERTS	0x0b	0-1	1	AK Egress Allocations
TxS_REQUEST_OCCUPANCY	0x0c	0-1	1	Outbound Request Queue Occupancy
TxS_DATA_INSERTS_NCB	0x0d	0-1	1	Outbound Read Requests
TxS_DATA_INSERTS_NCS	0x0e	0-1	1	Outbound Read Requests
CACHE_TOTAL_OCCUPANCY	0x0f	0-1	511	Total Write Cache Occupancy
COHERENT_OPS	0x10	0-1	2	Coherent Ops
TRANSACTIONS	0x11	0-1	1	Inbound Transaction Count
SNOOP_RESP	0x12	0-1	2	Snoop Responses
P2P_TRANSACTIONS	0x13	0-1	1	P2P Transactions
P2P_INSERTS	0x14	0-1	1	P2P Requests
P2P_OCCUPANCY	0x15	0-1	63	P2P Occupancy
FAF_TRANSACTIONS	0x16	0-1	1	FAF allocation -- sent to ADQ
FAF_FULL	0x17	0-1	1	FAF RF full
FAF_INSERTS	0x18	0-1	1	FAF - request insert from TC.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
FAF_OCCUPANCY	0x19	0-1	31	FAF occupancy
TxR2_AD0_STALL_CREDIT_CYCLES	0x1a	0-1	1	No AD0 Egress Credits Stalls
TxR2_AD1_STALL_CREDIT_CYCLES	0x1b	0-1	1	No AD1 Egress Credits Stalls
TxR2_AD01_STALL_CREDIT_CYCLES	0x1c	0-1	3	
TxR2_BL_STALL_CREDIT_CYCLES	0x1d	0-1	1	No BL Egress Credit Stalls
MISC0	0x1e	0-1	2	Counts Timeouts - Set 0
MISC1	0x1f	0-1	2	Misc Events - Set 1
IRP_ALL	0x20	0-1	3	

2.6.4 IRP Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the IRP Box.

CACHE_TOTAL_OCCUPANCY

- **Title:** Total Write Cache Occupancy
- **Category:** WRITE_CACHE Events
- **Event Code:** 0x0f
- **Max. Inc/Cyc:.** 511

Register Restrictions: 0-1

- **Definition:** Accumulates the number of reads and writes that are outstanding in the uncore in each cycle. This is effectively the sum of the READ_OCCUPANCY and WRITE_OCCUPANCY events.

Table 2-200. Unit Masks for CACHE_TOTAL_OCCUPANCY

Extension	umask [15:8]	Description
ANY	b00000001	Any Source Tracks all requests from any source port.
IV_Q	b00000010	Snoops
MEM	b00000100	Mem

CLOCKTICKS

- **Title:** Clockticks of the IO coherency tracker (IRP)
- **Category:** CLOCK Events
- **Event Code:** 0x01
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

COHERENT_OPS

- **Title:** Coherent Ops
- **Category:** Coherency Events
- **Event Code:** 0x10
- **Max. Inc/Cyc:.** 2

Register Restrictions: 0-1

- **Definition:** Counts the number of coherency related operations servied by the IRP

Table 2-201. Unit Masks for COHERENT_OPS

Extension	umask [15:8]	Description
PCIRDCUR	bxxxxxxx1	PCIRdCur
RFO	bxxxx1xxx	RFO
PCITOM	bxxx1xxxx	PCIItoM
WBMTOI	bx1xxxxxx	WbMtoI
CLFLUSH	b1xxxxxxx	CLFlush

FAF_FULL

- **Title:** FAF RF full
- **Category:** FAF Events
- **Event Code:** 0x17
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

FAF_INSERTS

- **Title:** FAF - request insert from TC.
- **Category:** FAF Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Read transactions
- **NOTE:** Read prefetch transactions no longer go through M2IOSF

FAF_OCCUPANCY

- **Title:** FAF occupancy
- **Category:** FAF Events
- **Event Code:** 0x19
- **Max. Inc/Cyc:.** 31

Register Restrictions: 0-1

- **Definition:**

FAF_TRANSACTIONS

- **Title:** FAF allocation -- sent to ADQ
- **Category:** FAF Events
- **Event Code:** 0x16
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

IRP_ALL

- **Title:**
- **Category:** IRP Buffer Events
- **Event Code:** 0x20
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-1

- **Definition:**

Table 2-202. Unit Masks for IRP_ALL

Extension	umask [15:8]	Description
INBOUND_INSERTS	b00000001	All Inserts Inbound (p2p + faf + cset)
OUTBOUND_INSERTS	b00000010	All Inserts Outbound (BL, AK, Snoops)
EVICTS	b00000100	All Inserts Outbound (BL, AK, Snoops)

MISC0

- **Title:** Counts Timeouts - Set 0
- **Category:** MISC Events
- **Event Code:** 0x1e
- **Max. Inc/Cyc:.** 2

Register Restrictions: 0-1

- **Definition:**

Table 2-203. Unit Masks for MISC0

Extension	umask [15:8]	Description
FAST_REQ	b000000x1	Fastpath Requests
FAST_REJ	b0000001x	Fastpath Rejects
2ND_RD_INSERT	bx00xx100	Cache Inserts of Read Transactions as Secondary
2ND_WR_INSERT	bx00x1x00	Cache Inserts of Write Transactions as Secondary
2ND_ATOMIC_INSERT	bx001xx00	Cache Inserts of Atomic Transactions as Secondary
FAST_XFER	bxx100000	Fastpath Transfers From Primary to Secondary
PF_ACK_HINT	bx1x00000	Prefetch Ack Hints From Primary to Secondary
SLOWPATH_FWPF_NO_PRF	b1xx00000	Slow path fwpf didn't find prefetch

MISC1

- **Title:** Misc Events - Set 1
- **Category:** MISC Events
- **Event Code:** 0x1f
- **Max. Inc/Cyc:.** 2

Register Restrictions: 0-1

- **Definition:**

Table 2-204. Unit Masks for MISC1

Extension	umask [15:8]	Description
SLOW_I	b000xxxx1	Slow Transfer of I Line Snoop took cacheline ownership before write from data was committed.
SLOW_S	b000xxx1x	Slow Transfer of S Line Secondary received a transfer that did not have sufficient MESI state
SLOW_E	b000xx1xx	Slow Transfer of E Line Secondary received a transfer that did have sufficient MESI state
SLOW_M	b000x1xxx	Slow Transfer of M Line Snoop took cacheline ownership before write from data was committed.
LOST_FWD	b0001xxxx	Lost Forward Snoop pulled away ownership before a write was committed
SEC_RCVD_INVLD	bxx1x0000	Received Invalid Secondary received a transfer that did not have sufficient MESI state
SEC_RCVD_VLD	bx1xx0000	Received Valid Secondary received a transfer that did have sufficient MESI state

P2P_INSERTS

- **Title:** P2P Requests
- **Category:** P2P Events
- **Event Code:** 0x14
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** P2P requests from the ITC

P2P_OCCUPANCY

- **Title:** P2P Occupancy
- **Category:** P2P Events
- **Event Code:** 0x15
- **Max. Inc/Cyc:.** 63

Register Restrictions: 0-1

- **Definition:** P2P B & S Queue Occupancy

P2P_TRANSACTIONS

- **Title:** P2P Transactions
- **Category:** P2P Events
- **Event Code:** 0x13
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**
- **NOTE:** Top 4 bits can be Ored. Bottom 4 bits allow specific filtering. .REM = 0 and .LOC = 0 == ALL. .REM = 1 and .LOC = 1 == ALL. .REM = 1 and .LOC = 0 then you just measure remote traffic. .REM = 0 and .LOC = 1 then you just measure local traffic of whatever type of traffic chosen in first 4 bits. Unsure how to define P2P target. Some HW ID?

Table 2-205. Unit Masks for P2P_TRANSACTIONS

Extension	umask [15:8]	Description
RD	bxxxxxxx1	P2P reads
WR	bxxxxxx1x	P2P Writes
MSG	bxxxxx1xx	P2P Message
CMPL	bxxxx1xxx	P2P completions
REM	bxxx1xxxx	Match if remote only
REM_AND_TGT_MATCH	bxx1xxxxx	match if remote and target matches
LOC	bx1xxxxxx	match if local only
LOC_AND_TGT_MATCH	b1xxxxxxx	match if local and target matches

SNOOP_RESP

- **Title:** Snoop Responses
- **Category:** TRANSACTIONS Events
- **Event Code:** 0x12
- **Max. Inc/Cyc:.** 2

Register Restrictions: 0-1

- **Definition:**
- **NOTE:** The first 4 subevent bits are the Responses to the Code/Data/Invalid Snoops represented by the last 3 subevent bits. At least 1 of the bottom 4 bits must be combined with 1 of the top 3 bits to obtain counts. Unsure which combinations are possible.

Table 2-206. Unit Masks for SNOOP_RESP

Extension	umask [15:8]	Description
MISS	bxxxxxxx1	Miss
HIT_I	bxxxxxx1x	Hit I
HIT_ES	bxxxxx1xx	Hit E or S
HIT_M	bxxxx1xxx	Hit M
SNPCODE	bxxx1xxxx	SnpCode
SNPDATA	bxx1xxxxx	SnpData
SNPINV	bx1xxxxxx	SnpInv
ALL_MISS	b01110001	All Misses
ALL_HIT_I	b01110010	All Hits to I
ALL_HIT_ES	b01110100	All Hits to E or S
ALL_HIT_M	b01111000	All Hits to M
ALL_HIT	b01111110	All Hits

TRANSACTIONS

- **Title:** Inbound Transaction Count
- **Category:** TRANSACTIONS Events
- **Event Code:** 0x11
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number of "Inbound" transactions from the IRP to the Uncore. This can be filtered based on request type in addition to the source queue. Note the special filtering equation. We do OR-reduction on the request type. If the SOURCE bit is set, then we also do AND qualification based on the source portID.
- **NOTE:** Bit 7 is a filter that can be applied to the other subevents. Meaningless by itself.

Table 2-207. Unit Masks for TRANSACTIONS

Extension	umask [15:8]	Filter Dep	Description
WRITES	bxxxxxx1x		Writes Trackes only write requests. Each write request should have a prefetch, so there is no need to explicitly track these requests. For writes that are tickled and have to retry, the counter will be incremented for each retry.
WR_PREF	bxxxx1xxx		Write Prefetches Tracks the number of write prefetches.
ATOMIC	bxxx1xxxx		Atomic Tracks the number of atomic transactions

Table 2-207. Unit Masks for TRANSACTIONS

Extension	umask [15:8]	Filter Dep	Description
OTHER	bxx1xxxxx		Other Tracks the number of 'other' kinds of transactions.
ORDERINGQ	bx1xxxxxx	IRPFilter[4:0]	Select Source Tracks only those requests that come from the port specified in the IRP_PmonFilter.OrderingQ register. This register allows one to select one specific queue. It is not possible to monitor multiple queues at a time. If this bit is not set, then requests from all sources will be counted.

TxC_AK_INSERTS

- **Title:** AK Egress Allocations
- **Category:** AK Egress Events
- **Event Code:** 0x0b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_DRS_CYCLES_FULL

- **Title:** BL DRS Egress Cycles Full
- **Category:** BL Egress Events
- **Event Code:** 0x05
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_DRS_INSERTS

- **Title:** BL DRS Egress Inserts
- **Category:** BL Egress Events
- **Event Code:** 0x02
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_DRS_OCCUPANCY

- **Title:** BL DRS Egress Occupancy
- **Category:** BL Egress Events
- **Event Code:** 0x08
- **Max. Inc/Cyc:.** 63

Register Restrictions: 0-1

- **Definition:**

TxC_BL_NCB_CYCLES_FULL

- **Title:** BL NCB Egress Cycles Full
- **Category:** BL Egress Events
- **Event Code:** 0x06
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_NCB_INSERTS

- **Title:** BL NCB Egress Inserts
- **Category:** BL Egress Events
- **Event Code:** 0x03
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_NCB_OCCUPANCY

- **Title:** BL NCB Egress Occupancy
- **Category:** BL Egress Events
- **Event Code:** 0x09
- **Max. Inc/Cyc:.** 31

Register Restrictions: 0-1

- **Definition:**

TxC_BL_NCS_CYCLES_FULL

- **Title:** BL NCS Egress Cycles Full
- **Category:** BL Egress Events
- **Event Code:** 0x07
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_NCS_INSERTS

- **Title:** BL NCS Egress Inserts
- **Category:** BL Egress Events
- **Event Code:** 0x04
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:**

TxC_BL_NCS_OCCUPANCY

- **Title:** BL NCS Egress Occupancy
- **Category:** BL Egress Events
- **Event Code:** 0x0a
- **Max. Inc/Cyc:.** 15

Register Restrictions: 0-1

- **Definition:**

TxR2_AD01_STALL_CREDIT_CYCLES

- **Title:**
- **Category:** STALL_CYCLES Events
- **Event Code:** 0x1c
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-1

- **Definition:** Counts the number times when it is not possible to issue a request to the M2PCIe because there are no Egress Credits available on AD0, A1 or AD0&AD1 both. Stalls on both AD0 and AD1 will count as 2

TxR2_AD0_STALL_CREDIT_CYCLES

- **Title:** No AD0 Egress Credits Stalls
- **Category:** STALL_CYCLES Events
- **Event Code:** 0x1a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number times when it is not possible to issue a request to the M2PCIe because there are no AD0 Egress Credits available.

TxR2_AD1_STALL_CREDIT_CYCLES

- **Title:** No AD1 Egress Credits Stalls
- **Category:** STALL_CYCLES Events
- **Event Code:** 0x1b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number times when it is not possible to issue a request to the M2PCIe because there are no AD1 Egress Credits available.

TxR2_BL_STALL_CREDIT_CYCLES

- **Title:** No BL Egress Credit Stalls
- **Category:** STALL_CYCLES Events
- **Event Code:** 0x1d
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number times when it is not possible to issue data to the R2PCIe because there are no BL Egress Credits available.

TxS_DATA_INSERTS_NCB

- **Title:** Outbound Read Requests
- **Category:** OUTBOUND_REQUESTS Events
- **Event Code:** 0x0d
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number of requests issued to the switch (towards the devices).

TxS_DATA_INSERTS_NCS

- **Title:** Outbound Read Requests
- **Category:** OUTBOUND_REQUESTS Events
- **Event Code:** 0x0e
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number of requests issued to the switch (towards the devices).

TxS_REQUEST_OCCUPANCY

- **Title:** Outbound Request Queue Occupancy
- **Category:** OUTBOUND_REQUESTS Events
- **Event Code:** 0x0c
- **Max. Inc/Cyc:.** 1

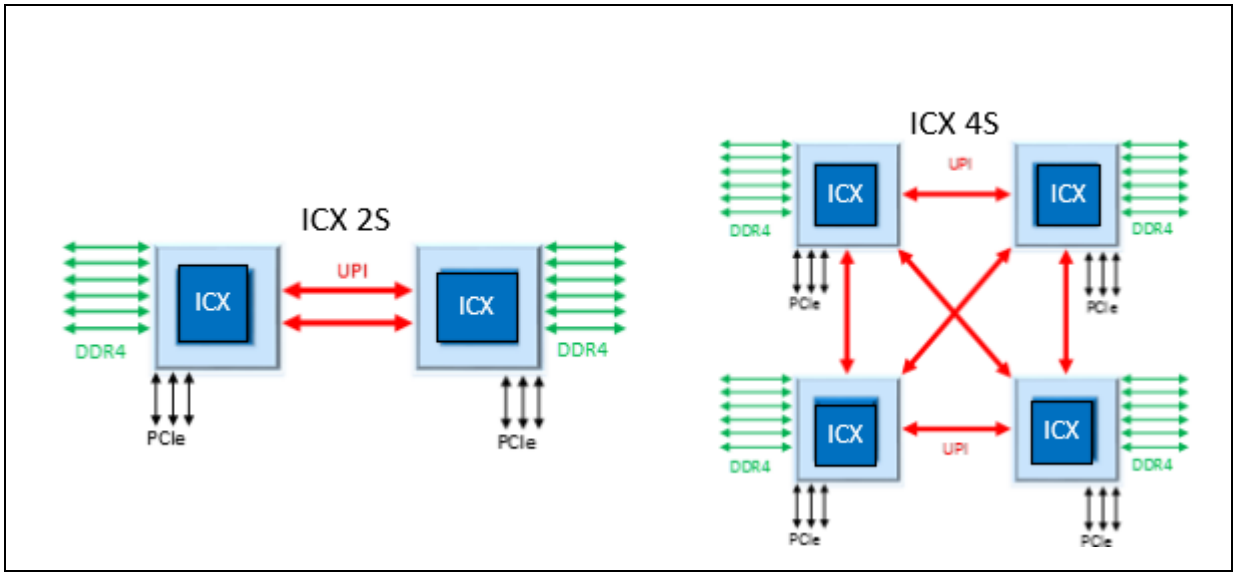
Register Restrictions: 0-1

- **Definition:** Accumultes the number of outstanding outbound requests from the IRP to the switch (towards the devices). This can be used in conjuection with the allocations event in order to calculate average latency of outbound requests.

2.7 Intel® UPI Link Layer Performance Monitoring

3rd Gen Intel Xeon Processor Scalable Family uses a coherent interconnect for scaling to multiple sockets known as Intel® Ultra Path Interconnect (UPI). Intel® UPI technology provides a cache coherent socket to socket external communication interface between processors. The processor implements up to 4 Intel® UPI links depending on the specific product. Figures below show a 2-socket or 4-socket server systems, as examples. Intel® UPI is also used as a coherent communication interface between processors and OEM 3rd party Node Controllers (XNC).

Figure 2-8. Ice Lake Server Intel UPI for Basic Multi Socket Configurations



There are two Intel® UPI agents that share a single mesh stop and a third agent in the EX part with its own mesh stop. These links can be connected to a single destination (such as in DP), or can be connected to two separate destinations (4s Ring or sDP). Therefore, it will be necessary to count Intel® UPI statistics for each agent separately.

The Intel® UPI module supports 1 Intel® UPI link (per mesh stop) and is comprised of the following layers for each Intel® UPI link:

- **Physical Layer** - The Intel® UPI Physical layer (PHY) is a hardware layer that lies between the Link layer above it, and the physical wires that connect to other devices. The Physical layer is further sub-divided into the logical and electrical sub-blocks.
- **Link Layer** - The Intel® UPI link layer bi-directionally converts between protocol layer messages and Link layer Flits, passes them through shared buffers, and manages the flow control information per virtual channel. The link layer also detects errors and retransmits packets on errors.
- **Routing Layer** - The Routing Layer is distributed among all agents that send Intel® UPI messages on the mesh (Intel® UPI, CHA, PCIe, IMC). The Intel® UPI Module provides a routing function to determine the correct mesh stop from which to forward a given packet.
- **Protocol Layer** - The Intel® UPI module does not implement the Protocol Layer. A protocol agent is a proxy for some entity which injects, generates, or services Intel® UPI transactions such as memory requests, interrupts, and so forth. The Protocol Layer is implemented in the following modules: Coherency Home Agent(CHA), PCIe, Configuration Agent (Ubox). A Coherency agent (CA) in the CHA both generates requests and services snoops. A Home Agent (HA) in the CHA services requests, generates snoops, and resolves conflicts. CHA will sometimes behave as CA, sometimes as a HA, and sometimes both at the same time. The PCIe module handles most IO proxy responsibilities. The Ubox handles internal configuration space and some other interrupt and messaging flows. A HA acts as proxy for DRAM, while the PCIe/Ubox handle all non-DRAM (NCB and NCS) requests.

The UPI Subsystem implements a UPI port as a bi-directional interface, where each direction is 20 lanes wide. As such, the UPI Subsystem implements both transmit and receive interfaces and functionality. 3rd Gen Intel Xeon Processor Scalable Family supports two UPI operational speeds - 10.4 GT/s or 11.2 GT/s.

The Intel® UPI Link Layer is responsible for packetizing requests from the caching agent on the way out to the system interface. The UPI link layer processes information at a flit granularity.

On 3rd Gen Intel Xeon Processor Scalable Family, Intel® UPI is split into two layers – M3UPI and the UPI Link Layer. M3UPI (Section 2.10, “M3UPI Performance Monitoring”) provides the interface to the Mesh for the Link Layer. M3UPI converts mesh packets (received from CHA) into UPI flits and vice-versa. M3UPI ingress is the point where remote UPI VNA/VN0 link credits are acquired. UPI Link Layer passes flits through shared buffers, and manages their flow control. The Link layer also detects errors, and on their occurrence, retransmits affected packets (corrected errors). Finally, the Link layer delivers packets to the caching agent.

A single UPI flit can pack up to 3 mesh packets in 3 slots. The Intel® UPI Link Layer has the ability to transmit up to 3 mesh packets per cycle in each direction. In order to accommodate this, many of the events in the Link Layer can increment by 0, 1, or 2 in each cycle. It is not possible to monitor Rx (received) and Tx (transmitted) flit information at the same time on the same counter.

Note: Flit slots are not symmetric in their ability to relay flit traffic. Any analysis of UPI BW should keep this in mind.

2.7.1 Intel® UPI Performance Monitoring Overview

Each Intel® UPI Link supports event monitoring through four 48b wide counters (U_Ly_PCI_PMON_CTR/CTL{3:0}). Each of these four counters can be programmed to count any Intel® UPI event. The Intel® UPI counters can increment by a maximum of 9b per cycle.

Each Intel® UPI Link also includes a mask/match register that allows a user to match packets, according to various standard packet fields such as message class, opcode, and so forth as they leave the UPI Link.

2.7.2 Additional Intel® UPI Performance Monitoring

2.7.2.1 Intel® UPI PMON Counter Control - Difference From Baseline

The following table defines the difference in the layout of the Intel® UPI performance monitor control registers from the baseline presented in Chapter 1.

Figure 2-9. UPI Counter Control Register for 3rd Gen Intel Xeon Processor Scalable Family Server

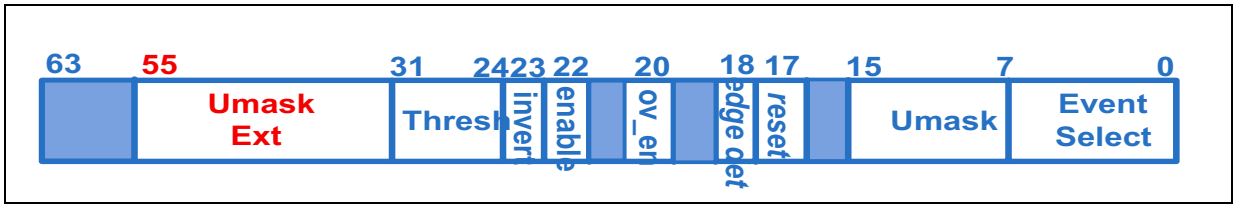


Table 2-208. U_Ly_PCI_PMON_CTL{3-0} Difference from Baseline – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
ig	63:56	RV	0	Reserved
umask_ext	55:32	RW-V	0	Umask Extension - Used for TxL/RxL_BASIC_HDR_MATCH events - allows a opcode, message class, local/remote and a couple other types of matching. - See Table 2-209, "UmaskExt Field Details for TxL/RxL_BASIC_HDR_MATCH.* events"

2.7.2.2 Intel® UPI Basic Header Packet Matching

In 3rd Gen Intel Xeon Processor Scalable Family, the ability to more precisely breakdown traffic by traffic type has been moved to the individual counter control registers.

The matching extension is enabled when measuring events 0x04 (TxL_BASIC_HDR_MATCH) and 0x05 (RxL_BASIC_HDR_MATCH). Filtering can be performed according to the packet's Opcode, Message Class, DNID, RCSNID, and so forth. Details below.

Table 2-209. UmaskExt Field Details for TxL/RxL_BASIC_HDR_MATCH.* events

Field	Bits	Description
llcrd_implnull	55	LLCRD Implied Null - NOTE: Only applies to Slot 2 MC + Opcode Match
llcrd_non0	54	LLCRD Nonzero - NOTE: Only applies to Slot 2 MC + Opcode Match
slot2	53	Slot 2 - NOTE: bits 53:51 == 0x0 is equivalent to bits 53:51 == 0x7. i.e. if NO slot bits are set, all slots are enabled and used for the match.
slot1	52	Slot 1
slot0	51	Slot 0
en_rcsnid	50	Enable RCS Node ID Match - NOTE: Assumes matching applied to MC+Opcode combinations that have an R/C/S NID
rcsnid	49:46	RCS Node ID
en_dnidd	45	Enable Destination Node ID Match
ig	44	Reserved
dnid	43:40	Destination Node ID
isoch	39	Isoch
sglslot	38	Basic Header - Single Slot - Single / Dual are mutually exclusive. Only valid to set 1 bit.
dualslot	37	Basic Header - Dual Slot
nondata	36	Basic Header - NonData - Data / NonData are mutually exclusive. Only valid to set 1 bit.
data	35	Basic Header - Data
rem	34	Remote - received packet is targeting a CHA in a different socket. - NOTE: Only valid for Rx event. - Local / Remote are mutually exclusive. Only valid to set 1 bit.
loc	33	Local - received packet is targeting a CHA in this socket. - NOTE: Only valid for Rx event.
opc	32	Enable Opcode Matching - See Table 3-7, "Intel UPI Opcode Match by Message Class" for available opcodes - The 4b opcode encoding should be written to the upper 4b of the umask field. Counter control bits 15:12

2.7.3 Intel® UPI LL Performance Monitoring Events

The Intel® UPI Link Layer provides events to gather information on topics such as:

- Tracking incoming (mesh bound)/outgoing (system bound) transactions,
- Various queues that track those transactions,
- The Link Layer's power consumption as expressed by the time spent in the Link power states L0p (half of lanes are disabled).
- A variety of static events such as Direct2Core and Direct2UPI statistics and when output credit is unavailable.

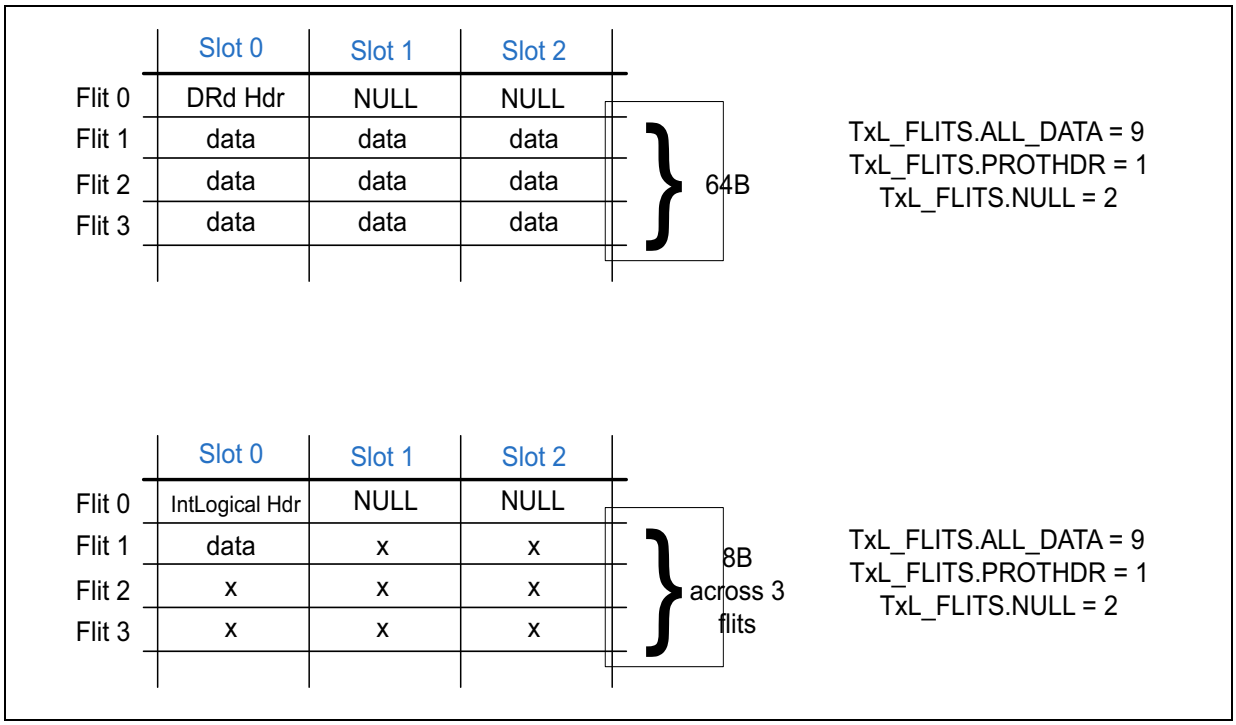
- Of particular interest, total link utilization may be calculated by capturing and subtracting transmitted/received idle flits from Intel® UPI clocks.

Many of these events can be further broken down by message class, including link utilization.

A quick illustration on calculating UPI Bandwidth. Here are two basic examples. The first is a typical DRd (data read) packet and the other is an IntLogical (logically addressed interrupt) packet. The point is, in both these cases, the number of flits sent are the same even in the rare case a full cacheline's worth of data isn't transmitted.

When measuring the amount of bandwidth consumed by transmission of the data (i.e. NOT including the header), it should be .ALL_DATA / 9 * 64B.

Figure 2-10. Calculating UPI Bandwidth. Quick example



Note: If the DRd header takes up more than one slot or some other packet header (one without data) is packed in with the DRd Hdr flit, then .PROTHDR >1 and .NULL <2.

2.7.4 Intel UPI LL Box Events Ordered By Code

The following table summarizes the directly measured Intel UPI LL Box events.

Symbol Name	Event Code	Ctrs	Extra Select Bit	Max Inc/ Cyc	Description
CLOCKTICKS	0x01	0-3	0	1	Number of kfclks
TxL_FLITS	0x02	0-3	0	3	Valid Flits Sent
RxL_FLITS	0x03	0-3	0	3	Valid Flits Received
TxL_BASIC_HDR_MATCH	0x04	0-3	0	3	Matches on Transmit path of a UPI Port

Symbol Name	Event Code	Ctrs	Extra Select Bit	Max Inc/ Cyc	Description
RxL_BASIC_HDR_MATCH	0x05	0-3	0	3	Matches on Receive path of a UPI Port
RxL_CRC_LLR_REQ_TRANSMIT	0x08	0-3	0	1	LLR Requests Sent
RxL_CRC_ERRORS	0x0b	0-3	0	1	CRC Errors Detected
DIRECT_ATTEMPTS	0x12	0-3	0	1	Direct packet attempts
M3_BYP_BLOCKED	0x14	0-3	0	1	
M3_RXQ_BLOCKED	0x15	0-3	0	1	
M3_CRD_RETURN_BLOCKED	0x16	0-3	0	1	
FLOWQ_NO_VNA_CRD	0x18	0-3	0	1	
PHY_INIT_CYCLES	0x20	0-3	0	1	Cycles where phy is not in L0, L0c, L0p, L1
L1_POWER_CYCLES	0x21	0-3	0	1	Cycles in L1
POWER_L1_REQ	0x22	0-3	0	1	L1 Req (same as L1 Ack).
POWER_L1_NACK	0x23	0-3	0	1	L1 Req Nack
RxL0_POWER_CYCLES	0x24	0-3	0	1	Cycles in L0
RxL0P_POWER_CYCLES	0x25	0-3	0	1	Cycles in L0p
TxL0_POWER_CYCLES	0x26	0-3	0	1	Cycles in L0
TxL0P_POWER_CYCLES	0x27	0-3	0	1	Cycles in L0p
TxL0P_POWER_CYCLES_LL_ENTER	0x28	0-3	0	1	
TxL0P_POWER_CYCLES_M3_EXIT	0x29	0-3	0	1	
TxL0P_CLK_ACTIVE	0x2a	0-3	0	1	
RxL_INSERTS	0x30	0-3	0	3	RxQ Flit Buffer Allocations
RxL_BYPASSED	0x31	0-3	0	3	RxQ Flit Buffer Bypassed
RxL_OCCUPANCY	0x32	0-3	0	128	RxQ Occupancy - All Packets
RxL_SLOT_BYPASS	0x33	0-3	0	1	
RxL_CREDITS_CONSUMED_VNA	0x38	0-3	0	1	VNA Credit Consumed
RxL_CREDITS_CONSUMED_VN0	0x39	0-3	0	1	VN0 Credit Consumed
RxL_CREDITS_CONSUMED_VN1	0x3a	0-3	0	1	VN1 Credit Consumed
TxL_INSERTS	0x40	0-3	0	1	Tx Flit Buffer Allocations
TxL_BYPASSED	0x41	0-3	0	1	Tx Flit Buffer Bypassed
TxL_OCCUPANCY	0x42	0-3	0	32	Tx Flit Buffer Occupancy
VNA_CREDIT_RETURN_OCCUPANCY	0x44	0-3	0	128	VNA Credits Pending Return - Occupancy
VNA_CREDIT_RETURN_BLOCKED_VN01	0x45	0-3	0	1	
REQ_SLOT2_FROM_M3	0x46	0-3	0	1	

2.7.5 Intel UPI LL Box Common Metrics (Derived Events)

The following table summarizes metrics commonly calculated from Intel UPI LL Box events.

Symbol Name: Definition	Equation
DRS_E_FROM_UPI: DRS response in F or E states received from Intel UPI in bytes. To calculate the total data response for each cache line state, it's necessary to add the contribution from three flavors {DataC, DataC_FrcAckCnflt, DataC_Cmp} of data response packets for each cache line state.	RxL_BASIC_HDR_MATCH.{umask,opc}={0x1C,1} * 64
DRS_M_FROM_UPI: Data Response DataM packets received from Intel UPI. Expressed in bytes	RxL_BASIC_HDR_MATCH.{umask,opc}={0x0C,1} * 64
DRS_WB_FROM_UPI: DRS writeback packets received from Intel UPI in bytes. This is the sum of Wb{I,S,E} DRS packets	DRS_WbI_FROM_UPI + DRS_WbS_FROM_UPI + DRS_WbE_FROM_UPI
DRS_WbE_FROM_UPI: DRS writeback 'change M to E state' packets received from Intel UPI in bytes	RxL_BASIC_HDR_MATCH.{umask,opc}={0x2D,1} *64
DRS_WbI_FROM_UPI: DRS writeback 'change M to I state' packets received from Intel UPI in bytes	RxL_BASIC_HDR_MATCH.{umask,opc}={0x0D,1} *64
DRS_WbS_FROM_UPI: DRS writeback 'change M to S state' packets received from Intel UPI in bytes	RxL_BASIC_HDR_MATCH.{umask,opc}={0x1D,1} *64
NCB_DATA_FROM_UPI_TO_NODEx: NCB Data packets (Any - Interrupts) received from Intel UPI sent to Node ID 'x'. Expressed in bytes	RxL_BASIC_HDR_MATCH.{umask,endnid,dnid} = {0xE,1,x} * 64
PCT_LINK_CRC_RETRY_CYCLES: Percent of Cycles the Intel UPI link layer is in retry mode due to CRC errors	RxL_CRC_CYCLES_IN_LLR / CLOCKTICKS
PCT_LINK_FULL_POWER_CYCLES: Percent of Cycles the Intel UPI link is at Full Power	RxL0_POWER_CYCLES / CLOCKTICKS
PCT_LINK_HALF_DISABLED_CYCLES: Percent of Cycles the Intel UPI link in power mode where half of the lanes are disabled.	RxL0P_POWER_CYCLES / CLOCKTICKS
PCT_LINK_SHUTDOWN_CYCLES: Percent of Cycles the Intel UPI link is Shutdown	L1_POWER_CYCLES / CLOCKTICKS
UPI_SPEED: UPI Speed - In GT/s (GigaTransfers / Second) - Max Intel UPI Bandwidth is 2 * ROUND (Intel UPI Speed, 0)	ROUND ((CLOCKTICKS / TSC) * TSC_SPEED, 0) * (8 / 1000)

2.7.6 Intel UPI LL Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the Intel UPI LL Box.

CLOCKTICKS

- **Title:** Number of kfclks
- **Category:** CFCLK Events
- **Event Code:** 0x01
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of clocks in the Intel UPI LL. This clock runs at 1/8th the "GT/s" speed of the Intel UPI link. For example, a 8GT/s link will have qfclk or 1 GHz. Current products do not support dynamic link speeds, so this frequency is fixed.

DIRECT_ATTEMPTS

- **Title:** Direct packet attempts
- **Category:** DIRECT2CORE Events
- **Event Code:** 0x12
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of DRS packets that we attempted to do direct2core/ direct2UPI on. There are 4 mutually exclusive filters. Filter [0] can be used to get successful spawns, while [1:3] provide the different failure cases. Note that this does not count packets that are not candidates for Direct2Core. The only candidates for Direct2Core are DRS packets destined for Cbos.

Table 2-210. Unit Masks for DIRECT_ATTEMPTS

Extension	umask [15:8]	Description
D2C	bxxxxxxx1	D2C
D2K	bxxxxxx1x	D2K

FLOWQ_NO_VNA_CRD

- **Title:**
- **Category:** LL to M3 Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-211. Unit Masks for FLOWQ_NO_VNA_CRD

Extension	umask [15:8]	Description
AD_VNA_EQ0	bxxxxxxx1	
AD_VNA_EQ1	bxxxxxx1x	
AD_VNA_EQ2	bxxxxx1xx	
BL_VNA_EQ0	bxxxx1xxx	
AK_VNA_EQ0	bxxx1xxxx	
AK_VNA_EQ1	bxx1xxxxx	
AK_VNA_EQ2	bx1xxxxxx	
AK_VNA_EQ3	b1xxxxxxx	

L1_POWER_CYCLES

- **Title:** Cycles in L1
- **Category:** POWER Events
- **Event Code:** 0x21
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Intel UPI qfclk cycles spent in L1 power mode. L1 is a mode that totally shuts down an Intel UPI link. Use edge detect to count the number of instances when the Intel UPI link entered L1. Link power states are per link and per direction, so for example the Tx direction could be in one state while Rx was in another. Because L1 totally shuts down the link, it takes a good amount of time to exit this mode.

M3_BYP_BLOCKED

- **Title:**
- **Category:** LL to M3 Events
- **Event Code:** 0x14
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-212. Unit Masks for M3_BYP_BLOCKED

Extension	umask [15:8]	Description
FLOWQ_AD_VNA_LE2	bxxxxxxx1	
FLOWQ_BL_VNA_EQ0	bxxxxxx1x	
FLOWQ_AK_VNA_LE3	bxxxxx1xx	
BGF_CRD	bxxxx1xxx	
GV_BLOCK	bxxx1xxxx	

M3_CRD_RETURN_BLOCKED

- **Title:**
- **Category:** LL to M3 Events
- **Event Code:** 0x16
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

M3_RXQ_BLOCKED

- **Title:**
- **Category:** LL to M3 Events
- **Event Code:** 0x15
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-213. Unit Masks for M3_RXQ_BLOCKED

Extension	umask [15:8]	Description
FLOWQ_AD_VNA_LE2	bxxxxxxx1	
FLOWQ_AD_VNA_BTW_2_THRESH	bxxxxxx1x	
FLOWQ_BL_VNA_EQ0	bxxxxx1xx	
FLOWQ_BL_VNA_BTW_0_THRESH	bxxxx1xxx	

Table 2-213. Unit Masks for M3_RXQ_BLOCKED

Extension	umask [15:8]	Description
FLOWQ_AK_VNA_LE3	bxxx1xxxx	
BGF_CRD	bxx1xxxxx	
GV_BLOCK	bx1xxxxxx	

PHY_INIT_CYCLES

- **Title:** Cycles where phy is not in L0, L0c, L0p, L1
- **Category:** POWER Events
- **Event Code:** 0x20
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

POWER_L1_NACK

- **Title:** L1 Req Nack
- **Category:** POWER Events
- **Event Code:** 0x23
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times a link sends/receives a LinkReqNAck. When the Intel UPI links would like to change power state, the Tx side initiates a request to the Rx side requesting to change states. This requests can either be accepted or denied. If the Rx side replies with an Ack, the power mode will change. If it replies with NAck, no change will take place. This can be filtered based on Rx and Tx. An Rx LinkReqNAck refers to receiving an NAck (meaning this agent's Tx originally requested the power change). A Tx LinkReqNAck refers to sending this command (meaning the peer agent's Tx originally requested the power change and this agent accepted it).
- **NOTE:** L1 only

POWER_L1_REQ

- **Title:** L1 Req (same as L1 Ack).
- **Category:** POWER Events
- **Event Code:** 0x22
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times a link sends/receives a LinkReqAck. When the Intel UPI links would like to change power state, the Tx side initiates a request to the Rx side requesting to change states. This requests can either be accepted or denied. If the Rx side replies with an Ack, the power mode will change. If it replies with NAck, no change will take place. This can be filtered based on Rx and Tx. An Rx LinkReqAck refers to receiving an Ack (meaning this agent's Tx originally requested the power change). A Tx LinkReqAck refers to sending this command (meaning the peer agent's Tx originally requested the power change and this agent accepted it).
- **NOTE:** L1 only

REQ_SLOT2_FROM_M3

- **Title:**
- **Category:** VNA_CREDIT_RETURN Events
- **Event Code:** 0x46
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-214. Unit Masks for REQ_SLOT2_FROM_M3

Extension	umask [15:8]	Description
VNA	bxxxxxxx1	
VN0	bxxxxxx1x	
VN1	bxxxxx1xx	
ACK	bxxxx1xxx	

RxL0P_POWER_CYCLES

- **Title:** Cycles in L0p
- **Category:** POWER_RX Events
- **Event Code:** 0x25
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Intel UPI qfclk cycles spent in L0p power mode. L0p is a mode where we disable 1/2 of the Intel UPI lanes, decreasing our bandwidth in order to save power. It increases snoop and data transfer latencies and decreases overall bandwidth. This mode can be very useful in NUMA optimized workloads that largely only utilize Intel UPI for snoops and their responses. Use edge detect to count the number of instances when the Intel UPI link entered L0p. Link power states are per link and per direction, so for example the Tx direction could be in one state while Rx was in another.
- **NOTE:** Using .edge_det to count transitions does not function if L1_POWER_CYCLES > 0.

RxL0_POWER_CYCLES

- **Title:** Cycles in L0
- **Category:** POWER_RX Events
- **Event Code:** 0x24
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Intel UPI qfclk cycles spent in L0 power mode in the Link Layer. L0 is the default mode which provides the highest performance with the most power. Use edge detect to count the number of instances that the link entered L0. Link power states are per link and per direction, so for example the Tx direction could be in one state while Rx was in another. The phy layer sometimes leaves L0 for training, which will not be captured by this event.

RxL_BASIC_HDR_MATCH

- **Title:** Matches on Receive path of an Intel UPI Port
- **Category:** FLIT match Events
- **Event Code:** 0x05
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Filter Dependency:** CtrCtrl[55:32]
- **Definition:** Matches on Receive path of an Intel UPI port.
- **NOTE:** This event is subject to finer grain filtering. See doc for more information. Components (message class, opcode, local, remote, datahdr, ndatahdr, dual slot header, single slot header and pe) ANDed per Slot. Then slots are ORed.

Table 2-215. Unit Masks for RxL_BASIC_HDR_MATCH

Extension	umask [15:8]	xtra [32]	Description
REQ_OPC	bXXXX1000	0x1	Request, Match Opcode
REQ	bxxxx1000	0x0	Request
SNP	bxxxx1001	0x0	Snoop
SNP_OPC	bXXXX1001	0x1	Snoop, Match Opcode
RSP_NODATA_OPC	bXXXX1010	0x1	Response - No Data, Match Opcode
RSP_NODATA	bxxxx1010	0x0	Response - No Data
RSP_DATA	bxxxx1100	0x0	Response - Data
RSP_DATA_OPC	bXXXX1100	0x1	Response - Data, Match Opcode
WB_OPC	bxxxx1101	0x1	Writeback, Match Opcode
WB	bxxxx1101	0x0	Writeback
NCB_OPC	bxxxx1110	0x1	Non-Coherent Bypass, Match Opcode
NCB	bxxxx1110	0x0	Non-Coherent Bypass
NCS_OPC	bxxxx1111	0x1	Non-Coherent Standard, Match Opcode
NCS	bxxxx1111	0x0	Non-Coherent Standard
RSPI	b00101010	0x1	Response - Invalid
RSPCNFLT	b10101010	0x1	Response - Conflict

RxL_BYPASSED

- **Title:** RxQ Flit Buffer Bypassed
- **Category:** RXQ Events
- **Event Code:** 0x31
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:** Counts the number of times that an incoming flit was able to bypass the flit buffer and pass directly across the BGF and into the Egress. This is a latency optimization, and should generally be the common case. If this value is less than the number of flits transfered, it implies that there was queuing getting onto the ring, and thus the transactions saw higher latency.

Table 2-216. Unit Masks for RxL_BYPASSED

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Slot 0
SLOT1	bxxxxxx1x	Slot 1
SLOT2	bxxxxx1xx	Slot 2

RxL_CRC_ERRORS

- **Title:** CRC Errors Detected
- **Category:** CRC_ERRORS_RX Events
- **Event Code:** 0x0b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of CRC errors detected in the Intel UPI Agent. Each Intel UPI flit incorporates 8 bits of CRC for error detection. This counts the number of flits where the CRC was able to detect an error. After an error has been detected, the Intel UPI agent will send a request to the transmitting socket to resend the flit (as well as any flits that came after it).

RxL_CRC_LLR_REQ_TRANSMIT

- **Title:** LLR Requests Sent
- **Category:** CRC_ERRORS_RX Events
- **Event Code:** 0x08
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of LLR Requests were transmitted. This should generally be <= the number of CRC errors detected. If multiple errors are detected before the Rx side receives a LLC_REQ_ACK from the Tx side, there is no need to send more LLR_REQ_NACKs.
- **NOTE:** We detected an error in Rx, and so we transmit to enter LLR mode. If we get an error and we have not yet received the LLR_ACK, we will not send another request (unless we timeout when we will send another).

RxL_CREDITS_CONSUMED_VN0

- **Title:** VN0 Credit Consumed
- **Category:** RX_CREDITS_CONSUMED Events
- **Event Code:** 0x39
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times that an RxQ VN0 credit was consumed (i.e. message uses a VN0 credit for the Rx Buffer). This includes packets that went through the RxQ and those that were bypassed.

RxL_CREDITS_CONSUMED_VN1

- **Title:** VN1 Credit Consumed
- **Category:** RX_CREDITS_CONSUMED Events
- **Event Code:** 0x3a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times that an RxQ VN1 credit was consumed (i.e. message uses a VN1 credit for the Rx Buffer). This includes packets that went through the RxQ and those that were bypassed.

RxL_CREDITS_CONSUMED_VNA

- **Title:** VNA Credit Consumed
- **Category:** RX_CREDITS_CONSUMED Events
- **Event Code:** 0x38
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times that an RxQ VNA credit was consumed (i.e. message uses a VNA credit for the Rx Buffer). This includes packets that went through the RxQ and those that were bypassed.

RxL_FLITS

- **Title:** Valid Flits Received
- **Category:** Flit Events
- **Event Code:** 0x03
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:** Shows legal flit time (hides impact of L0p and L0c).
- **NOTE:** When Umask is set to all 1's then all Flits should be counted as 3 since a full flit is counted for each valid slot. By counting all legal flit time we exclude impact of L0p, L0c, and the 5/6 ratio in L0. Slot 0 Dual is counted in slot 0 and slot 1 (as a protocol header)

Table 2-217. Unit Masks for RxL_FLITS

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Slot 0 Count Slot 0 - Other mask bits determine types of headers to count.
SLOT1	bxxxxxx1x	Slot 1 Count Slot 1 - Other mask bits determine types of headers to count.
SLOT2	bxxxxx1xx	Slot 2 Count Slot 2 - Other mask bits determine types of headers to count.
DATA	bxxxx1xxx	Data Count Data Flits (which consume all slots), but how much to count is based on Slot0-2 mask, so count can be 0-3 depending on which slots are enabled for counting.
ALL_DATA	b00001111	All Data
LLCRD	bxxx1xxxx	LLCRD Not Empty Enables counting of LLCRD (with non-zero payload). This only applies to slot 2 since LLCRD is only allowed in slot 2
NULL	bxx1xxxxx	Slot NULL or LLCRD Empty LLCRD with all zeros is treated as NULL. Slot 1 is not treated as NULL if slot 0 is a dual slot. This can apply to slot 0,1, or 2.
ALL_NULL	b00100111	Null FLITs received from any slot
LLCTRL	bx1xxxxxx	LLCTRL Equivalent to an idle packet. Enables counting of slot 0 LLCTRL messages.
IDLE	b01000111	Null FLITs received from any slot
PROTHDR	b1xxxxxxx	Protocol Header Enables count of protocol headers in slot 0,1,2 (depending on slot uMask bits)
NON_DATA	b10010111	All Non Data

RxL_INSERTS

- **Title:** RxQ Flit Buffer Allocations
- **Category:** RXQ Events
- **Event Code:** 0x30
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:** Number of allocations into the Intel UPI Rx Flit Buffer. Generally, when data is transmitted across Intel UPI, it will bypass the RxQ and pass directly to the ring interface. If things back up getting transmitted onto the ring, however, it may need to allocate into this buffer, thus increasing the latency. This event can be used in conjunction with the Flit Buffer Occupancy event in order to calculate the average flit buffer lifetime.

Table 2-218. Unit Masks for RxL_INSERTS

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Slot 0
SLOT1	bxxxxxx1x	Slot 1
SLOT2	bxxxxx1xx	Slot 2

RxL_OCCUPANCY

- **Title:** RxQ Occupancy - All Packets
- **Category:** RXQ Events
- **Event Code:** 0x32
- **Max. Inc/Cyc:.** 128

Register Restrictions: 0-3

- **Definition:** Accumulates the number of elements in the Intel UPI RxQ in each cycle. Generally, when data is transmitted across Intel UPI, it will bypass the RxQ and pass directly to the ring interface. If things back up getting transmitted onto the ring, however, it may need to allocate into this buffer, thus increasing the latency. This event can be used in conjunction with the Flit Buffer Not Empty event to calculate average occupancy, or with the Flit Buffer Allocations event to track average lifetime.

Table 2-219. Unit Masks for RxL_OCCUPANCY

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Slot 0
SLOT1	bxxxxxx1x	Slot 1
SLOT2	bxxxxx1xx	Slot 2

RxL_SLOT_BYPASS

- **Title:**
- **Category:** RXQ Events
- **Event Code:** 0x33
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-220. Unit Masks for RxL_SLOT_BYPASS

Extension	umask [15:8]	Description
S0_RXQ1	bxxxxxxx1	
S0_RXQ2	bxxxxxx1x	
S1_RXQ0	bxxxxx1xx	
S1_RXQ2	bxxxx1xxx	
S2_RXQ0	bxxx1xxxx	
S2_RXQ1	bxx1xxxxx	

TxL0P_CLK_ACTIVE

- **Title:**
- **Category:** POWER_TX Events
- **Event Code:** 0x2a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-221. Unit Masks for TxL0P_CLK_ACTIVE

Extension	umask [15:8]	Description
CFG_CTL	bxxxxxxx1	
RXQ	bxxxxxx1x	
RXQ_BYPASS	bxxxxx1xx	
RXQ_CRED	bxxxx1xxx	
TXQ	bxxx1xxxx	
RETRY	bxx1xxxxx	
DFX	bx1xxxxxx	
SPARE	b1xxxxxxx	

TxL0P_POWER_CYCLES

- **Title:** Cycles in L0p
- **Category:** POWER_TX Events
- **Event Code:** 0x27
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Intel UPI qfclk cycles spent in L0p power mode. L0p is a mode where we disable 1/2 of the Intel UPI lanes, decreasing our bandwidth in order to save power. It increases snoop and data transfer latencies and decreases overall bandwidth. This mode can be very useful in NUMA optimized workloads that largely only utilize Intel UPI for snoops and their responses. Use edge detect to count the number of instances when the Intel UPI link entered L0p. Link power states are per link and per direction, so for example the Tx direction could be in one state while Rx was in another.
- **NOTE:** Using.edge_det to count transitions does not function if L1_POWER_CYCLES > 0.

TxL0P_POWER_CYCLES_LL_ENTER

- **Title:**
- **Category:** POWER_TX Events
- **Event Code:** 0x28
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

TxL0P_POWER_CYCLES_M3_EXIT

- **Title:**
- **Category:** POWER_TX Events
- **Event Code:** 0x29
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

TxL0_POWER_CYCLES

- **Title:** Cycles in L0
- **Category:** POWER_TX Events
- **Event Code:** 0x26
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Intel UPI qfclk cycles spent in L0 power mode in the Link Layer. L0 is the default mode which provides the highest performance with the most power. Use edge detect to count the number of instances that the link entered L0. Link power states are per link and per direction, so for example the Tx direction could be in one state while Rx was in another. The phy layer sometimes leaves L0 for training, which will not be captured by this event.
- **NOTE:** Includes L0p cycles. To get just L0, subtract TxL0P_POWER_CYCLES

TxL_BASIC_HDR_MATCH

- **Title:** Matches on Transmit path of an Intel UPI Port
- **Category:** FLIT match Events
- **Event Code:** 0x04
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Filter Dependency:** CtrCtrl[55:32]
- **Definition:** Matches on Transmit path of an Intel UPI port.
- **NOTE:** This event is subject to finer grain filtering. See doc for more information. Components (message class, opcode, local, remote, datahdr, ndatahdr, dual slot header, single slot header and pe) ANDed per Slot. Then slots are ORed.

Table 2-222. Unit Masks for TxL_BASIC_HDR_MATCH

Extension	umask [15:8]	xtra [32]	Description
REQ	bxxxx1000	0x0	Request
REQ_OPC	bXXXX1000	0x1	Request, Match Opcode
SNP_OPC	bXXXX1001	0x1	Snoop, Match Opcode
SNP	bxxxx1001	0x0	Snoop
RSP_NODATA	bxxxx1010	0x0	Response - No Data

Table 2-222. Unit Masks for TxL_BASIC_HDR_MATCH

Extension	umask [15:8]	xtra [32]	Description
RSP_NODATA_OPC	bXXXX1010	0x1	Response - No Data, Match Opcode
RSP_DATA_OPC	bXXXX1100	0x1	Response - Data, Match Opcode
RSP_DATA	bxxxx1100	0x0	Response - Data
WB_OPC	bxxxx1101	0x1	Writeback, Match Opcode
WB	bxxxx1101	0x0	Writeback
NCB	bxxxx1110	0x0	Non-Coherent Bypass
NCB_OPC	bxxxx1110	0x1	Non-Coherent Bypass, Match Opcode
NCS	bxxxx1111	0x0	Non-Coherent Standard
NCS_OPC	bxxxx1111	0x1	Non-Coherent Standard, Match Opcode
RSPI	b00101010	0x1	Response - Invalid
RSPCNFLT	b10101010	0x1	Response - Conflict

TxL_BYPASSED

- **Title:** Tx Flit Buffer Bypassed
- **Category:** TXQ Events
- **Event Code:** 0x41
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times that an incoming flit was able to bypass the Tx flit buffer and pass directly out the Intel UPI Link. Generally, when data is transmitted across Intel UPI, it will bypass the TxQ and pass directly to the link. However, the TxQ will be used with L0p and when LLR occurs, increasing latency to transfer out to the link.

TxL_FLITS

- **Title:** Valid Flits Sent
- **Category:** Flit Events
- **Event Code:** 0x02
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:** Shows legal flit time (hides impact of L0p and L0c).
- **NOTE:** You can OR any of 5 MSB together and apply against any combination of slots and they will be added together, but a slot MUST be selected.

Table 2-223. Unit Masks for TxL_FLITS (Sheet 1 of 2)

Extension	umask [15:8]	Description
SLOT0	bxxxxxxx1	Slot 0 Count Slot 0 - Other mask bits determine types of headers to count.
SLOT1	bxxxxxx1x	Slot 1 Count Slot 1 - Other mask bits determine types of headers to count.
SLOT2	bxxxxx1xx	Slot 2 Count Slot 2 - Other mask bits determine types of headers to count.
DATA	bxxxx1xxx	Data Count Data Flits (which consume all slots), but how much to count is based on Slot0-2 mask, so count can be 0-3 depending on which slots are enabled for counting.

Table 2-223. Unit Masks for TxL_FLITS (Sheet 2 of 2)

Extension	umask [15:8]	Description
ALL_DATA	b00001111	All Data
LLCRD	bxxx1xxxx	LLCRD Not Empty Enables counting of LLCRD (with non-zero payload). This only applies to slot 2 since LLCRD is only allowed in slot 2
NULL	bxx1xxxxx	Slot NULL or LLCRD Empty LLCRD with all zeros is treated as NULL. Slot 1 is not treated as NULL if slot 0 is a dual slot. This can apply to slot 0,1, or 2.
ALL_NULL	b00100111	Null FLITs transmitted to any slot
LLCTRL	bx1xxxxxx	LLCTRL Equivalent to an idle packet. Enables counting of slot 0 LLCTRL messages.
IDLE	b01000111	Idle
PROTHDR	b1xxxxxxx	Protocol Header Enables count of protocol headers in slot 0,1,2 (depending on slot uMask bits)
NON_DATA	b10010111	All Non Data

TxL_INSERTS

- **Title:** Tx Flit Buffer Allocations
- **Category:** TXQ Events
- **Event Code:** 0x40
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of allocations into the Intel UPI Tx Flit Buffer. Generally, when data is transmitted across Intel UPI, it will bypass the TxQ and pass directly to the link. However, the TxQ will be used with L0p and when LLR occurs, increasing latency to transfer out to the link. This event can be used in conjunction with the Flit Buffer Occupancy event in order to calculate the average flit buffer lifetime.

TxL_OCCUPANCY

- **Title:** Tx Flit Buffer Occupancy
- **Category:** TXQ Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 32

Register Restrictions: 0-3

- **Definition:** Accumulates the number of flits in the TxQ. Generally, when data is transmitted across Intel UPI, it will bypass the TxQ and pass directly to the link. However, the TxQ will be used with L0p and when LLR occurs, increasing latency to transfer out to the link. This can be used with the cycles not empty event to track average occupancy, or the allocations event to track average lifetime in the TxQ.

VNA_CREDIT_RETURN_BLOCKED_VN01

- **Title:**
- **Category:** VNA_CREDIT_RETURN Events
- **Event Code:** 0x45
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

VNA_CREDIT_RETURN_OCCUPANCY

- **Title:** VNA Credits Pending Return - Occupancy
- **Category:** VNA_CREDIT_RETURN Events
- **Event Code:** 0x44
- **Max. Inc/Cyc:.** 128

Register Restrictions: 0-3

- **Definition:** Number of VNA credits in the Rx side that are waiting to be returned back across the link.

2.8 M2M Performance Monitoring

M2M blocks manage the interface between the Mesh (operating on both Mesh and the SMI3 protocol) and the Memory Controllers. M2M acts as intermediary between the local CHA issuing memory transactions to its attached Memory Controller. Commands from M2M to the MC are serialized by a scheduler and only one can cross the interface at a time.

In 2 Level Memory Mode (2LM), DDR4 becomes 'Near' memory acting as a cache to the 'Far' memory PMM DIMM. An important responsibility of M2M is to manage traffic (both commands and data) between the two levels of memory which may reside on the same channel or different channels in the attached memory controller.

The BL transgress is used for NMC evictions (writes to far memory), while the AD transgress routes fill commands (reads).

2.8.1 M2M Performance Monitoring Overview

Each M2M Box supports event monitoring through four 48b wide counters (M2Mn_PCI_PMON_CTR/CTL{3:0}). Each of these four counters can be programmed to count almost any M2M event (see NOTE for exceptions). the M2M counters can increment by a maximum of 5b per cycle.

M2M PMON also includes mask/match registers that allow a user to match packets of traffic heading to DRAM or heading to the Mesh, according to various standard packet fields such as message class, opcode, and so forth.

2.8.2 Additional M2M Performance Monitoring

M2M performance monitoring control registers provide a small amount of additional functionality. The following table defines those cases.

Figure 2-11. M2M Counter Control Register for 3rd Gen Intel Xeon Processor Scalable Family

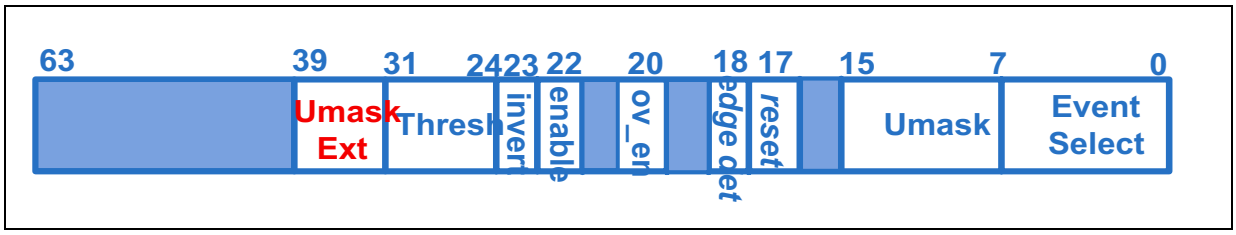


Table 2-224. M2Mn_MSR_PMON_CTL{3-0} Register – Field Definitions Different from baseline

Field	Bits	Attr	HW Reset Val	Description
rsv	63:40	RW-V	0	Reserved. SW must write 0 else behavior is undefined.
umask_ext	39:32	RW-V	0	Extension to umask. Adds additional filtering capabilities to certain special events.

Table 2-225. Additional M2M Performance Monitoring Registers (PCICFG)

Register Name	PCICFG Address	Size (bits)	Description
PCICFG Base Address	Dev:Func DeviceID		
M2M 0 PMON Registers	B2:D8:F0 0x2066		
M2M 1 PMON Registers	B2:D9:F0 0x2066		
Box-Level Filters			
M2Mn_PCI_PMON_OPCODE_MM	278	64	M2M Opcode Mask/Match
M2Mn_PCI_PMON_ADDRMASK0	270	64	M2M PMON Address Mask 0 - b[31:0]
M2Mn_PCI_PMON_ADDRMASK1	274	64	M2M PMON Address Mask 1 - b[45:32]
M2Mn_PCI_PMON_ADDRMATCH0	268	64	M2M PMON Address Match 0 - b[31:0]
M2Mn_PCI_PMON_ADDRMATCH1	26C	64	M2M PMON Address Match 1 - b[45:32]

2.8.2.1 M2M Filter Registers

In addition to generic event counting, each M2M has several registers that allow a user to opcode match and/or address match packet traffic into and out of M2M. The filter registers report the number of matches through the PKT_MATCH event. The opcode mask/match register provides the ability to simultaneously filter incoming and outgoing traffic.

Figure 2-12. M2M PMON Opcode Filter Register

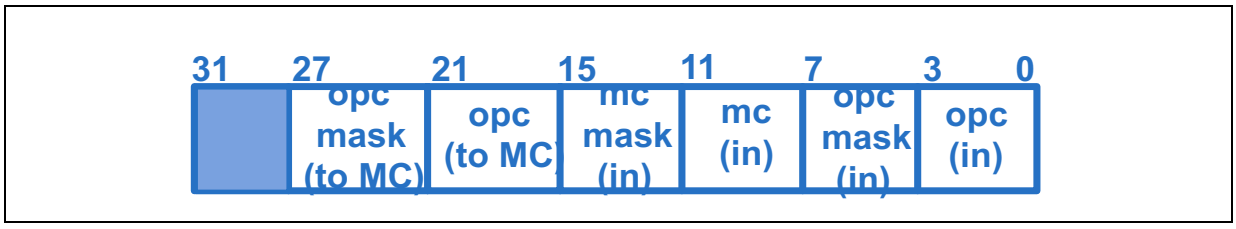


Table 2-226. M2Mn_PCI_PMON_OPCODE_MM Register – Field Definitions

Field	Bits	Atrtr	HW Reset Val	Description
rsv	31:28	RV	0	Reserved SW must set to 0 else behavior is undefined
mcopc_mask	27:22	RW	0	Outgoing Opcode Mask encoding - See the SMI3 encodings in the Packet Match table Table 3-9, "SMI3 Opcode Match by Message Class" on page 326
mcopc	21:16	RW	0	Outgoing Opcode Match encoding - See the SMI3 encodings in the Packet Match table Table 3-9, "SMI3 Opcode Match by Message Class" on page 326
mc_mask	15:12	RW	0	Incoming Message Class Mask
mc	11:8	RW	0	Incoming Message Class Match 4'b0000: REQ (AD) 4'b0010: RSP (AD) 4'b1010: NCB (BL) 4'b1011: NCS (BL) 4'b1100: WB (BL)
opc_mask	7:4	RW	0	Incoming Opcode Mask encoding - See the SMI3 encodings in the Packet Match table Table 3-9, "SMI3 Opcode Match by Message Class" on page 326
opc	3:0	RW	0	Incoming Opcode Match encoding - See the SMI3 encodings in the Packet Match table Table 3-9, "SMI3 Opcode Match by Message Class" on page 326

Table 2-227. M2Mn_PCI_PMON_ADDR{MASK,MATCH}0 Register – Field Definitions

Field	Bits	Atrtr	HW Reset Val	Description
addr	31:0	RW	0	LSB [31:0] of transaction address to mask/match on

Table 2-228. M2Mn_PCI_PMON_ADDR{MASK,MATCH}1 Register – Field Definitions

Field	Bits	Atrtr	HW Reset Val	Description
rsv	31:15	RV	0	Reserved SW must set to 0 else behavior is undefined
addr	14:0	RW	0	MSB [45:32] of transaction address to mask/match on

2.8.3 M2M Performance Monitoring Events

M2M provides events to track information related to all the traffic passing through it's boundaries.

Mesh Section 2.2, "Mesh Performance Monitoring."

- IMC credit tracking - credits rejected, acquired and used all broken down by message Class.

- Reads and Writes issued to the iMC

2.8.4 M2M Box Events Ordered By Code

The following table summarizes the directly measured M2M Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
CLOCKTICKS	0x00		0	Clockticks of the mesh to memory (M2M)
RxC_AD_INSERTS	0x01		0	AD Ingress (from CMS) Allocations
RxC_AD_OCCUPANCY	0x02		0	AD Ingress (from CMS) Occupancy
RxC_AD_CYCLES_NE	0x03		0	AD Ingress (from CMS) Not Empty
RxC_AD_CYCLES_FULL	0x04		0	AD Ingress (from CMS) Full
RxC_BL_INSERTS	0x05		0	BL Ingress (from CMS) Allocations
RxC_BL_OCCUPANCY	0x06		0	BL Ingress (from CMS) Occupancy
RxC_BL_CYCLES_NE	0x07		0	BL Ingress (from CMS) Not Empty
RxC_BL_CYCLES_FULL	0x08		0	BL Ingress (from CMS) Full
TxC_AD_INSERTS	0x09		0	AD Egress (to CMS) Allocations
TxC_AD_OCCUPANCY	0x0a		0	AD Egress (to CMS) Occupancy
TxC_AD_CYCLES_NE	0x0b		0	AD Egress (to CMS) Not Empty
TxC_AD_CYCLES_FULL	0x0c		0	AD Egress (to CMS) Full
TxC_AD_CREDITS_ACQUIRED	0x0d		0	AD Egress (to CMS) Credit Acquired
TxC_AD_CREDIT_OCCUPANCY	0x0e		0	AD Egress (to CMS) Credits Occupancy
TxC_AD_NO_CREDIT_CYCLES	0x0f		0	Cycles with No AD Egress (to CMS) Credits
TxC_AD_NO_CREDIT_STALLED	0x10		0	Cycles Stalled with No AD Egress (to CMS) Credits
TxC_AK_INSERTS	0x11		0	AK Egress (to CMS) Allocations
TxC_AK_OCCUPANCY	0x12		0	AK Egress (to CMS) Occupancy
TxC_AK_CYCLES_NE	0x13		0	AK Egress (to CMS) Not Empty
TxC_AK_CYCLES_FULL	0x14		0	AK Egress (to CMS) Full
TxC_BL_INSERTS	0x15		0	BL Egress (to CMS) Allocations
TxC_BL_CYCLES_NE	0x17		0	BL Egress (to CMS) Not Empty
TxC_BL_CYCLES_FULL	0x18		0	BL Egress (to CMS) Full
TxC_BL_CREDITS_ACQUIRED	0x19		0	BL Egress (to CMS) Credit Acquired
TxC_BL_NO_CREDIT_CYCLES	0x1b		0	Cycles with No BL Egress (to CMS) Credits
TxC_BL_NO_CREDIT_STALLED	0x1c		0	Cycles Stalled with No BL Egress (to CMS) Credits
TxC_AK_CREDITS_ACQUIRED	0x1d		0	AK Egress (to CMS) Credit Acquired
TxC_AK_NO_CREDIT_CYCLES	0x1f		0	Cycles with No AK Egress (to CMS) Credits
TxC_AK_NO_CREDIT_STALLED	0x20		0	Cycles Stalled with No AK Egress (to CMS) Credits
BYPASS_M2M_INGRESS	0x21	0-3	0	M2M to iMC Bypass
BYPASS_M2M_EGRESS	0x22		0	M2M to iMC Bypass
DIRECT2CORE_NOT_TAKEN_DIRSTATE	0x24	0-3	0	Cycles when direct to core mode, which bypasses the CHA, was disabled
DIRECT2CORE_TXN_OVERRIDE	0x25	0-3	0	Number of reads in which direct to core transaction was overridden

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
DIRECT2UPI_NOT_TAKEN_DIRSTATE	0x27		0	Cycles when Direct2UPI was Disabled
DIRECT2UPI_NOT_TAKEN_CREDITS	0x28		0	Number of reads in which direct to Intel UPI transactions were overridden
DIRECT2UPI_TXN_OVERRIDE	0x29		0	Number of times a direct to UPI transaction was overridden.
DIRECTORY_HIT	0x2a		0	Directory Hit
DIRECTORY_MISS	0x2b		0	Directory Miss
TAG_HIT	0x2c		0	Tag Hit
DIRECTORY_LOOKUP	0x2d	0-3	0	Multi-socket cacheline Directory Lookups
SCOREBOARD_RD_ACCEPTS	0x2f		0	Scoreboard Accepts
SCOREBOARD_RD_REJECTS	0x30		0	Scoreboard Rejects
SCOREBOARD_WR_ACCEPTS	0x31		0	Scoreboard Accepts
SCOREBOARD_WR_REJECTS	0x32		0	Scoreboard Rejects
SCOREBOARD_AD_RETRY_ACCEPTS	0x33		0	
SCOREBOARD_AD_RETRY_REJECTS	0x34		0	
SCOREBOARD_BL_RETRY_ACCEPTS	0x35		0	Retry - Mem Mirroring Mode
SCOREBOARD_BL_RETRY_REJECTS	0x36		0	Retry - Mem Mirroring Mode
IMC_READS	0x37	0-3	0	M2M Reads Issued to iMC
IMC_WRITES	0x38	0-3	0	M2M Writes Issued to iMC
TxC_AK	0x39		0	Outbound Ring Transactions on AK
TxC_BL	0x40		0	Outbound DRS Ring Transactions to Cache
TGR_AD_CREDITS	0x41		0	Number AD Ingress Credits
TGR_BL_CREDITS	0x42		0	Number BL Ingress Credits
RPQ_NO_REG_CRD	0x43		0	M2M to iMC RPQ Cycles w/Credits - Regular
RPQ_NO_SPEC_CRD	0x44		0	M2M to iMC RPQ Cycles w/Credits - Special
TRACKER_FULL	0x45		0	Tracker Cycles Full
TRACKER_NE	0x46		0	Tracker Cycles Not Empty
TRACKER_OCCUPANCY	0x47		0	Tracker Occupancy
TRACKER_INSERTS	0x49		0	Tracker Inserts
WR_TRACKER_FULL	0x4a		0	Write Tracker Cycles Full
WR_TRACKER_NE	0x4b		0	Write Tracker Cycles Not Empty
PKT_MATCH	0x4c		0	Number Packet Header Matches
WPQ_NO_REG_CRD	0x4d		0	M2M->iMC WPQ Cycles w/Credits - Regular
WPQ_NO_SPEC_CRD	0x4e		0	M2M->iMC WPQ Cycles w/Credits - Special
RPQ_NO_REG_CRD_PMM	0x4f		0	M2M->iMC RPQ Cycles w/Credits - PMM
WPQ_NO_REG_CRD_PMM	0x51		0	M2M->iMC WPQ Cycles w/Credits - PMM
WR_TRACKER_OCCUPANCY	0x55		0	Write Tracker Occupancy
WR_TRACKER_INSERTS	0x56		0	Write Tracker Inserts
WPQ_FLUSH	0x58		0	WPQ Flush
RxC_AK_WR_CMP	0x5c		0	AK Egress (to CMS) Allocations
WR_TRACKER_POSTED_OCCUPANCY	0x5d		0	Write Tracker Posted Occupancy

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
WR_TRACKER_POSTED_INSERTS	0x5e		0	Write Tracker Posted Inserts
TxC_AKC_CREDITS	0x5f		0	AKC Credits
DIRECT2CORE_NOT_TAKEN_NOTFORKED	0x60	0-3	0	
TAG_MISS	0x61		0	Tag Miss
WR_TRACKER_NONPOSTED_OCCUPANCY	0x62		0	Write Tracker Non-Posted Occupancy
WR_TRACKER_NONPOSTED_INSERTS	0x63		0	Write Tracker Non-Posted Inserts
MIRR_WRQ_INSERTS	0x64		0	Write Tracker Inserts
MIRR_WRQ_OCCUPANCY	0x65		0	Write Tracker Occupancy
PREFCAM_OCCUPANCY	0x6a		0	Prefetch CAM Occupancy
PREFCAM_CYCLES_FULL	0x6b		0	Prefetch CAM Cycles Full
PREFCAM_CYCLES_NE	0x6c		0	Prefetch CAM Cycles Not Empty
PREFCAM_INSERTS	0x6d		0	Prefetch CAM Inserts
PREFCAM_DEALLOCS	0x6e		0	Prefetch CAM Deallocs
PREFCAM_DEMAND_DROPS	0x6f		0	Data Prefetches Dropped
PREFCAM_DROP_REASONS_CH0	0x70		0	Data Prefetches Dropped Ch0 - Reasons
PREFCAM_DROP_REASONS_CH1	0x71		0	Data Prefetches Dropped Ch1 - Reasons
PREFCAM_DROP_REASONS_CH2	0x72		0	Data Prefetches Dropped Ch2 - Reasons
PREFCAM_CIS_DROPS	0x73		0	
PREFCAM_DEMAND_MERGE	0x74		0	Demands Merged with CAMed Prefetches
PREFCAM_DEMAND_NO_MERGE	0x75		0	Demands Not Merged with CAMed Prefetches
PREFCAM_RESP_MISS	0x76		0	
PREFCAM_RxC_OCCUPANCY	0x77		0	
RxC_AD_PREF_OCCUPANCY	0x77		0	AD Ingress (from CMS) Occupancy - Prefetches
PREFCAM_RxC_INSERTS	0x78		0	
PREFCAM_RxC_CYCLES_NE	0x79		0	
PREFCAM_RxC_DEALLOCS	0x7a		0	
DISTRESS_PMM_MEMMODE	0xf1	0-3	0	
DISTRESS_PMM	0xf2	0-3	0	

2.8.5 M2M Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the M2M Box.

BYPASS_M2M_EGRESS

- **Title:** M2M to iMC Bypass
- **Category:** BL Egress Events
- **Event Code:** 0x22
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-229. Unit Masks for BYPASS_M2M_EGRESS

Extension	umask [15:8]	Description
TAKEN	bxxxxxxx1	Taken
NOT_TAKEN	bxxxxxx1x	Not Taken

BYPASS_M2M_INGRESS

- **Title:** M2M to iMC Bypass
- **Category:** BL Ingress Events
- **Event Code:** 0x21
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

Table 2-230. Unit Masks for BYPASS_M2M_INGRESS

Extension	umask [15:8]	Description
TAKEN	bxxxxxxx1	Taken
NOT_TAKEN	bxxxxxx1x	Not Taken

CLOCKTICKS

- **Title:** Clockticks of the mesh to memory (M2M)
- **Category:** UCLK Events
- **Event Code:** 0x00
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

DIRECT2CORE_NOT_TAKEN_DIRSTATE

- **Title:** Cycles when direct to core mode, which bypasses the CHA, was disabled
- **Category:** DIRECT2CORE Events
- **Event Code:** 0x24
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

DIRECT2CORE_NOT_TAKEN_NOTFORKED

- **Title:**
- **Category:** DIRECT2CORE Events
- **Event Code:** 0x60
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

DIRECT2CORE_TXN_OVERRIDE

- **Title:** Number of reads in which direct to core transaction was overridden
- **Category:** DIRECT2CORE Events
- **Event Code:** 0x25
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

DIRECT2UPI_NOT_TAKEN_CREDITS

- **Title:** Number of reads in which direct to Intel UPI transactions were overridden
- **Category:** DIRECT2UPI Events
- **Event Code:** 0x28
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

DIRECT2UPI_NOT_TAKEN_DIRSTATE

- **Title:** Cycles when Direct2UPI was Disabled
- **Category:** DIRECT2UPI Events
- **Event Code:** 0x27
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

DIRECT2UPI_TXN_OVERRIDE

- **Title:** Number of times a direct to UPI transaction was overridden.
- **Category:** DIRECT2UPI Events
- **Event Code:** 0x29
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

DIRECTORY_HIT

- **Title:** Directory Hit
- **Category:** Directory State Events
- **Event Code:** 0x2a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Covers NearMem Reads (Demand and Underfill).

Table 2-231. Unit Masks for DIRECTORY_HIT

Extension	umask [15:8]	Description
DIRTY_I	bxxxxxxx1	On Dirty Line in I State
DIRTY_S	bxxxxxx1x	On Dirty Line in S State
DIRTY_P	bxxxxx1xx	On Dirty Line in L State
DIRTY_A	bxxxx1xxx	On Dirty Line in A State

Table 2-231. Unit Masks for DIRECTORY_HIT

Extension	umask [15:8]	Description
CLEAN_I	bxxx1xxxx	On NonDirty Line in I State
CLEAN_S	bxx1xxxxx	On NonDirty Line in S State
CLEAN_P	bx1xxxxxx	On NonDirty Line in L State
CLEAN_A	b1xxxxxxx	On NonDirty Line in A State

DIRECTORY_LOOKUP

- **Title:** Multi-socket cacheline Directory Lookups
- **Category:** DIRECTORY Events
- **Event Code:** 0x2d
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

Table 2-232. Unit Masks for DIRECTORY_LOOKUP

Extension	umask [15:8]	Description
ANY	bxxxxxxx1	Found in any state
STATE_I	bxxxxxx1x	Found in I state
STATE_S	bxxxxx1xx	Found in S state
STATE_A	bxxxx1xxx	Found in A state

DIRECTORY_MISS

- **Title:** Directory Miss
- **Category:** Directory State Events
- **Event Code:** 0x2b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Covers NearMem Reads (Demand and Underfill).

Table 2-233. Unit Masks for DIRECTORY_MISS

Extension	umask [15:8]	Description
DIRTY_I	bxxxxxxx1	On Dirty Line in I State
DIRTY_S	bxxxxxx1x	On Dirty Line in S State
DIRTY_P	bxxxxx1xx	On Dirty Line in L State
DIRTY_A	bxxxx1xxx	On Dirty Line in A State
CLEAN_I	bxxx1xxxx	On NonDirty Line in I State
CLEAN_S	bxx1xxxxx	On NonDirty Line in S State
CLEAN_P	bx1xxxxxx	On NonDirty Line in L State
CLEAN_A	b1xxxxxxx	On NonDirty Line in A State

DISTRESS_PMM

- **Title:**
- **Category:** Distress Events
- **Event Code:** 0xf2
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

DISTRESS_PMM_MEMMODE

- **Title:**
- **Category:** Distress Events
- **Event Code:** 0xf1
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**

IMC_READS

- **Title:** M2M Reads Issued to iMC
- **Category:** IMC Events
- **Event Code:** 0x37
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**
- **NOTE:** Scrub Reads due to ECC errors not currently included

Table 2-234. Unit Masks for IMC_READS

Extension	umask [15:8]	xtra [39:32]	Description
CH1_NORMAL	bxxxxxxx1		Normal Priority - Ch1
NORMAL	bxxxxxxx1		Normal Priority - All Channels
CH0_NORMAL	bxxxxxxx1		Normal Priority - Ch0
CH1_ISOCH	bxxxxxx1x		Critical Priority - Ch1
ISOCH	bxxxxxx1x		Critical Priority - All Channels
CH0_ISOCH	bxxxxxx1x		Critical Priority - Ch0
ALL	bxxxxx1xx		All, regardless of priority. - All Channels
CH0_ALL	bxxxxx1xx		All, regardless of priority. - Ch0
CH1_ALL	bxxxxx1xx		All, regardless of priority. - Ch1
CH1_TO_DDR_AS_MEM	bxxxx1xxx		DDR - Ch1
CH0_TO_DDR_AS_MEM	bxxxx1xxx		DDR - Ch0
TO_DDR_AS_MEM	bxxxx1xxx		DDR - All Channels
CH1_TO_DDR_AS_CACHE	bxxx1xxxx		DDR, acting as Cache - Ch1
CH0_TO_DDR_AS_CACHE	bxxx1xxxx		DDR, acting as Cache - Ch0
TO_DDR_AS_CACHE	bxxx1xxxx		DDR, acting as Cache - All Channels
CH0_TO_PMM	bxx1xxxxx		PMM - Ch0 Counts all PMM dimm read requests(full line) sent from M2M to iMC

Table 2-234. Unit Masks for IMC_READS

Extension	umask [15:8]	xtra [39:32]	Description
CH1_TO_PMM	bxx1xxxxx		PMM - Ch1 Counts all PMM dimm read requests(full line) sent from M2M to iMC
TO_PMM	bxx1xxxxx		PMM - All Channels
FROM_TGR	bx1xxxxxx		From TGR - All Channels
CH1_FROM_TGR	bx1xxxxxx		From TGR - Ch1
CH2_FROM_TGR	bx1xxxxxx		From TGR - Ch2
CH0_FROM_TGR	bx1xxxxxx		From TGR - Ch0

IMC_WRITES

- **Title:** M2M Writes Issued to iMC
- **Category:** IMC Events
- **Event Code:** 0x38
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-3

- **Definition:**
- **NOTE:** Scrub Writes due to ECC errors not currently included

Table 2-235. Unit Masks for IMC_WRITES

Extension	umask [15:8]	xtra [39:32]	Description
CH0_NI	bxxxxxxxx		Non-Inclusive - Ch0
CH1_FROM_TGR	bxxxxxxxx		From TGR - Ch1
FROM_TGR	bxxxxxxxx		From TGR - All Channels
CH0_NI_MISS	bxxxxxxxx		Non-Inclusive Miss - Ch0
NI	bxxxxxxxx		Non-Inclusive - All Channels
CH1_NI_MISS	bxxxxxxxx		Non-Inclusive Miss - Ch1
CH0_FROM_TGR	bxxxxxxxx		From TGR - Ch0
NI_MISS	bxxxxxxxx		Non-Inclusive Miss - All Channels
CH1_NI	bxxxxxxxx		Non-Inclusive - Ch1
CH1_FULL	bxxxxxxx1		Full Line Non-ISOCH - Ch1
CH0_FULL	bxxxxxxx1		Full Line Non-ISOCH - Ch0
FULL	bxxxxxxx1		Full Non-ISOCH - All Channels
CH1_PARTIAL	bxxxxxx1x		Partial Non-ISOCH - Ch1
PARTIAL	bxxxxxx1x		Partial Non-ISOCH - All Channels
CH0_PARTIAL	bxxxxxx1x		Partial Non-ISOCH - Ch0
CH0_FULL_ISOCH	bxxxxx1xx		ISOCH Full Line - Ch0
FULL_ISOCH	bxxxxx1xx		ISOCH Full Line - All Channels
CH1_FULL_ISOCH	bxxxxx1xx		ISOCH Full Line - Ch1
PARTIAL_ISOCH	bxxxx1xxx		ISOCH Partial - All Channels
CH0_PARTIAL_ISOCH	bxxxx1xxx		ISOCH Partial - Ch0
CH1_PARTIAL_ISOCH	bxxxx1xxx		ISOCH Partial - Ch1
CH0_ALL	bxxx1xxxx		All Writes - Ch0

Table 2-235. Unit Masks for IMC_WRITES

Extension	umask [15:8]	xtra [39:32]	Description
ALL	bxxx1xxxx		All Writes - All Channels
CH1_ALL	bxxx1xxxx		All Writes - Ch1
CH1_TO_DDR_AS_MEM	bxx1xxxxx		DDR - Ch1
CH0_TO_DDR_AS_MEM	bxx1xxxxx		DDR - Ch0
TO_DDR_AS_MEM	bxx1xxxxx		DDR - All Channels
CH1_TO_DDR_AS_CACHE	bx1xxxxxx		DDR, acting as Cache - Ch1
CH0_TO_DDR_AS_CACHE	bx1xxxxxx		DDR, acting as Cache - Ch0
TO_DDR_AS_CACHE	bx1xxxxxx		DDR, acting as Cache - All Channels
CH0_TO_PMM	b1xxxxxxx		PMM - Ch0 Counts all PMM dimm writes requests(full line and partial) sent from M2M to iMC
TO_PMM	b1xxxxxxx		PMM - All Channels
CH1_TO_PMM	b1xxxxxxx		PMM - Ch1 Counts all PMM dimm writes requests(full line and partial) sent from M2M to iMC

MIRR_WRQ_INSERTS

- **Title:** Write Tracker Inserts
- **Category:** Mirror WriteQ EVENTS
- **Event Code:** 0x64
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

MIRR_WRQ_OCCUPANCY

- **Title:** Write Tracker Occupancy
- **Category:** Mirror WriteQ EVENTS
- **Event Code:** 0x65
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PKT_MATCH

- **Title:** Number Packet Header Matches
- **Category:** PACKET MATCH Events
- **Event Code:** 0x4c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-236. Unit Masks for PKT_MATCH

Extension	umask [15:8]	Description
MESH	bxxxxxxx1	Mesh Match
MC	bxxxxxx1x	MC Match

PREFCAM_CIS_DROPS

- **Title:**
- **Category:** Prefetch CAM Events
- **Event Code:** 0x73
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PREFCAM_CYCLES_FULL

- **Title:** Prefetch CAM Cycles Full
- **Category:** Prefetch CAM Events
- **Event Code:** 0x6b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-237. Unit Masks for PREFCAM_CYCLES_FULL

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
ALLCH	bxxxxx111	All Channels

PREFCAM_CYCLES_NE

- **Title:** Prefetch CAM Cycles Not Empty
- **Category:** Prefetch CAM Events
- **Event Code:** 0x6c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-238. Unit Masks for PREFCAM_CYCLES_NE

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
ALLCH	bxxxxx111	All Channels

PREFCAM_DEALLOCS

- **Title:** Prefetch CAM Deallocs
- **Category:** Prefetch CAM Events
- **Event Code:** 0x6e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-239. Unit Masks for PREFCAM_DEALLOCS

Extension	umask [15:8]	xtra [39:32]	Description
CH2_RSP_PDRESET	bxxxxxxxx		
CH2_HITA1_INVAL	bxxxxxxxx		
CH2_HITA0_INVAL	bxxxxxxxx		
CH2_MISS_INVAL	bxxxxxxxx		
CH0_HITA0_INVAL	bxxxxxxx1		
CH0_HITA1_INVAL	bxxxxxx1x		
CH0_MISS_INVAL	bxxxxx1xx		
CH0_RSP_PDRESET	bxxxx1xxx		
CH1_HITA0_INVAL	bxxx1xxxx		
CH1_HITA1_INVAL	bxx1xxxxx		
CH1_MISS_INVAL	bx1xxxxxx		
CH1_RSP_PDRESET	b1xxxxxxx		

PREFCAM_DEMAND_DROPS

- **Title:** Data Prefetches Dropped
- **Category:** Prefetch CAM Events
- **Event Code:** 0x6f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-240. Unit Masks for PREFCAM_DEMAND_DROPS

Extension	umask [15:8]	Description
CH0_XPT	bxxxxxxx1	XPT - Ch 0
CH0_UPI	bxxxxxx1x	UPI - Ch 0
CH1_XPT	bxxxxx1xx	XPT - Ch 1
CH1_UPI	bxxxx1xxx	UPI - Ch 1
CH2_XPT	bxxx1xxxx	XPT - Ch 2
XPT_ALLCH	b000010101	XPT - All Channels
CH2_UPI	bxx1xxxxx	UPI - Ch 2
UPI_ALLCH	b000101010	UPI - All Channels

PREFCAM_DEMAND_MERGE

- **Title:** Demands Merged with CAMed Prefetches
- **Category:** Prefetch CAM Events
- **Event Code:** 0x74
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-241. Unit Masks for PREFCAM_DEMAND_MERGE

Extension	umask [15:8]	Description
CH0_XPTUPI	bxxxxxxx1	XPT - Ch 0
CH1_XPTUPI	bxxxxx1xx	XPT - Ch 1
CH2_XPTUPI	bxxx1xxxx	XPT - Ch 2
XPTUPI_ALLCH	b000010101	XPT - All Channels

PREFCAM_DEMAND_NO_MERGE

- **Title:** Demands Not Merged with CAMed Prefetches
- **Category:** Prefetch CAM Events
- **Event Code:** 0x75
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-242. Unit Masks for PREFCAM_DEMAND_NO_MERGE

Extension	umask [15:8]	Description
CH0_XPTUPI	bxxxxxxx1	XPT - Ch 0
CH1_XPTUPI	bxxxxx1xx	XPT - Ch 1
CH2_XPTUPI	bxxx1xxxx	XPT - Ch 2
XPTUPI_ALLCH	b000010101	XPT - All Channels

PREFCAM_DROP_REASONS_CH0

- **Title:** Data Prefetches Dropped Ch0 - Reasons
- **Category:** Prefetch CAM Events
- **Event Code:** 0x70
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-243. Unit Masks for PREFCAM_DROP_REASONS_CH0

Extension	umask [15:8]	xtra [39:32]	Description
XPT_THRESH	bxxxxxxxx		
RPQ_PROXY	bxxxxxxxx		
UPI_THRESH	bxxxxxxxx		
PF_SECURE_DROP	bxxxxxxx1		
NOT_PF_SAD_REGION	bxxxxxx1x		
PF_CAM_HIT	bxxxxx1xx		
STOP_B2B	bxxxx1xxx		
ERRORBLK_RxC	bxxx1xxxx		
PF_AD_CRD	bxx1xxxxx		

Table 2-243. Unit Masks for PREFCAM_DROP_REASONS_CH0

Extension	umask [15:8]	xtra [39:32]	Description
PF_CAM_FULL	bx1xxxxxx		
WPQ_PROXY	b1xxxxxxx		

PREFCAM_DROP_REASONS_CH1

- **Title:** Data Prefetches Dropped Ch1 - Reasons
- **Category:** Prefetch CAM Events
- **Event Code:** 0x71
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-244. Unit Masks for PREFCAM_DROP_REASONS_CH1

Extension	umask [15:8]	xtra [39:32]	Description
RPQ_PROXY	bxxxxxxxx		
UPI_THRESH	bxxxxxxxx		
XPT_THRESH	bxxxxxxxx		
PF_SECURE_DROP	bxxxxxxx1		
NOT_PF_SAD_REGION	bxxxxxx1x		
PF_CAM_HIT	bxxxxx1xx		
STOP_B2B	bxxxx1xxx		
ERRORBLK_RxC	bxxx1xxxx		
PF_AD_CRD	bxx1xxxxx		
PF_CAM_FULL	bx1xxxxxx		
WPQ_PROXY	b1xxxxxxx		

PREFCAM_DROP_REASONS_CH2

- **Title:** Data Prefetches Dropped Ch2 - Reasons
- **Category:** Prefetch CAM Events
- **Event Code:** 0x72
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-245. Unit Masks for PREFCAM_DROP_REASONS_CH2

Extension	umask [15:8]	xtra [39:32]	Description
RPQ_PROXY	bxxxxxxxx		
UPI_THRESH	bxxxxxxxx		
XPT_THRESH	bxxxxxxxx		
PF_SECURE_DROP	bxxxxxxx1		
NOT_PF_SAD_REGION	bxxxxxx1x		
PF_CAM_HIT	bxxxxx1xx		

Table 2-245. Unit Masks for PREFCAM_DROP_REASONS_CH2

Extension	umask [15:8]	xtra [39:32]	Description
STOP_B2B	bxxxx1xxx		
ERRORBLK_RxC	bxxx1xxxx		
PF_AD_CRD	bxx1xxxxx		
PF_CAM_FULL	bx1xxxxxx		
WPQ_PROXY	b1xxxxxxx		

PREFCAM_INSERTS

- **Title:** Prefetch CAM Inserts
- **Category:** Prefetch CAM Events
- **Event Code:** 0x6d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-246. Unit Masks for PREFCAM_INSERTS

Extension	umask [15:8]	Description
CH0_XPT	bxxxxxxx1	XPT - Ch 0
CH0_UPI	bxxxxxx1x	UPI - Ch 0
CH1_XPT	bxxxxx1xx	XPT - Ch 1
CH1_UPI	bxxxx1xxx	UPI - Ch 1
CH2_XPT	bxxx1xxxx	XPT - Ch 2
XPT_ALLCH	b000010101	XPT - All Channels
CH2_UPI	bxx1xxxxx	UPI - Ch 2
UPI_ALLCH	b000101010	UPI - All Channels

PREFCAM_OCCUPANCY

- **Title:** Prefetch CAM Occupancy
- **Category:** Prefetch CAM Events
- **Event Code:** 0x6a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-247. Unit Masks for PREFCAM_OCCUPANCY

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
ALLCH	bxxxxx111	All Channels

PREFCAM_RESP_MISS

- **Title:**
- **Category:** Prefetch CAM Events
- **Event Code:** 0x76
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-248. Unit Masks for PREFCAM_RESP_MISS

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
ALLCH	bxxxxx111	All Channels

PREFCAM_RxC_CYCLES_NE

- **Title:**
- **Category:** Prefetch CAM Events
- **Event Code:** 0x79
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PREFCAM_RxC_DEALLOCS

- **Title:**
- **Category:** Prefetch CAM Events
- **Event Code:** 0x7a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-249. Unit Masks for PREFCAM_RxC_DEALLOCS

Extension	umask [15:8]	Description
SQUASHED	bxxxxxxx1	
1LM_POSTED	bxxxxxx1x	
PMM_MEMMODE_ACCEPT	bxxxxx1xx	
CIS	bxxxx1xxx	

PREFCAM_RxC_INSERTS

- **Title:**
- **Category:** Prefetch CAM Events
- **Event Code:** 0x78
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

PREFCAM_RxC_OCCUPANCY

- **Title:**
- **Category:** Prefetch CAM Events
- **Event Code:** 0x77
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RPQ_NO_REG_CRD

- **Title:** M2M to iMC RPQ Cycles w/Credits - Regular
- **Category:** RPQ CREDIT Events
- **Event Code:** 0x43
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** To Count # cycles w/o credits, either set .thresh to 1 and .invert to 1 OR subtract this from total cycles

Table 2-250. Unit Masks for RPQ_NO_REG_CRD

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

RPQ_NO_REG_CRD_PMM

- **Title:** M2M->iMC RPQ Cycles w/Credits - PMM
- **Category:** RPQ CREDIT Events
- **Event Code:** 0x4f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** To Count # cycles w/o credits, either set .thresh to 1 and .invert to 1 OR subtract this from total cycles

Table 2-251. Unit Masks for RPQ_NO_REG_CRD_PMM

Extension	umask [15:8]	Description
CHN0	bxxxxxxx1	Channel 0
CHN1	bxxxxxx1x	Channel 1
CHN2	bxxxxx1xx	Channel 2

RPQ_NO_SPEC_CRD

- **Title:** M2M to iMC RPQ Cycles w/Credits - Special
- **Category:** RPQ CREDIT Events
- **Event Code:** 0x44
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** To Count # cycles w/o credits, either set .thresh to 1 and .invert to 1 OR subtract this from total cycles

Table 2-252. Unit Masks for RPQ_NO_SPEC_CRD

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

RxC_AD_CYCLES_FULL

- **Title:** AD Ingress (from CMS) Full
- **Category:** AD Ingress Events
- **Event Code:** 0x04
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_AD_CYCLES_NE

- **Title:** AD Ingress (from CMS) Not Empty
- **Category:** AD Ingress Events
- **Event Code:** 0x03
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_AD_INSERTS

- **Title:** AD Ingress (from CMS) Allocations
- **Category:** AD Ingress Events
- **Event Code:** 0x01
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_AD_OCCUPANCY

- **Title:** AD Ingress (from CMS) Occupancy
- **Category:** AD Ingress Events
- **Event Code:** 0x02
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_AD_PREF_OCCUPANCY

- **Title:** AD Ingress (from CMS) Occupancy - Prefetches
- **Category:** AD Ingress Events
- **Event Code:** 0x77
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_AK_WR_CMP

- **Title:** AK Egress (to CMS) Allocations
- **Category:** AK Egress Events
- **Event Code:** 0x5c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_BL_CYCLES_FULL

- **Title:** BL Ingress (from CMS) Full
- **Category:** BL Ingress Events
- **Event Code:** 0x08
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_BL_CYCLES_NE

- **Title:** BL Ingress (from CMS) Not Empty
- **Category:** BL Ingress Events
- **Event Code:** 0x07
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_BL_INSERTS

- **Title:** BL Ingress (from CMS) Allocations
- **Category:** BL Ingress Events
- **Event Code:** 0x05
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

RxC_BL_OCCUPANCY

- **Title:** BL Ingress (from CMS) Occupancy
- **Category:** BL Ingress Events
- **Event Code:** 0x06
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_AD_RETRY_ACCEPTS

- **Title:**
- **Category:** Scoreboard Events
- **Event Code:** 0x33
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_AD_RETRY_REJECTS

- **Title:**
- **Category:** Scoreboard Events
- **Event Code:** 0x34
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_BL_RETRY_ACCEPTS

- **Title:** Retry - Mem Mirroring Mode
- **Category:** Scoreboard Events
- **Event Code:** 0x35
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_BL_RETRY_REJECTS

- **Title:** Retry - Mem Mirroring Mode
- **Category:** Scoreboard Events
- **Event Code:** 0x36
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_RD_ACCEPTS

- **Title:** Scoreboard Accepts
- **Category:** Scoreboard Events
- **Event Code:** 0x2f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_RD_REJECTS

- **Title:** Scoreboard Rejects
- **Category:** Scoreboard Events
- **Event Code:** 0x30
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_WR_ACCEPTS

- **Title:** Scoreboard Accepts
- **Category:** Scoreboard Events
- **Event Code:** 0x31
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

SCOREBOARD_WR_REJECTS

- **Title:** Scoreboard Rejects
- **Category:** Scoreboard Events
- **Event Code:** 0x32
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

TAG_HIT

- **Title:** Tag Hit
- **Category:** Directory State Events
- **Event Code:** 0x2c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Tag Hit indicates when a request sent to the iMC hit in Near Memory.
- **NOTE:** When the iMC returns data, the response's tag is compared against request tag, A hit indicates to M2M the data was found in NearMem.

Table 2-253. Unit Masks for TAG_HIT

Extension	umask [15:8]	Description
NM_RD_HIT_CLEAN	bxxxxxxx1	Clean NearMem Read Hit Counts clean full line read hits (reads and RFOs).
NM_RD_HIT_DIRTY	bxxxxxx1x	Dirty NearMem Read Hit Counts dirty full line read hits (reads and RFOs).
NM_UFILL_HIT_CLEAN	bxxxxx1xx	Clean NearMem Underfill Hit Counts clean underfill hits due to a partial write
NM_UFILL_HIT_DIRTY	bxxxx1xxx	Dirty NearMem Underfill Hit Counts dirty underfill read hits due to a partial write

TAG_MISS

- **Title:** Tag Miss
- **Category:** Directory State Events
- **Event Code:** 0x61
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

TGR_AD_CREDITS

- **Title:** Number AD Ingress Credits
- **Category:** Transgress Credit Events
- **Event Code:** 0x41
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

TGR_BL_CREDITS

- **Title:** Number BL Ingress Credits
- **Category:** Transgress Credit Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

TRACKER_FULL

- **Title:** Tracker Cycles Full
- **Category:** TRACKER Events
- **Event Code:** 0x45
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-254. Unit Masks for TRACKER_FULL

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

TRACKER_INSERTS

- **Title:** Tracker Inserts
- **Category:** TRACKER Events
- **Event Code:** 0x49
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-255. Unit Masks for TRACKER_INSERTS

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

TRACKER_NE

- **Title:** Tracker Cycles Not Empty
- **Category:** TRACKER Events
- **Event Code:** 0x46
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-256. Unit Masks for TRACKER_NE

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

TRACKER_OCCUPANCY

- **Title:** Tracker Occupancy
- **Category:** TRACKER Events
- **Event Code:** 0x47
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Tie to Packet Mask/Match?

Table 2-257. Unit Masks for TRACKER_OCCUPANCY

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

TxC_AD_CREDITS_ACQUIRED

- **Title:** AD Egress (to CMS) Credit Acquired
- **Category:** AD CMS/Mesh Egress Credit Events
- **Event Code:** 0x0d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_CREDIT_OCCUPANCY

- **Title:** AD Egress (to CMS) Credits Occupancy
- **Category:** AD CMS/Mesh Egress Credit Events
- **Event Code:** 0x0e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_CYCLES_FULL

- **Title:** AD Egress (to CMS) Full
- **Category:** AD Egress Events
- **Event Code:** 0x0c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_CYCLES_NE

- **Title:** AD Egress (to CMS) Not Empty
- **Category:** AD Egress Events
- **Event Code:** 0x0b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_INSERTS

- **Title:** AD Egress (to CMS) Allocations
- **Category:** AD Egress Events
- **Event Code:** 0x09
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_NO_CREDIT_CYCLES

- **Title:** Cycles with No AD Egress (to CMS) Credits
- **Category:** AD CMS/Mesh Egress Credit Events
- **Event Code:** 0x0f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_NO_CREDIT_STALLED

- **Title:** Cycles Stalled with No AD Egress (to CMS) Credits
- **Category:** AD CMS/Mesh Egress Credit Events
- **Event Code:** 0x10
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AD_OCCUPANCY

- **Title:** AD Egress (to CMS) Occupancy
- **Category:** AD Egress Events
- **Event Code:** 0x0a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Not a per channel event?

TxC_AK

- **Title:** Outbound Ring Transactions on AK
- **Category:** OUTBOUND_TX Events
- **Event Code:** 0x39
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-258. Unit Masks for TxC_AK

Extension	umask [15:8]	Description
NDR	bxxxxxxx1	NDR Transactions
CRD_CBO	bxxxxxx1x	CRD Transactions to Cbo

TxC_AKC_CREDITS

- **Title:** AKC Credits
- **Category:** AK Egress Events
- **Event Code:** 0x5f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

TxC_AK_CREDITS_ACQUIRED

- **Title:** AK Egress (to CMS) Credit Acquired
- **Category:** AK CMS/Mesh Egress Credit Events
- **Event Code:** 0x1d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-259. Unit Masks for TxC_AK_CREDITS_ACQUIRED

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side

TxC_AK_CYCLES_FULL

- **Title:** AK Egress (to CMS) Full
- **Category:** AK Egress Events
- **Event Code:** 0x14
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** Some extra wild guesses as to what the subevents count - and why does it seem to track 0 credits for each CMS agent, but the other related events don't?

Table 2-260. Unit Masks for TxC_AK_CYCLES_FULL

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All
RDCRD0	b0xxx1xxx	
WRCRD0	b0xx1xxxx	
WRCMP0	b0x1xxxxx	
RDCRD1	b1xxx1xxx	
WRCRD1	b1xx1xxxx	
WRCMP1	b1x1xxxxx	

TxC_AK_CYCLES_NE

- **Title:** AK Egress (to CMS) Not Empty
- **Category:** AK Egress Events
- **Event Code:** 0x13
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-261. Unit Masks for TxC_AK_CYCLES_NE

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All
RDCRD	bxxxx1xxx	
WRCRD	bxxx1xxxx	
WRCMP	bxx1xxxxx	

TxC_AK_INSERTS

- **Title:** AK Egress (to CMS) Allocations
- **Category:** AK Egress Events
- **Event Code:** 0x11
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-262. Unit Masks for TxC_AK_INSERTS

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All
RDCRD	bxxxx1xxx	
WRCRD	bxxx1xxxx	
WRCMP	bxx1xxxxx	
PREF_RD_CAM_HIT	bx1xxxxxx	

TxC_AK_NO_CREDIT_CYCLES

- **Title:** Cycles with No AK Egress (to CMS) Credits
- **Category:** AK CMS/Mesh Egress Credit Events
- **Event Code:** 0x1f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-263. Unit Masks for TxC_AK_NO_CREDIT_CYCLES

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side

TxC_AK_NO_CREDIT_STALLED

- **Title:** Cycles Stalled with No AK Egress (to CMS) Credits
- **Category:** AK CMS/Mesh Egress Credit Events
- **Event Code:** 0x20
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-264. Unit Masks for TxC_AK_NO_CREDIT_STALLED

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side

TxC_AK_OCCUPANCY

- **Title:** AK Egress (to CMS) Occupancy
- **Category:** AK Egress Events
- **Event Code:** 0x12
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-265. Unit Masks for TxC_AK_OCCUPANCY

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All
RDCRD	bxxxx1xxx	
WRCRD	bxxx1xxxx	
WRCMP	bxx1xxxxx	

TxC_BL

- **Title:** Outbound DRS Ring Transactions to Cache
- **Category:** OUTBOUND_TX Events
- **Event Code:** 0x40
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-266. Unit Masks for TxC_BL

Extension	umask [15:8]	Description
DRS_CACHE	bxxxxxxx1	Data to Cache
DRS_CORE	bxxxxxx1x	Data to Core
DRS_UPI	bxxxxx1xx	Data to QPI

TxC_BL_CREDITS_ACQUIRED

- **Title:** BL Egress (to CMS) Credit Acquired
- **Category:** BL CMS/Mesh Egress Credit Events
- **Event Code:** 0x19
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-267. Unit Masks for TxC_BL_CREDITS_ACQUIRED

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side

TxC_BL_CYCLES_FULL

- **Title:** BL Egress (to CMS) Full
- **Category:** BL Egress Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-268. Unit Masks for TxC_BL_CYCLES_FULL

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All

TxC_BL_CYCLES_NE

- **Title:** BL Egress (to CMS) Not Empty
- **Category:** BL Egress Events
- **Event Code:** 0x17
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-269. Unit Masks for TxC_BL_CYCLES_NE

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All

TxC_BL_INSERTS

- **Title:** BL Egress (to CMS) Allocations
- **Category:** BL Egress Events
- **Event Code:** 0x15
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-270. Unit Masks for TxC_BL_INSERTS

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side
ALL	bxxxxxx11	All

TxC_BL_NO_CREDIT_CYCLES

- **Title:** Cycles with No BL Egress (to CMS) Credits
- **Category:** BL CMS/Mesh Egress Credit Events
- **Event Code:** 0x1b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-271. Unit Masks for TxC_BL_NO_CREDIT_CYCLES

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side

TxC_BL_NO_CREDIT_STALLED

- **Title:** Cycles Stalled with No BL Egress (to CMS) Credits
- **Category:** BL CMS/Mesh Egress Credit Events
- **Event Code:** 0x1c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** only on g_chnl==0?

Table 2-272. Unit Masks for TxC_BL_NO_CREDIT_STALLED

Extension	umask [15:8]	Description
CMS0	bxxxxxxx1	Common Mesh Stop - Near Side
CMS1	bxxxxxx1x	Common Mesh Stop - Far Side

WPQ_FLUSH

- **Title:** WPQ Flush
- **Category:** WPQ EVENTS
- **Event Code:** 0x58
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-273. Unit Masks for WPQ_FLUSH

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

WPQ_NO_REG_CRD

- **Title:** M2M->iMC WPQ Cycles w/Credits - Regular
- **Category:** WPQ CREDIT Events
- **Event Code:** 0x4d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** To Count # cycles w/o credits, either set .thresh to 1 and .invert to 1 OR subtract this from total cycles

Table 2-274. Unit Masks for WPQ_NO_REG_CRD

Extension	umask [15:8]	Description
CHN0	bxxxxxxx1	Channel 0
CHN1	bxxxxxx1x	Channel 1
CHN2	bxxxxx1xx	Channel 2

WPQ_NO_REG_CRD_PMM

- **Title:** M2M->iMC WPQ Cycles w/Credits - PMM
- **Category:** WPQ CREDIT Events
- **Event Code:** 0x51
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** To Count # cycles w/o credits, either set .thresh to 1 and .invert to 1 OR subtract this from total cycles

Table 2-275. Unit Masks for WPQ_NO_REG_CRD_PMM

Extension	umask [15:8]	Description
CHN0	bxxxxxxx1	Channel 0
CHN1	bxxxxxx1x	Channel 1
CHN2	bxxxxx1xx	Channel 2

WPQ_NO_SPEC_CRD

- **Title:** M2M->iMC WPQ Cycles w/Credits - Special
- **Category:** WPQ CREDIT Events
- **Event Code:** 0x4e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**
- **NOTE:** To Count # cycles w/o credits, either set .thresh to 1 and .invert to 1 OR subtract this from total cycles

Table 2-276. Unit Masks for WPQ_NO_SPEC_CRD

Extension	umask [15:8]	Description
CHN0	bxxxxxxx1	Channel 0
CHN1	bxxxxxx1x	Channel 1
CHN2	bxxxxx1xx	Channel 2

WR_TRACKER_FULL

- **Title:** Write Tracker Cycles Full
- **Category:** Write Tracker Events
- **Event Code:** 0x4a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-277. Unit Masks for WR_TRACKER_FULL

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
MIRR	bxxxx1xxx	Mirror

WR_TRACKER_INSERTS

- **Title:** Write Tracker Inserts
- **Category:** Write Tracker Events
- **Event Code:** 0x56
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-278. Unit Masks for WR_TRACKER_INSERTS

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

WR_TRACKER_NE

- **Title:** Write Tracker Cycles Not Empty
- **Category:** Write Tracker Events
- **Event Code:** 0x4b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-279. Unit Masks for WR_TRACKER_NE

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
MIRR	bxxxx1xxx	Mirror
MIRR_NONTGR	bxxx1xxxx	
MIRR_PWR	bxx1xxxxx	

WR_TRACKER_NONPOSTED_INSERTS

- **Title:** Write Tracker Non-Posted Inserts
- **Category:** Write Tracker Events
- **Event Code:** 0x63
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-280. Unit Masks for WR_TRACKER_NONPOSTED_INSERTS

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

WR_TRACKER_NONPOSTED_OCCUPANCY

- **Title:** Write Tracker Non-Posted Occupancy
- **Category:** Write Tracker Events
- **Event Code:** 0x62
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-281. Unit Masks for WR_TRACKER_NONPOSTED_OCCUPANCY

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0

Table 2-281. Unit Masks for WR_TRACKER_NONPOSTED_OCCUPANCY

Extension	umask [15:8]	Description
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

WR_TRACKER_OCCUPANCY

- **Title:** Write Tracker Occupancy
- **Category:** Write Tracker Events
- **Event Code:** 0x55
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-282. Unit Masks for WR_TRACKER_OCCUPANCY

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2
MIRR	bxxxx1xxx	Mirror
MIRR_NONTGR	bxxx1xxxx	
MIRR_PWR	bxx1xxxxx	

WR_TRACKER_POSTED_INSERTS

- **Title:** Write Tracker Posted Inserts
- **Category:** Write Tracker Events
- **Event Code:** 0x5e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-283. Unit Masks for WR_TRACKER_POSTED_INSERTS

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

WR_TRACKER_POSTED_OCCUPANCY

- **Title:** Write Tracker Posted Occupancy
- **Category:** Write Tracker Events
- **Event Code:** 0x5d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-284. Unit Masks for WR_TRACKER_POSTED_OCCUPANCY

Extension	umask [15:8]	Description
CH0	bxxxxxxx1	Channel 0
CH1	bxxxxxx1x	Channel 1
CH2	bxxxxx1xx	Channel 2

2.9 M2PCIe Performance Monitoring

M2PCIe blocks manage the interface between the Mesh and each IIO stack.

2.9.1 M2PCIe Performance Monitoring Overview

Each M2PCIe Box supports event monitoring through four 48b wide counters (M2n_PCI_PMON_CTR/CTL{3:0}). Each of these four counters can be programmed to count almost any M2PCIe event (see NOTE for exceptions). the M2PCIe counters can increment by a maximum of 5b per cycle.

Note: Only counter 0 can be used for tracking occupancy events. .

2.9.2 M2PCIe Performance Monitoring Events

M2PCIe provides events to track information related to all the traffic passing through it's boundaries.

- IIO credit tracking - credits rejected, acquired and used all broken down by message Class.
  Mesh Section 2.2, "Mesh Performance Monitoring."

2.9.3 M2PCIe Box Events Ordered By Code

The following table summarizes the directly measured M2PCIe Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
CLOCKTICKS	0x01	0-3	1	Clockticks of the mesh to PCI (M2P)
RxC_CYCLES_NE	0x10	0-3	1	Ingress (from CMS) Queue Cycles Not Empty
RxC_INSERTS	0x11	0-3	1	Ingress (from CMS) Queue Inserts
P2P_CRD_OCCUPANCY	0x14	0-1	127	P2P Credit Occupancy
P2P_SHAR_RECEIVED	0x15	0-3	1	Shared Credits Received
P2P_DED_RECEIVED	0x16	0-3	1	Dedicated Credits Received
LOCAL_P2P_SHAR_RETURNED	0x17	0-3	3	Local P2P Shared Credits Returned
REMOTE_P2P_SHAR_RETURNED	0x18	0-3	3	Remote P2P Shared Credits Returned
LOCAL_P2P_DED_RETURNED_0	0x19	0-3	1	Local P2P Dedicated Credits Returned - 0
LOCAL_P2P_DED_RETURNED_1	0x1a	0-3	1	Local P2P Dedicated Credits Returned - 1
REMOTE_P2P_DED_RETURNED	0x1b	0-3	1	Remote P2P Dedicated Credits Returned
TxC_CYCLES_NE	0x23	0-1	1	Egress (to CMS) Cycles Not Empty
TxC_INSERTS	0x24	0-3	1	Egress (to CMS) Ingress
TxC_CYCLES_FULL	0x25	0-3	1	Egress (to CMS) Cycles Full
TxC_CREDITS	0x2d	0-1	4	
IIO_CREDITS_USED	0x32	0-3	1	M2PCIe IIO Credits in Use
IIO_CREDITS_ACQUIRED	0x33	0-3	1	M2PCIe IIO Credit Acquired
IIO_CREDITS_REJECT	0x34	0-3	1	M2PCIe IIO Failed to Acquire a Credit
LOCAL_SHAR_P2P_CRD_TAKEN_0	0x40	0-3	1	Local Shared P2P Credit Taken - 0
LOCAL_SHAR_P2P_CRD_TAKEN_1	0x41	0-3	1	Local Shared P2P Credit Taken - 1
REMOTE_SHAR_P2P_CRD_TAKEN_0	0x42	0-3	1	Remote Shared P2P Credit Taken - 0
REMOTE_SHAR_P2P_CRD_TAKEN_1	0x43	0-3	1	Remote Shared P2P Credit Taken - 1
LOCAL_SHAR_P2P_CRD_RETURNED	0x44	0-3	1	Local Shared P2P Credit Returned to credit ring
REMOTE_SHAR_P2P_CRD_RETURNED	0x45	0-3	3	Remote Shared P2P Credit Returned to credit ring
LOCAL_DED_P2P_CRD_TAKEN_0	0x46	0-3	3	Local Dedicated P2P Credit Taken - 0
LOCAL_DED_P2P_CRD_TAKEN_1	0x47	0-3	1	Local Dedicated P2P Credit Taken - 1
REMOTE_DED_P2P_CRD_TAKEN_0	0x48	0-3	1	Remote Dedicated P2P Credit Taken - 0
REMOTE_DED_P2P_CRD_TAKEN_1	0x49	0-3	1	Remote Dedicated P2P Credit Taken - 1
LOCAL_SHAR_P2P_CRD_WAIT_0	0x4a	0-3	1	Waiting on Local Shared P2P Credit - 0
LOCAL_SHAR_P2P_CRD_WAIT_1	0x4b	0-3	1	Waiting on Local Shared P2P Credit - 1
REMOTE_SHAR_P2P_CRD_WAIT_0	0x4c	0-3	1	Waiting on Remote Shared P2P Credit - 0
REMOTE_SHAR_P2P_CRD_WAIT_1	0x4d	0-3	1	Waiting on Remote Shared P2P Credit - 1

2.9.4 M2PCIe Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the M2PCIe Box.

CLOCKTICKS

- **Title:** Clockticks of the mesh to PCI (M2P)
- **Category:** UCLK Events
- **Event Code:** 0x01
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of uclks in the M3 uclk domain. This could be slightly different than the count in the Ubox because of enable/freeze delays. However, because the M3 is close to the Ubox, they generally should not diverge by more than a handful of cycles.

IIO_CREDITS_ACQUIRED

- **Title:** M2PCIe IIO Credit Acquired
- **Category:** IIO_CREDITS Events
- **Event Code:** 0x33
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of credits that are acquired in the M2PCIe agent for sending transactions into the IIO on either NCB or NCS are in use. Transactions from the BL ring going into the IIO Agent must first acquire a credit. These credits are for either the NCB or NCS message classes. NCB, or non-coherent bypass messages are used to transmit data without coherency (and are common). NCS is used for reads to PCIe (and should be used sparingly).

Table 2-285. Unit Masks for IIO_CREDITS_ACQUIRED

Extension	umask [15:8]	Description
DRS_0	bxxxxxxx1	DRS Credits for transfer through CMS Port 0 to the IIO for the DRS message class.
DRS_1	bxxxxxx1x	DRS Credits for transfer through CMS Port 0 to the IIO for the DRS message class.
NCB_0	bxxxxx1xx	NCB Credits for transfer through CMS Port 0 to the IIO for the NCB message class.
NCB_1	bxxxx1xxx	NCB Credits for transfer through CMS Port 0 to the IIO for the NCB message class.
NCS_0	bxxx1xxxx	NCS Credits for transfer through CMS Port 0 to the IIO for the NCS message class.
NCS_1	bxx1xxxxx	NCS Credit for transfer through CMS Port 0s to the IIO for the NCS message class.

IIO_CREDITS_REJECT

- **Title:** M2PCIe IIO Failed to Acquire a Credit
- **Category:** IIO_CREDITS Events
- **Event Code:** 0x34
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of times that a request pending in the BL Ingress attempted to acquire either a NCB or NCS credit to transmit into the IIO, but was rejected because no credits were available. NCB, or non-coherent bypass messages

are used to transmit data without coherency (and are common). NCS is used for reads to PCIe (and should be used sparingly).

Table 2-286. Unit Masks for IIO_CREDITS_REJECT

Extension	umask [15:8]	Description
DRS	bxxxx1xxx	DRS Credits to the IIO for the DRS message class.
NCB	bxxx1xxxx	NCB Credits to the IIO for the NCB message class.
NCS	bxx1xxxxx	NCS Credits to the IIO for the NCS message class.

IIO_CREDITS_USED

- **Title:** M2PCIe IIO Credits in Use
- **Category:** IIO_CREDITS Events
- **Event Code:** 0x32
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when one or more credits in the M2PCIe agent for sending transactions into the IIO on either NCB or NCS are in use. Transactions from the BL ring going into the IIO Agent must first acquire a credit. These credits are for either the NCB or NCS message classes. NCB, or non-coherent bypass messages are used to transmit data without coherency (and are common). NCS is used for reads to PCIe (and should be used sparingly).

Table 2-287. Unit Masks for IIO_CREDITS_USED

Extension	umask [15:8]	Description
DRS_0	bxxxxxxx1	DRS to CMS Port 0 Credits for transfer through CMS Port 0 to the IIO for the DRS message class.
DRS_1	bxxxxxx1x	DRS to CMS Port 1 Credits for transfer through CMS Port 0 to the IIO for the DRS message class.
NCB_0	bxxxxx1xx	NCB to CMS Port 0 Credits for transfer through CMS Port 0 to the IIO for the NCB message class.
NCB_1	bxxxx1xxx	NCB to CMS Port 1 Credits for transfer through CMS Port 0 to the IIO for the NCB message class.
NCS_0	bxxx1xxxx	NCS to CMS Port 0 Credits for transfer through CMS Port 0 to the IIO for the NCS message class.
NCS_1	bxx1xxxxx	NCS to CMS Port 1 Credit for transfer through CMS Port 0s to the IIO for the NCS message class.

LOCAL_DED_P2P_CRD_TAKEN_0

- **Title:** Local Dedicated P2P Credit Taken - 0
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x46
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:**

Table 2-288. Unit Masks for LOCAL_DED_P2P_CRD_TAKEN_0

Extension	umask [15:8]	Description
M2IOSF0_NCB	bxxxxxxx1	M2IOSF0 - NCB
M2IOSF0_NCS	bxxxxxx1x	M2IOSF0 - NCS
M2IOSF1_NCB	bxxxxx1xx	M2IOSF1 - NCB
M2IOSF1_NCS	bxxxx1xxx	M2IOSF1 - NCS
M2IOSF2_NCB	bxxx1xxxx	M2IOSF2 - NCB
M2IOSF2_NCS	bxx1xxxxx	M2IOSF2 - NCS
M2IOSF3_NCB	bx1xxxxxx	M2IOSF3 - NCB
M2IOSF3_NCS	b1xxxxxxx	M2IOSF3 - NCS

LOCAL_DED_P2P_CRD_TAKEN_1

- **Title:** Local Dedicated P2P Credit Taken - 1
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x47
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-289. Unit Masks for LOCAL_DED_P2P_CRD_TAKEN_1

Extension	umask [15:8]	Description
M2IOSF4_NCB	bxxxxxxx1	M2IOSF4 - NCB
M2IOSF4_NCS	bxxxxxx1x	M2IOSF4 - NCS
M2IOSF5_NCB	bxxxxx1xx	M2IOSF5 - NCB
M2IOSF5_NCS	bxxxx1xxx	M2IOSF5 - NCS

LOCAL_P2P_DED_RETURNED_0

- **Title:** Local P2P Dedicated Credits Returned - 0
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x19
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-290. Unit Masks for LOCAL_P2P_DED_RETURNED_0

Extension	umask [15:8]	Description
MS2IOSF0_NCB	bxxxxxxx1	M2IOSF0 - NCB
MS2IOSF0_NCS	bxxxxxx1x	M2IOSF0 - NCS
MS2IOSF1_NCB	bxxxxx1xx	M2IOSF1 - NCB
MS2IOSF1_NCS	bxxxx1xxx	M2IOSF1 - NCS
MS2IOSF3_NCB	bxxx1xxxx	M2IOSF3 - NCB

Table 2-290. Unit Masks for LOCAL_P2P_DED_RETURNED_0

Extension	umask [15:8]	Description
MS2IOSF2_NCB	bxxx1xxxx	M2IOSF2 - NCB
MS2IOSF2_NCS	bxx1xxxxx	M2IOSF2 - NCS
MS2IOSF3_NCS	bxx1xxxxx	M2IOSF3 - NCS

LOCAL_P2P_DED_RETURNED_1

- **Title:** Local P2P Dedicated Credits Returned - 1
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x1a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-291. Unit Masks for LOCAL_P2P_DED_RETURNED_1

Extension	umask [15:8]	Description
MS2IOSF4_NCB	bxxxxxxx1	M2IOSF4 - NCB
MS2IOSF4_NCS	bxxxxxx1x	M2IOSF4 - NCS
MS2IOSF5_NCB	bxxxxx1xx	M2IOSF5 - NCB
MS2IOSF5_NCS	bxxxx1xxx	M2IOSF5 - NCS

LOCAL_P2P_SHAR_RETURNED

- **Title:** Local P2P Shared Credits Returned
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x17
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:**

Table 2-292. Unit Masks for LOCAL_P2P_SHAR_RETURNED

Extension	umask [15:8]	Description
AGENT_0	bxxxxxxx1	Agent0
AGENT_1	bxxxxxx1x	Agent1
AGENT_2	bxxxxx1xx	Agent2

LOCAL_SHAR_P2P_CRD_RETURNED

- **Title:** Local Shared P2P Credit Returned to credit ring
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x44
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-293. Unit Masks for LOCAL_SHAR_P2P_CRD_RETURNED

Extension	umask [15:8]	Description
AGENT_0	bxxxxxxx1	Agent0
AGENT_1	bxxxxxx1x	Agent1
AGENT_2	bxxxxx1xx	Agent2
AGENT_3	bxxxx1xxx	Agent3
AGENT_4	bxxx1xxxx	Agent4
AGENT_5	bxx1xxxxx	Agent5

LOCAL_SHAR_P2P_CRD_TAKEN_0

- **Title:** Local Shared P2P Credit Taken - 0
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x40
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-294. Unit Masks for LOCAL_SHAR_P2P_CRD_TAKEN_0

Extension	umask [15:8]	Description
M2IOSF0_NCB	bxxxxxxx1	M2IOSF0 - NCB
M2IOSF0_NCS	bxxxxxx1x	M2IOSF0 - NCS
M2IOSF1_NCB	bxxxxx1xx	M2IOSF1 - NCB
M2IOSF1_NCS	bxxxx1xxx	M2IOSF1 - NCS
M2IOSF2_NCB	bxxx1xxxx	M2IOSF2 - NCB
M2IOSF2_NCS	bxx1xxxxx	M2IOSF2 - NCS
M2IOSF3_NCB	bx1xxxxxx	M2IOSF3 - NCB
M2IOSF3_NCS	b1xxxxxxx	M2IOSF3 - NCS

LOCAL_SHAR_P2P_CRD_TAKEN_1

- **Title:** Local Shared P2P Credit Taken - 1
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x41
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-295. Unit Masks for LOCAL_SHAR_P2P_CRD_TAKEN_1

Extension	umask [15:8]	Description
M2IOSF4_NCB	bxxxxxxx1	M2IOSF4 - NCB
M2IOSF4_NCS	bxxxxxx1x	M2IOSF4 - NCS
M2IOSF5_NCB	bxxxxx1xx	M2IOSF5 - NCB
M2IOSF5_NCS	bxxxx1xxx	M2IOSF5 - NCS

LOCAL_SHAR_P2P_CRD_WAIT_0

- **Title:** Waiting on Local Shared P2P Credit - 0
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x4a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-296. Unit Masks for LOCAL_SHAR_P2P_CRD_WAIT_0

Extension	umask [15:8]	Description
M2IOSF0_NCB	bxxxxxxx1	M2IOSF0 - NCB
M2IOSF0_NCS	bxxxxxx1x	M2IOSF0 - NCS
M2IOSF1_NCB	bxxxxx1xx	M2IOSF1 - NCB
M2IOSF1_NCS	bxxxx1xxx	M2IOSF1 - NCS
M2IOSF2_NCB	bxxx1xxxx	M2IOSF2 - NCB
M2IOSF2_NCS	bxx1xxxxx	M2IOSF2 - NCS
M2IOSF3_NCB	bx1xxxxxx	M2IOSF3 - NCB
M2IOSF3_NCS	b1xxxxxxx	M2IOSF3 - NCS

LOCAL_SHAR_P2P_CRD_WAIT_1

- **Title:** Waiting on Local Shared P2P Credit - 1
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x4b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-297. Unit Masks for LOCAL_SHAR_P2P_CRD_WAIT_1

Extension	umask [15:8]	Description
M2IOSF4_NCB	bxxxxxxx1	M2IOSF4 - NCB
M2IOSF4_NCS	bxxxxxx1x	M2IOSF4 - NCS
M2IOSF5_NCB	bxxxxx1xx	M2IOSF5 - NCB
M2IOSF5_NCS	bxxxx1xxx	M2IOSF5 - NCS

P2P_CRD_OCCUPANCY

- **Title:** P2P Credit Occupancy
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x14
- **Max. Inc/Cyc:.** 127

Register Restrictions: 0-1

- **Definition:**

Table 2-298. Unit Masks for P2P_CRD_OCCUPANCY

Extension	umask [15:8]	Description
LOCAL_NCB	bxxxxxxx1	Local NCB
LOCAL_NCS	bxxxxxx1x	Local NCS
REMOTE_NCB	bxxxxx1xx	Remote NCB
REMOTE_NCS	bxxxx1xxx	Remote NCS
ALL	bxxx1xxxx	All

P2P_DED_RECEIVED

- **Title:** Dedicated Credits Received
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x16
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-299. Unit Masks for P2P_DED_RECEIVED

Extension	umask [15:8]	Description
LOCAL_NCB	bxxxxxxx1	Local NCB
LOCAL_NCS	bxxxxxx1x	Local NCS
REMOTE_NCB	bxxxxx1xx	Remote NCB
REMOTE_NCS	bxxxx1xxx	Remote NCS
ALL	bxxx1xxxx	All

P2P_SHAR_RECEIVED

- **Title:** Shared Credits Received
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x15
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-300. Unit Masks for P2P_SHAR_RECEIVED

Extension	umask [15:8]	Description
LOCAL_NCB	bxxxxxxx1	Local NCB
LOCAL_NCS	bxxxxxx1x	Local NCS
REMOTE_NCB	bxxxxx1xx	Remote NCB
REMOTE_NCS	bxxxx1xxx	Remote NCS
ALL	bxxx1xxxx	All

REMOTE_DED_P2P_CRD_TAKEN_0

- **Title:** Remote Dedicated P2P Credit Taken - 0
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x48
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-301. Unit Masks for REMOTE_DED_P2P_CRD_TAKEN_0

Extension	umask [15:8]	Description
UPI0_DRS	bxxxxxxx1	UPI0 - DRS
UPI0_NCB	bxxxxxx1x	UPI0 - NCB
UPI0_NCS	bxxxxx1xx	UPI0 - NCS
UPI1_DRS	bxxxx1xxx	UPI1 - DRS
UPI1_NCB	bxxx1xxxx	UPI1 - NCB
UPI1_NCS	bxx1xxxxx	UPI1 - NCS

REMOTE_DED_P2P_CRD_TAKEN_1

- **Title:** Remote Dedicated P2P Credit Taken - 1
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x49
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-302. Unit Masks for REMOTE_DED_P2P_CRD_TAKEN_1

Extension	umask [15:8]	Description
UPI2_DRS	bxxxxxxx1	UPI2 - DRS
UPI2_NCB	bxxxxxx1x	UPI2 - NCB
UPI2_NCS	bxxxxx1xx	UPI2 - NCS

REMOTE_P2P_DED_RETURNED

- **Title:** Remote P2P Dedicated Credits Returned
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x1b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-303. Unit Masks for REMOTE_P2P_DED_RETURNED

Extension	umask [15:8]	Description
UPI0_NCB	bxxxxxxx1	UPI0 - NCB
UPI0_NCS	bxxxxxx1x	UPI0 - NCS

Table 2-303. Unit Masks for REMOTE_P2P_DED_RETURNED

Extension	umask [15:8]	Description
UPI1_NCB	bxxxxx1xx	UPI1 - NCB
UPI1_NCS	bxxxx1xxx	UPI1 - NCS
UPI2_NCB	bxxx1xxxx	UPI2 - NCB
UPI2_NCS	bxx1xxxxx	UPI2 - NCS

REMOTE_P2P_SHAR_RETURNED

- **Title:** Remote P2P Shared Credits Returned
- **Category:** INGRESS P2P Credit Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:**

Table 2-304. Unit Masks for REMOTE_P2P_SHAR_RETURNED

Extension	umask [15:8]	Description
AGENT_0	bxxxxxxx1	Agent0
AGENT_1	bxxxxxx1x	Agent1
AGENT_2	bxxxxx1xx	Agent2

REMOTE_SHAR_P2P_CRD_RETURNED

- **Title:** Remote Shared P2P Credit Returned to credit ring
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x45
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:**

Table 2-305. Unit Masks for REMOTE_SHAR_P2P_CRD_RETURNED

Extension	umask [15:8]	Description
AGENT_0	bxxxxxxx1	Agent0
AGENT_1	bxxxxxx1x	Agent1
AGENT_2	bxxxxx1xx	Agent2

REMOTE_SHAR_P2P_CRD_TAKEN_0

- **Title:** Remote Shared P2P Credit Taken - 0
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-306. Unit Masks for REMOTE_SHAR_P2P_CRD_TAKEN_0

Extension	umask [15:8]	Description
UPI0_DRS	bxxxxxxx1	UPI0 - DRS
UPI0_NCB	bxxxxxx1x	UPI0 - NCB
UPI0_NCS	bxxxxx1xx	UPI0 - NCS
UPI1_DRS	bxxxx1xxx	UPI1 - DRS
UPI1_NCB	bxxx1xxxx	UPI1 - NCB
UPI1_NCS	bxx1xxxxx	UPI1 - NCS

REMOTE_SHAR_P2P_CRD_TAKEN_1

- **Title:** Remote Shared P2P Credit Taken - 1
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x43
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-307. Unit Masks for REMOTE_SHAR_P2P_CRD_TAKEN_1

Extension	umask [15:8]	Description
UPI2_DRS	bxxxxxxx1	UPI2 - DRS
UPI2_NCB	bxxxxxx1x	UPI2 - NCB
UPI2_NCS	bxxxxx1xx	UPI2 - NCS

REMOTE_SHAR_P2P_CRD_WAIT_0

- **Title:** Waiting on Remote Shared P2P Credit - 0
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x4c
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-308. Unit Masks for REMOTE_SHAR_P2P_CRD_WAIT_0

Extension	umask [15:8]	Description
UPI0_DRS	bxxxxxxx1	UPI0 - DRS
UPI0_NCB	bxxxxxx1x	UPI0 - NCB
UPI0_NCS	bxxxxx1xx	UPI0 - NCS
UPI1_DRS	bxxxx1xxx	UPI1 - DRS
UPI1_NCB	bxxx1xxxx	UPI1 - NCB
UPI1_NCS	bxx1xxxxx	UPI1 - NCS

REMOTE_SHAR_P2P_CRD_WAIT_1

- **Title:** Waiting on Remote Shared P2P Credit - 1
- **Category:** EGRESS P2P Credit Events
- **Event Code:** 0x4d
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-309. Unit Masks for REMOTE_SHAR_P2P_CRD_WAIT_1

Extension	umask [15:8]	Description
UPI2_DRS	bxxxxxxx1	UPI2 - DRS
UPI2_NCB	bxxxxxx1x	UPI2 - NCB
UPI2_NCS	bxxxxx1xx	UPI2 - NCS

RxC_CYCLES_NE

- **Title:** Ingress (from CMS) Queue Cycles Not Empty
- **Category:** INGRESS Events
- **Event Code:** 0x10
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the M2PCIe Ingress is not empty.

Table 2-310. Unit Masks for RxC_CYCLES_NE

Extension	umask [15:8]	Description
CHA_IDI	bxxxxxxx1	
CHA_NCB	bxxxxxx1x	
CHA_NCS	bxxxxx1xx	
UPI_NCB	bxxxx1xxx	
UPI_NCS	bxxx1xxxx	
IIO_NCB	bxx1xxxxx	
IIO_NCS	bx1xxxxxx	
ALL	b1xxxxxxx	

RxC_INSERTS

- **Title:** Ingress (from CMS) Queue Inserts
- **Category:** INGRESS Events
- **Event Code:** 0x11
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of entries inserted into the M2PCIe Ingress Queue. This can be used in conjunction with the M2PCIe Ingress Occupancy Accumulator event in order to calculate average queue latency.

Table 2-311. Unit Masks for RxC_INSERTS

Extension	umask [15:8]	Description
CHA_IDI	bxxxxxxx1	
CHA_NCB	bxxxxxx1x	
CHA_NCS	bxxxxx1xx	
UPI_NCB	bxxxx1xxx	
UPI_NCS	bxxx1xxxx	
IIO_NCB	bxx1xxxxx	
IIO_NCS	bx1xxxxxx	
ALL	b1xxxxxxx	

TxC_CREDITS

- **Title:**
- **Category:** EGRESS Events
- **Event Code:** 0x2d
- **Max. Inc/Cyc:.** 4

Register Restrictions: 0-1

- **Definition:**

Table 2-312. Unit Masks for TxC_CREDITS

Extension	umask [15:8]	Description
PRQ	bxxxxxxx1	
PMM	bxxxxxx1x	

TxC_CYCLES_FULL

- **Title:** Egress (to CMS) Cycles Full
- **Category:** EGRESS Events
- **Event Code:** 0x25
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the M2PCIe Egress is full. This tracks messages for one of the two CMS ports that are used by the M2PCIe agent.

Table 2-313. Unit Masks for TxC_CYCLES_FULL

Extension	umask [15:8]	Description
AD_0	bxxxxxxx1	
AK_0	bxxxxxx1x	
BL_0	bxxxxx1xx	
PMM_BLOCK_1	bxxxx1xxx	
AD_1	bxxx1xxxx	
AK_1	bxx1xxxxx	

Table 2-313. Unit Masks for TxC_CYCLES_FULL

Extension	umask [15:8]	Description
BL_1	bx1xxxxxx	
PMM_BLOCK_0	b1xxxxxxx	

TxC_CYCLES_NE

- **Title:** Egress (to CMS) Cycles Not Empty
- **Category:** EGRESS Events
- **Event Code:** 0x23
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-1

- **Definition:** Counts the number of cycles when the M2PCIe Egress is not empty. This tracks messages for one of the two CMS ports that are used by the M2PCIe agent. This can be used in conjunction with the M2PCIe Ingress Occupancy Accumulator event in order to calculate average queue occupancy. Multiple egress buffers can be tracked at a given time using multiple counters.

Table 2-314. Unit Masks for TxC_CYCLES_NE

Extension	umask [15:8]	Description
AD_0	bxxxxxxx1	
AK_0	bxxxxxx1x	
BL_0	bxxxxx1xx	
PMM_DISTRESS_1	bxxxx1xxx	
AD_1	bxxx1xxxx	
AK_1	bxx1xxxxx	
BL_1	bx1xxxxxx	
PMM_DISTRESS_0	b1xxxxxxx	

TxC_INSERTS

- **Title:** Egress (to CMS) Ingress
- **Category:** EGRESS Events
- **Event Code:** 0x24
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of number of messages inserted into the the M2PCIe Egress queue. This tracks messages for one of the two CMS ports that are used by the M2PCIe agent. This can be used in conjunction with the M2PCIe Ingress Occupancy Accumulator event in order to calculate average queue occupancy.

Table 2-315. Unit Masks for TxC_INSERTS

Extension	umask [15:8]	Description
AD_0	bxxxxxxx1	
BL_0	bxxxxx1xx	
AK_CRD_0	bxxxx1xxx	
AD_1	bxxx1xxxx	

Table 2-315. Unit Masks for TxC_INSERTS

Extension	umask [15:8]	Description
BL_1	bx1xxxxxx	
AK_CRD_1	b1xxxxxxx	

2.10 M3UPI Performance Monitoring

M3UPI is the interface between the mesh and the Intel® UPI Link Layer. It is responsible for translating between mesh protocol packets and flits that are used for transmitting data across the Intel® UPI interface. It performs credit checking between the local Intel® UPI LL, the remote Intel® UPI LL and other agents on the local mesh.

The M3UPI agent provides several functions:

- Interface between Mesh and Intel® UPI:
  One of the primary attributes of the mesh is its ability to convey Intel® UPI semantics with no translation. For example, this architecture enables initiators to communicate with a local Home agent in exactly the same way as a remote Home agent on another 3rd Gen Intel Xeon Processor Scalable Family socket. With this philosophy, the M3UPI block is lean and does very little with regards to the Intel® UPI protocol aside from mirror the request between the mesh and the Intel® UPI interface.
- Intel® UPI routing:
  In order to optimize layout and latency, both full width Intel® UPI interfaces share the same mesh stop. Therefore, a Intel® UPI packet might be received on one interface and simply forwarded along on the other Intel® UPI interface. The M3UPI has sufficient routing logic to determine if a request, snoop or response is targeting the local socket or if it should be forwarded along to the other interface. This routing remains isolated to M3UPI and does not impede traffic on the Mesh.
- Intel® UPI Home Snoop Protocol (with early snoop optimizations for DP):
  The M3UPI agent implements a latency-reducing optimization for dual sockets which issues snoops within the socket for incoming requests as well as a latency-reducing optimization to return data satisfying Direct2Core (D2C) requests.

2.10.1 M3UPI Performance Monitoring Overview

Each M3UPI Link in supports event monitoring through three 48b wide counters (M3_Ly_PCI_PMON_CTR/CTL{2:0}). Each of these three counters can be programmed to count almost any M3UPI event (see NOTE for exceptions). The M3UPI counters can increment by a maximum of 6b per cycle.

Note: Only counter 0 can be used for tracking occupancy events. Only counter 2 can be used to count mesh events.

Note: There is a fourth counter that's broken?

2.10.2 M3UPI Performance Monitoring Events

M3UPI provides events to track information related to all the traffic passing through it's boundaries.

- VN/IIO credit tracking - in addition to tracking the occupancy of the full VNA queue, M3UPI provides a great deal of additional information: credits rejected, acquired and used often broken down by Message Class.
  Mesh Section 2.2, "Mesh Performance Monitoring."

2.10.3 M3UPI Box Events Ordered By Code

The following table summarizes the directly measured M3UPI Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
CLOCKTICKS	0x01	0-3	1	Clockticks of the mesh to Intel UPI (M3UPI)
TxC_AD_FLQ_OCCUPANCY	0x1c	0	8	AD Flow Q Occupancy
TxC_BL_FLQ_OCCUPANCY	0x1d	0	8	BL Flow Q Occupancy
TxC_AK_FLQ_OCCUPANCY	0x1e	0	8	AK Flow Q Occupancy
TxC_BL_WB_FLQ_OCCUPANCY	0x1f	0	8	BL Flow Q Occupancy
UPI_PEER_AD_CREDITS_EMPTY	0x20	0-3	1	UPI0 AD Credits Empty
UPI_PEER_BL_CREDITS_EMPTY	0x21	0-3	1	UPI0 BL Credits Empty
CHA_AD_CREDITS_EMPTY	0x22	0-3	1	CBox AD Credits Empty
M2_BL_CREDITS_EMPTY	0x23	0-3	1	M2 BL Credits Empty
TxC_AD_FLQ_CYCLES_NE	0x27	0-3	1	AD Flow Q Not Empty
TxC_BL_FLQ_CYCLES_NE	0x28	0-3	1	BL Flow Q Not Empty
UPI_PREFETCH_SPAWN	0x29	0-3	1	FlowQ Generated Prefetch
D2U_SENT	0x2a	0-3	1	D2U Sent
D2C_SENT	0x2b	0-3	1	D2C Sent
TxC_AD_FLQ_BYPASS	0x2c	0-3	1	AD FlowQ Bypass
TxC_AD_FLQ_INSERTS	0x2d	0-3	3	AD Flow Q Inserts
TxC_BL_FLQ_INSERTS	0x2e	0-3	1	BL Flow Q Inserts
TxC_AK_FLQ_INSERTS	0x2f	0-3	2	AK Flow Q Inserts
TxC_AD_ARB_FAIL	0x30		0	Failed ARB for AD
TxC_BL_ARB_FAIL	0x35		0	Failed ARB for BL
MULTI_SLOT_RCVD	0x3e	0-3	3	Multi Slot Flit Received
RxC_BYPASSED	0x40	0-2	1	Ingress Queue Bypasses
RxC_INSERTS_VN0	0x41		0	VN0 Ingress (from CMS) Queue - Inserts
RxC_INSERTS_VN1	0x42		0	VN1 Ingress (from CMS) Queue - Inserts
RxC_CYCLES_NE_VN0	0x43		0	VN0 Ingress (from CMS) Queue - Cycles Not Empty
RxC_CYCLES_NE_VN1	0x44		0	VN1 Ingress (from CMS) Queue - Cycles Not Empty
RxC_OCCUPANCY_VN0	0x45		0	VN0 Ingress (from CMS) Queue - Occupancy
RxC_OCCUPANCY_VN1	0x46		0	VN1 Ingress (from CMS) Queue - Occupancy
RxC_ARB_NOCRD_VN0	0x47		0	No Credits to Arb for VN0
RxC_ARB_NOCRD_VN1	0x48		0	No Credits to Arb for VN1

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
RxC_ARB_NOREQ_VN0	0x49		0	Can't Arb for VN0
RxC_ARB_NOREQ_VN1	0x4a		0	Can't Arb for VN1
RxC_ARB_LOST_VN0	0x4b		0	Lost Arb for VN0
RxC_ARB_LOST_VN1	0x4c		0	Lost Arb for VN1
RxC_ARB_MISC	0x4d		0	Arb Miscellaneous
RxC_PACKING_MISS_VN0	0x4e	0-2	0	VN0 message can't slot into flit
RxC_PACKING_MISS_VN1	0x4f	0-2	0	VN1 message can't slot into flit
RxC_HELD	0x50	0-2	0	Message Held
RxC_FLIT_GEN_HDR1	0x51		0	Flit Gen - Header 1
RxC_FLIT_GEN_HDR2	0x52		0	Flit Gen - Header 2
RxC_HDR_FLIT_NOT_SENT	0x53		0	Header Not Sent
RxC_HDR_FLITS_SENT	0x54		0	Sent Header Flit
RxC_DATA_FLITS_NOT_SENT	0x55		0	Data Flit Not Sent
RxC_FLITS_SLOT_BL	0x56		0	Slotting BL Message Into Header Flit
RxC_FLITS_GEN_BL	0x57		0	Generating BL Data Flit Sequence
RxC_FLITS_MISC	0x58		0	
RxC_VNA_CRD_MISC	0x59		0	
RxC_VNA_CRD	0x5a		0	Remote VNA Credits
VN0_CREDITS_USED	0x5b		0	VN0 Credit Used
VN1_CREDITS_USED	0x5c		0	VN1 Credit Used
VN0_NO_CREDITS	0x5d		0	VN0 No Credits
VN1_NO_CREDITS	0x5e		0	VN1 No Credits
RxC_CRD_MISC	0x5f		0	Miscellaneous Credit Events
RxC_CRD_OCC	0x60		0	Credit Occupancy
XPT_PFTCH	0x61		0	
WB_PENDING	0x7d		0	
WB_OCC_COMPARE	0x7e		0	

2.10.4 M3UPI Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the M3UPI Box.

CHA_AD_CREDITS_EMPTY

- **Title:** CBox AD Credits Empty
- **Category:** EGRESS Credit Events
- **Event Code:** 0x22
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** No credits available to send to Cbox on the AD Ring (covers higher CBoxes)

Table 2-316. Unit Masks for CHA_AD_CREDITS_EMPTY

Extension	umask [15:8]	Description
VNA	bxxxxxxx1	VNA Messages
WB	bxxxxxx1x	Writebacks
REQ	bxxxxx1xx	Requests
SNP	bxxxx1xxx	Snoops

CLOCKTICKS

- **Title:** Clockticks of the mesh to Intel UPI (M3UPI)
- **Category:** UCLK Events
- **Event Code:** 0x01
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of uclks in the M3 uclk domain. This could be slightly different than the count in the Ubox because of enable/freeze delays. However, because the M3 is close to the Ubox, they generally should not diverge by more than a handful of cycles.

D2C_SENT

- **Title:** D2C Sent
- **Category:** Special Egress Events
- **Event Code:** 0x2b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Count cases BL sends direct to core

D2U_SENT

- **Title:** D2U Sent
- **Category:** Special Egress Events
- **Event Code:** 0x2a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Cases where SMI3 sends D2U command
- **NOTE:** NOT required anymore

M2_BL_CREDITS_EMPTY

- **Title:** M2 BL Credits Empty
- **Category:** EGRESS Credit Events
- **Event Code:** 0x23
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** No vn0 and vna credits available to send to M2

Table 2-317. Unit Masks for M2_BL_CREDITS_EMPTY

Extension	umask [15:8]	Description
IIO1_NCB	bxxxxxxx1	IIO0 and IIO1 share the same ring destination. (1 VN0 credit only)
IIO2_NCB	bxxxxxx1x	IIO2
IIO3_NCB	bxxxxx1xx	IIO3
IIO4_NCB	bxxxx1xxx	IIO4
IIO5_NCB	bxxx1xxxx	IIO5
UBOX_NCB	bxx1xxxxx	Ubox
NCS	bx1xxxxxx	All IIO targets for NCS are in single mask. ORs them together
NCS_SEL	b1xxxxxxx	Selected M2p BL NCS credits

MULTI_SLOT_RCVD

- **Title:** Multi Slot Flit Received
- **Category:** Special Egress Events
- **Event Code:** 0x3e
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:** Multi slot flit received - S0, S1 and/or S2 populated (can use AK S0/S1 masks for AK allocations)
- **NOTE:** subevents added to 5b?

Table 2-318. Unit Masks for MULTI_SLOT_RCVD

Extension	umask [15:8]	Description
AD_SLOT0	bxxxxxxx1	AD - Slot 0
AD_SLOT1	bxxxxxx1x	AD - Slot 1
AD_SLOT2	bxxxxx1xx	AD - Slot 2
BL_SLOT0	bxxxx1xxx	BL - Slot 0
AK_SLOT0	bxxx1xxxx	AK - Slot 0
AK_SLOT2	bxx1xxxxx	AK - Slot 2

RxC_ARB_LOST_VN0

- **Title:** Lost Arb for VN0
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x4b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** VN0 message requested but lost arbitration

Table 2-319. Unit Masks for RxC_ARB_LOST_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_ARB_LOST_VN1

- **Title:** Lost Arb for VN1
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x4c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** VN1 message requested but lost arbitration

Table 2-320. Unit Masks for RxC_ARB_LOST_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.

Table 2-320. Unit Masks for RxC_ARB_LOST_VN1

Extension	umask [15:8]	Description
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_ARB_MISC

- **Title:** Arb Miscellaneous
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x4d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-321. Unit Masks for RxC_ARB_MISC

Extension	umask [15:8]	Description
NO_PROG_AD_VN0	bxxxxxxx1	No Progress on Pending AD VN0 Arbitration stage made no progress on pending ad vn0 messages because slotting stage cannot accept new message
NO_PROG_AD_VN1	bxxxxxx1x	No Progress on Pending AD VN1 Arbitration stage made no progress on pending ad vn1 messages because slotting stage cannot accept new message
NO_PROG_BL_VN0	bxxxxx1xx	No Progress on Pending BL VN0 Arbitration stage made no progress on pending bl vn0 messages because slotting stage cannot accept new message
NO_PROG_BL_VN1	bxxxx1xxx	No Progress on Pending BL VN1 Arbitration stage made no progress on pending bl vn1 messages because slotting stage cannot accept new message
ADBL_PARALLEL_WIN_VN0	bxxx1xxxx	AD, BL Parallel Win VN0 AD and BL messages won arbitration concurrently / in parallel
ADBL_PARALLEL_WIN_VN1	bxx1xxxxx	AD, BL Parallel Win VN1 AD and BL messages won arbitration concurrently / in parallel
VN01_PARALLEL_WIN	bx1xxxxxx	VN0, VN1 Parallel Win VN0 and VN1 arbitration sub-pipelines had parallel winners (at least one AD or BL on each side)
ALL_PARALLEL_WIN	b1xxxxxxx	Max Parallel Win VN0 and VN1 arbitration sub-pipelines both produced AD and BL winners (maximum possible parallel winners)

RxC_ARB_NOCRD_VN0

- **Title:** No Credits to Arb for VN0
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x47
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** VN0 message is blocked from requesting arbitration due to lack of remote Intel UPI credits

Table 2-322. Unit Masks for RxC_ARB_NOCRD_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_ARB_NOCRD_VN1

- **Title:** No Credits to Arb for VN1
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x48
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** VN1 message is blocked from requesting arbitration due to lack of remote Intel UPI credits

Table 2-323. Unit Masks for RxC_ARB_NOCRD_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.

Table 2-323. Unit Masks for RxC_ARB_NOCRD_VN1

Extension	umask [15:8]	Description
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_ARB_NOREQ_VN0

- **Title:** Can't Arb for VN0
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x49
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** VN0 message was not able to request arbitration while some other message won arbitration

Table 2-324. Unit Masks for RxC_ARB_NOREQ_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_ARB_NOREQ_VN1

- **Title:** Can't Arb for VN1
- **Category:** INGRESS Arbitration Events
- **Event Code:** 0x4a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** VN1 message was not able to request arbitration while some other message won arbitration

Table 2-325. Unit Masks for RxC_ARB_NOREQ_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_BYPASSED

- **Title:** Ingress Queue Bypasses
- **Category:** INGRESS Events
- **Event Code:** 0x40
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-2

- **Definition:** Number ot times message is bypassed around the Ingress Queue

Table 2-326. Unit Masks for RxC_BYPASSED

Extension	umask [15:8]	Description
AD_S0_IDLE	bxxxxxxx1	AD to Slot 0 on Idle AD is taking bypass to slot 0 of independent flit while pipeline is idle
AD_S0_BL_ARB	bxxxxxx1x	AD to Slot 0 on BL Arb AD is taking bypass to slot 0 of independent flit while bl message is in arbitration
AD_S1_BL_SLOT	bxxxxx1xx	AD + BL to Slot 1 AD is taking bypass to flit slot 1 while merging with bl message in same flit
AD_S2_BL_SLOT	bxxxx1xxx	AD + BL to Slot 2 AD is taking bypass to flit slot 2 while merging with bl message in same flit

RxC_CRD_MISC

- **Title:** Miscellaneous Credit Events
- **Category:** INGRESS Credit Events
- **Event Code:** 0x5f
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-327. Unit Masks for RxC_CRD_MISC

Extension	umask [15:8]	Description
ANY_BGF_FIFO	bxxxxxxx1	Any In BGF FIFO Indication that at least one packet (flit) is in the bgf (fifo only)
ANY_BGF_PATH	bxxxxxx1x	Any in BGF Path Indication that at least one packet (flit) is in the bgf path (i.e. pipe to fifo)
VN0_NO_D2K_FOR_ARB	bxxxxx1xx	No D2K For Arb VN0 BL RSP message was blocked from arbitration request due to lack of D2K CMP credit
VN1_NO_D2K_FOR_ARB	bxxxx1xxx	VN1 BL RSP message was blocked from arbitration request due to lack of D2K CMP credits
LT1_FOR_D2K	bxxx1xxxx	d2k credit count is less than 1
LT2_FOR_D2K	bxx1xxxxx	d2k credit count is less than 2

RxC_CRD_OCC

- **Title:** Credit Occupancy
- **Category:** INGRESS Credit Events
- **Event Code:** 0x60
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-328. Unit Masks for RxC_CRD_OCC

Extension	umask [15:8]	Description
VNA_IN_USE	bxxxxxxx1	VNA In Use Remote Intel UPI VNA credit occupancy (number of credits in use), accumulated across all cycles
FLITS_IN_FIFO	bxxxxxx1x	Packets in BGF FIFO Occupancy of m3upi ingress -> Intel UPI link layer bgf; packets (flits) in fifo
FLITS_IN_PATH	bxxxxx1xx	Packets in BGF Path Occupancy of m3upi ingress -> Intel UPI link layer bgf; packets (flits) in path (i.e. pipe to fifo or fifo)
TxQ_CRD	bxxxx1xxx	Transmit Credits Link layer transmit queue credit occupancy (credits in use), accumulated across all cycles
D2K_CRD	bxxx1xxxx	D2K Credits D2K completion fifo credit occupancy (credits in use), accumulated across all cycles
P1P_TOTAL	bxx1xxxxx	count of bl messages in pump-1-pending state, in marker table and in fifo

Table 2-328. Unit Masks for RxC_CRD_OCC

Extension	umask [15:8]	Description
P1P_FIFO	bx1xxxxxx	count of bl messages in pump-1-pending state, in completion fifo only
CONSUMED	b1xxxxxxx	Credits Consumed number of remote vna credits consumed per cycle

RxC_CYCLES_NE_VN0

- **Title:** VN0 Ingress (from CMS) Queue - Cycles Not Empty
- **Category:** INGRESS Events
- **Event Code:** 0x43
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of cycles when the Intel UPI Ingress is not empty. This tracks one of the three rings that are used by the Intel UPI agent. This can be used in conjunction with the Intel UPI Ingress Occupancy Accumulator event in order to calculate average queue occupancy. Multiple ingress buffers can be tracked at a given time using multiple counters.

Table 2-329. Unit Masks for RxC_CYCLES_NE_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_CYCLES_NE_VN1

- **Title:** VN1 Ingress (from CMS) Queue - Cycles Not Empty
- **Category:** INGRESS Events
- **Event Code:** 0x44
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of allocations into the Intel UPI VN1 Ingress. This tracks one of the three rings that are used by the Intel UPI agent. This can be used in conjunction with the Intel UPI VN1 Ingress Occupancy Accumulator event in order

to calculate average queue latency. Multiple ingress buffers can be tracked at a given time using multiple counters.

Table 2-330. Unit Masks for RxC_CYCLES_NE_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_DATA_FLITS_NOT_SENT

- **Title:** Data Flit Not Sent
- **Category:** INGRESS Flit Events
- **Event Code:** 0x55
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Data flit is ready for transmission but could not be sent

Table 2-331. Unit Masks for RxC_DATA_FLITS_NOT_SENT

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All data flit is ready for transmission but could not be sent for any reason, For example, low credits, low tsv, stall injection
TSV_HI	bxxxxxx1x	TSV High data flit is ready for transmission but was not sent while tsv high
VALID_FOR_FLIT	bxxxxx1xx	Cycle valid for Flit data flit is ready for transmission but was not sent while cycle is valid for flit transmission
NO_BGF	bxxxx1xxx	No BGF Credits
NO_TXQ	bxxx1xxxx	No TxQ Credits

RxC_FLITS_GEN_BL

- **Title:** Generating BL Data Flit Sequence
- **Category:** INGRESS Flit Events
- **Event Code:** 0x57
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-332. Unit Masks for RxC_FLITS_GEN_BL

Extension	umask [15:8]	Description
P0_WAIT	bxxxxxxx1	Wait on Pump 0 generating bl data flit sequence; waiting for data pump 0
P1_WAIT	bxxxxxx1x	Wait on Pump 1 generating bl data flit sequence; waiting for data pump 1
P1P_TO_LIMBO	bxxxxx1xx	a bl message finished but is in limbo and moved to pump-1-pending logic
P1P_BUSY	bxxxx1xxx	pump-1-pending logic is tracking at least one message
P1P_AT_LIMIT	bxxx1xxxx	pump-1-pending logic is at capacity (pending table plus completion fifo at limit)
P1P_HOLD_P0	bxx1xxxxx	pump-1-pending logic is at or near capacity, such that pump-0-only bl messages are getting stalled in slotting stage
P1P_FIFO_FULL	bx1xxxxxx	pump-1-pending completion fifo is full

RxC_FLITS_MISC

- **Title:**
- **Category:** INGRESS Flit Events
- **Event Code:** 0x58
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-333. Unit Masks for RxC_FLITS_MISC

Extension	umask [15:8]	Description
S2REQ_RECEIVED	bxxxxxxx1	slot 2 request received from link layer while idle (with no slot 2 request active immediately prior)
S2REQ_WITHDRAWN	bxxxxxx1x	slot 2 request withdrawn during hold-off period or service window
S2REQ_IN_HOLDOFF	bxxxxx1xx	slot 2 request naturally serviced during hold-off period
S2REQ_IN_SERVICE	bxxxx1xxx	slot 2 request forcibly serviced during service window

RxC_FLITS_SLOT_BL

- **Title:** Slotting BL Message Into Header Flit
- **Category:** INGRESS Flit Events
- **Event Code:** 0x56
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-334. Unit Masks for RxC_FLITS_SLOT_BL

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All
NEED_DATA	bxxxxxx1x	Needs Data Flit BL message requires data flit sequence
P0_WAIT	bxxxxx1xx	Wait on Pump 0 Waiting for header pump 0
P1_WAIT	bxxxx1xxx	Wait on Pump 1 Waiting for header pump 1
P1_NOT_REQ	bxxx1xxxx	Don't Need Pump 1 Header pump 1 is not required for flit
P1_NOT_REQ_BUT_BUBBLE	bxx1xxxxx	Don't Need Pump 1 - Bubble Header pump 1 is not required for flit but flit transmission delayed
P1_NOT_REQ_NOT_AVAIL	bx1xxxxxx	Don't Need Pump 1 - Not Avail Header pump 1 is not required for flit and not available

RxC_FLIT_GEN_HDR1

- **Title:** Flit Gen - Header 1
- **Category:** INGRESS Flit Events
- **Event Code:** 0x51
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Events related to Header Flit Generation - Set 1

Table 2-335. Unit Masks for RxC_FLIT_GEN_HDR1

Extension	umask [15:8]	Description
ACCUM	bxxxxxxx1	Acumullate Header flit slotting control state machine is in any accumulate state; multi-message flit may be assembled over multiple cycles
ACCUM_READ	bxxxxxx1x	Accumulate Ready header flit slotting control state machine is in accum_ready state; flit is ready to send but transmission is blocked; more messages may be slotted into flit
ACCUM_WASTED	bxxxxx1xx	Accumulate Wasted Flit is being assembled over multiple cycles, but no additional message is being slotted into flit in current cycle; accumulate cycle is wasted
AHEAD_BLOCKED	bxxxx1xxx	Run-Ahead - Blocked Header flit slotting entered run-ahead state; new header flit is started while transmission of prior, fully assembled flit is blocked
AHEAD_MSG1_DURING	bxxx1xxxx	Run-Ahead - Message run-ahead mode: one message slotted during run-ahead
AHEAD_MSG2_AFTER	bxx1xxxxx	run-ahead mode: second message slotted immediately after run-ahead; potential run-ahead success
AHEAD_MSG2_SENT	bx1xxxxxx	run-ahead mode: two (or three) message flit sent immediately after run-ahead; complete run-ahead success
AHEAD_MSG1_AFTER	b1xxxxxxx	run-ahead mode: message was slotted only after run-ahead was over; run-ahead mode definitely wasted

RxC_FLIT_GEN_HDR2

- **Title:** Flit Gen - Header 2
- **Category:** INGRESS Flit Events
- **Event Code:** 0x52
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Events related to Header Flit Generation - Set 2

Table 2-336. Unit Masks for RxC_FLIT_GEN_HDR2

Extension	umask [15:8]	Description
RMSTALL	bxxxxxxx1	Rate-matching Stall Rate-matching stall injected
RMSTALL_NOMSG	bxxxxxx1x	Rate-matching Stall - No Message Rate matching stall injected, but no additional message slotted during stall cycle
PAR	bxxxxx1xx	Parallel Ok new header flit construction may proceed in parallel with data flit sequence
PAR_MSG	bxxxx1xxx	Parallel Message message is slotted into header flit in parallel with data flit sequence
PAR_FLIT	bxxx1xxxx	Parallel Flit Finished header flit finished assembly in parallel with data flit sequence

RxC_HDR_FLITS_SENT

- **Title:** Sent Header Flit
- **Category:** INGRESS Flit Events
- **Event Code:** 0x54
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-337. Unit Masks for RxC_HDR_FLITS_SENT

Extension	umask [15:8]	Description
1_MSG	bxxxxxxx1	One Message One message in flit; VNA or non-VNA flit
2_MSGS	bxxxxxx1x	Two Messages Two messages in flit; VNA flit
3_MSGS	bxxxxx1xx	Three Messages Three messages in flit; VNA flit
1_MSG_VNX	bxxxx1xxx	One Message in non-VNA One message in flit; non-VNA flit
SLOTS_1	bxxx1xxxx	One Slot Taken
SLOTS_2	bxx1xxxxx	Two Slots Taken
SLOTS_3	bx1xxxxxx	All Slots Taken

RxC_HDR_FLIT_NOT_SENT

- **Title:** Header Not Sent
- **Category:** INGRESS Flit Events
- **Event Code:** 0x53
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** header flit is ready for transmission but could not be sent

Table 2-338. Unit Masks for RxC_HDR_FLIT_NOT_SENT

Extension	umask [15:8]	Description
ALL	bxxxxxxx1	All header flit is ready for transmission but could not be sent for any reason, For example, no credits, low tsv, stall injection
TSV_HI	bxxxxxx1x	TSV High header flit is ready for transmission but was not sent while tsv high
VALID_FOR_FLIT	bxxxxx1xx	Cycle valid for Flit header flit is ready for transmission but was not sent while cycle is valid for flit transmission
NO_BGF_CRD	bxxxx1xxx	No BGF Credits No BGF credits available
NO_TXQ_CRD	bxxx1xxxx	No TxQ Credits No TxQ credits available
NO_BGF_NO_MSG	bxx1xxxxx	No BGF Credits + No Extra Message Slotted No BGF credits available; no additional message slotted into flit
NO_TXQ_NO_MSG	bx1xxxxxx	No TxQ Credits + No Extra Message Slotted No TxQ credits available; no additional message slotted into flit

RxC_HELD

- **Title:** Message Held
- **Category:** INGRESS Sloting Events
- **Event Code:** 0x50
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-2

- **Definition:**

Table 2-339. Unit Masks for RxC_HELD

Extension	umask [15:8]	Description
VN0	bxxxxxxx1	VN0 vn0 message(s) that couldn't be slotted into last vn0 flit are held in slotting stage while processing vn1 flit
VN1	bxxxxxx1x	VN1 vn1 message(s) that couldn't be slotted into last vn1 flit are held in slotting stage while processing vn0 flit
PARALLEL_ATTEMPT	bxxxxx1xx	Parallel Attempt ad and bl messages attempted to slot into the same flit in parallel
PARALLEL_SUCCESS	bxxxx1xxx	Parallel Success ad and bl messages were actually slotted into the same flit in paralle

Table 2-339. Unit Masks for RxC_HELD

Extension	umask [15:8]	Description
CANT_SLOT_AD	bxxx1xxxx	Can't Slot AD some AD message could not be slotted (logical OR of all AD events under INGR_SLOT_CANT_MC_VN{0,1})
CANT_SLOT_BL	bxx1xxxxx	Can't Slot BL some BL message could not be slotted (logical OR of all BL events under INGR_SLOT_CANT_MC_VN{0,1})

RxC_INSERTS_VN0

- **Title:** VN0 Ingress (from CMS) Queue - Inserts
- **Category:** INGRESS Events
- **Event Code:** 0x41
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of allocations into the Intel UPI Ingress. This tracks one of the three rings that are used by the Intel UPI agent. This can be used in conjunction with the Intel UPI Ingress Occupancy Accumulator event in order to calculate average queue latency. Multiple ingress buffers can be tracked at a given time using multiple counters.
- **NOTE:** subevents added to 5b?

Table 2-340. Unit Masks for RxC_INSERTS_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_INSERTS_VN1

- **Title:** VN1 Ingress (from CMS) Queue - Inserts
- **Category:** INGRESS Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Counts the number of allocations into the Intel UPI VN1 Ingress. This tracks one of the three rings that are used by the Intel UPI agent. This can be used in conjunction with the Intel UPI VN1 Ingress Occupancy Accumulator event in order to calculate average queue latency. Multiple ingress buffers can be tracked at a given time using multiple counters.
- **NOTE:** subevents added to 5b?

Table 2-341. Unit Masks for RxC_INSERTS_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_OCCUPANCY_VN0

- **Title:** VN0 Ingress (from CMS) Queue - Occupancy
- **Category:** INGRESS Events
- **Event Code:** 0x45
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Accumulates the occupancy of a given Intel UPI VN1 Ingress queue in each cycle. This tracks one of the three ring Ingress buffers. This can be used with the Intel UPI VN1 Ingress Not Empty event to calculate average occupancy or the Intel UPI VN1 Ingress Allocations event in order to calculate average queuing latency.

Table 2-342. Unit Masks for RxC_OCCUPANCY_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).

Table 2-342. Unit Masks for RxC_OCCUPANCY_VN0

Extension	umask [15:8]	Description
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_OCCUPANCY_VN1

- **Title:** VN1 Ingress (from CMS) Queue - Occupancy
- **Category:** INGRESS Events
- **Event Code:** 0x46
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Accumulates the occupancy of a given Intel UPI VN1 Ingress queue in each cycle. This tracks one of the three ring Ingress buffers. This can be used with the Intel UPI VN1 Ingress Not Empty event to calculate average occupancy or the Intel UPI VN1 Ingress Allocations event in order to calculate average queuing latency.

Table 2-343. Unit Masks for RxC_OCCUPANCY_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_PACKING_MISS_VN0

- **Title:** VN0 message can't slot into flit
- **Category:** INGRESS Sloting Events
- **Event Code:** 0x4e
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-2

- **Definition:** Count cases where Ingress has packets to send but did not have time to pack into flit before sending to Agent so slot was left NULL which could have been used.

Table 2-344. Unit Masks for RxC_PACKING_MISS_VN0

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_PACKING_MISS_VN1

- **Title:** VN1 message can't slot into flit
- **Category:** INGRESS Sloting Events
- **Event Code:** 0x4f
- **Max. Inc/Cyc:.** 0

Register Restrictions: 0-2

- **Definition:** Count cases where Ingress has packets to send but did not have time to pack into flit before sending to Agent so slot was left NULL which could have been used.

Table 2-345. Unit Masks for RxC_PACKING_MISS_VN1

Extension	umask [15:8]	Description
AD_REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
AD_SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.

Table 2-345. Unit Masks for RxC_PACKING_MISS_VN1

Extension	umask [15:8]	Description
AD_RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_RSP	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
BL_WB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
BL_NCB	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.
BL_NCS	bx1xxxxxx	NCS on BL Non-Coherent Standard (NCS) messages on BL.

RxC_VNA_CRD

- **Title:** Remote VNA Credits
- **Category:** INGRESS Credit Events
- **Event Code:** 0x5a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-346. Unit Masks for RxC_VNA_CRD

Extension	umask [15:8]	Description
CORRECTED	bxxxxxxx1	Corrected Number of remote vna credits corrected (local return) per cycle
LT1	bxxxxxx1x	Level < 1 Remote vna credit level is less than 1 (i.e. no vna credits available)
LT4	bxxxxx1xx	Level < 4 Remote vna credit level is less than 4; bl (or ad requiring 4 vna) cannot arb on vna
LT5	bxxxx1xxx	Level < 5 Remote vna credit level is less than 5; parallel ad/bl arb on vna not possible
LT10	bxxx1xxxx	Level < 10 remote vna credit level is less than 10; parallel vn0/vn1 arb not possible
ANY_IN_USE	bxx1xxxxx	Any In Use At least one remote vna credit is in use

RxC_VNA_CRD_MISC

- **Title:**
- **Category:** INGRESS Credit Events
- **Event Code:** 0x59
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-347. Unit Masks for RxC_VNA_CRD_MISC

Extension	umask [15:8]	Description
REQ_VN01_ALLOC_LT10	bxxxxxxx1	remote vna credit count was less than 10 and allocation to vn0 or vn1 was required
REQ_ADBL_ALLOC_L5	bxxxxxx1x	remote vna credit count was less than 5 and allocation to ad or bl messages was required
VN0_ONLY	bxxxxx1xx	remote vna credits were allocated only to vn0, not to vn1
VN1_ONLY	bxxxx1xxx	remote vna credits were allocated only to vn1, not to vn0
VN0_JUST_AD	bxxx1xxxx	on vn0, remote vna credits were allocated only to ad messages, not to bl
VN0_JUST_BL	bxx1xxxxx	on vn0, remote vna credits were allocated only to bl messages, not to ad
VN1_JUST_AD	bx1xxxxxx	on vn1, remote vna credits were allocated only to ad messages, not to bl
VN1_JUST_BL	b1xxxxxxx	on vn1, remote vna credits were allocated only to bl messages, not to ad

TxC_AD_ARB_FAIL

- **Title:** Failed ARB for AD
- **Category:** ARB Events
- **Event Code:** 0x30
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** AD arb but no win; arb request asserted but not won
- **NOTE:** subevents added to 5b?

Table 2-348. Unit Masks for TxC_AD_ARB_FAIL

Extension	umask [15:8]	Description
VN0_REQ	bxxxxxxx1	VN0 REQ Messages
VN0_SNP	bxxxxxx1x	VN0 SNP Messages
VN0_RSP	bxxxxx1xx	VN0 RSP Messages
VN0_WB	bxxxx1xxx	VN0 WB Messages
VN1_REQ	bxxx1xxxx	VN1 REQ Messages
VN1_SNP	bxx1xxxxx	VN1 SNP Messages
VN1_RSP	bx1xxxxxx	VN1 RSP Messages
VN1_WB	b1xxxxxxx	VN1 WB Messages

TxC_AD_FLQ_BYPASS

- **Title:** AD FlowQ Bypass
- **Category:** Special Egress Events
- **Event Code:** 0x2c
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts cases when the AD flowQ is bypassed (S0, S1 and S2 indicate which slot was bypassed with S0 having the highest priority and S2 the least)

Table 2-349. Unit Masks for TxC_AD_FLQ_BYPASS

Extension	umask [15:8]	Description
AD_SLOT0	bxxxxxxx1	
AD_SLOT1	bxxxxxx1x	
AD_SLOT2	bxxxxx1xx	
BL_EARLY_RSP	bxxxx1xxx	

TxC_AD_FLQ_CYCLES_NE

- **Title:** AD Flow Q Not Empty
- **Category:** FlowQ Events
- **Event Code:** 0x27
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of cycles the AD Egress queue is Not Empty
- **NOTE:** Counts the number of cycles when the QPI FlowQ is not empty. This tracks one of the three rings that are used by the QPI agent. This can be used in conjunction with the QPI FlowQ Occupancy Accumulator event in order to calculate average queue occupancy. Only a single FlowQ queue can be tracked at any given time. It is not possible to filter based on direction or polarity.

Table 2-350. Unit Masks for TxC_AD_FLQ_CYCLES_NE

Extension	umask [15:8]	Description
VN0_REQ	bxxxxxxx1	VN0 REQ Messages
VN0_SNP	bxxxxxx1x	VN0 SNP Messages
VN0_RSP	bxxxxx1xx	VN0 RSP Messages
VN0_WB	bxxxx1xxx	VN0 WB Messages
VN1_REQ	bxxx1xxxx	VN1 REQ Messages
VN1_SNP	bxx1xxxxx	VN1 SNP Messages
VN1_RSP	bx1xxxxxx	VN1 RSP Messages
VN1_WB	b1xxxxxxx	VN1 WB Messages

TxC_AD_FLQ_INSERTS

- **Title:** AD Flow Q Inserts
- **Category:** FlowQ Events
- **Event Code:** 0x2d
- **Max. Inc/Cyc:.** 3

Register Restrictions: 0-3

- **Definition:** Counts the number of allocations into the QPI FlowQ. This can be used in conjunction with the QPI FlowQ Occupancy Accumulator event in order to calculate average queue latency. Only a single FlowQ queue can be tracked at any given time. It is not possible to filter based on direction or polarity.

Table 2-351. Unit Masks for TxC_AD_FLQ_INSERTS

Extension	umask [15:8]	Description
VN0_REQ	bxxxxxxx1	VN0 REQ Messages
VN0_SNP	bxxxxxx1x	VN0 SNP Messages
VN0_RSP	bxxxxx1xx	VN0 RSP Messages
VN0_WB	bxxxx1xxx	VN0 WB Messages
VN1_REQ	bxxx1xxxx	VN1 REQ Messages
VN1_SNP	bxx1xxxxx	VN1 SNP Messages
VN1_RSP	bx1xxxxxx	VN1 RSP Messages

TxC_AD_FLQ_OCCUPANCY

- **Title:** AD Flow Q Occupancy
- **Category:** FlowQ Events
- **Event Code:** 0x1c
- **Max. Inc/Cyc:.** 8

Register Restrictions: 0

- **Definition:**

Table 2-352. Unit Masks for TxC_AD_FLQ_OCCUPANCY

Extension	umask [15:8]	Description
VN0_REQ	bxxxxxxx1	VN0 REQ Messages
VN0_SNP	bxxxxxx1x	VN0 SNP Messages
VN0_RSP	bxxxxx1xx	VN0 RSP Messages
VN0_WB	bxxxx1xxx	VN0 WB Messages
VN1_REQ	bxxx1xxxx	VN1 REQ Messages
VN1_SNP	bxx1xxxxx	VN1 SNP Messages
VN1_RSP	bx1xxxxxx	VN1 RSP Messages

TxC_AK_FLQ_INSERTS

- **Title:** AK Flow Q Inserts
- **Category:** FlowQ Events
- **Event Code:** 0x2f
- **Max. Inc/Cyc:.** 2

Register Restrictions: 0-3

- **Definition:**

TxC_AK_FLQ_OCCUPANCY

- **Title:** AK Flow Q Occupancy
- **Category:** FlowQ Events
- **Event Code:** 0x1e
- **Max. Inc/Cyc:.** 8

Register Restrictions: 0

- **Definition:**

TxC_BL_ARB_FAIL

- **Title:** Failed ARB for BL
- **Category:** ARB Events
- **Event Code:** 0x35
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** BL arb but no win; arb request asserted but not won
- **NOTE:** subevents added to 5b?

Table 2-353. Unit Masks for TxC_BL_ARB_FAIL

Extension	umask [15:8]	Description
VN0_RSP	bxxxxxxx1	VN0 RSP Messages
VN0_WB	bxxxxxx1x	VN0 WB Messages
VN0_NCB	bxxxxx1xx	VN0 NCB Messages
VN0_NCS	bxxxx1xxx	VN0 NCS Messages
VN1_RSP	bxxx1xxxx	VN1 RSP Messages
VN1_WB	bxx1xxxxx	VN1 WB Messages
VN1_NCB	bx1xxxxxx	VN1 NCS Messages
VN1_NCS	b1xxxxxxx	VN1 NCB Messages

TxC_BL_FLQ_CYCLES_NE

- **Title:** BL Flow Q Not Empty
- **Category:** FlowQ Events
- **Event Code:** 0x28
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of cycles the BL Egress queue is Not Empty
- **NOTE:** Counts the number of cycles when the QPI FlowQ is not empty. This tracks one of the three rings that are used by the QPI agent. This can be used in conjunction with the QPI FlowQ Occupancy Accumulator event in order to calculate average queue occupancy. Only a single FlowQ queue can be tracked at any given time. It is not possible to filter based on direction or polarity.

Table 2-354. Unit Masks for TxC_BL_FLQ_CYCLES_NE

Extension	umask [15:8]	Description
VN0_REQ	bxxxxxxx1	VN0 REQ Messages
VN0_SNP	bxxxxxx1x	VN0 SNP Messages
VN0_RSP	bxxxxx1xx	VN0 RSP Messages
VN0_WB	bxxxx1xxx	VN0 WB Messages
VN1_REQ	bxxx1xxxx	VN1 REQ Messages
VN1_SNP	bxx1xxxxx	VN1 SNP Messages
VN1_RSP	bx1xxxxxx	VN1 RSP Messages
VN1_WB	b1xxxxxxx	VN1 WB Messages

TxC_BL_FLQ_INSERTS

- **Title:** BL Flow Q Inserts
- **Category:** FlowQ Events
- **Event Code:** 0x2e
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of allocations into the QPI FlowQ. This can be used in conjunction with the QPI FlowQ Occupancy Accumulator event in order to calculate average queue latency. Only a single FlowQ queue can be tracked at any given time. It is not possible to filter based on direction or polarity.

Table 2-355. Unit Masks for TxC_BL_FLQ_INSERTS

Extension	umask [15:8]	Description
VN0_NCB	bxxxxxxx1	VN0 RSP Messages
VN0_NCS	bxxxxxx1x	VN0 WB Messages
VN0_WB	bxxxxx1xx	VN0 NCB Messages
VN0_RSP	bxxxx1xxx	VN0 NCS Messages
VN1_NCB	bxxx1xxxx	VN1 RSP Messages
VN1_NCS	bxx1xxxxx	VN1 WB Messages
VN1_WB	bx1xxxxxx	VN1_NCS Messages
VN1_RSP	b1xxxxxxx	VN1_NCB Messages

TxC_BL_FLQ_OCCUPANCY

- **Title:** BL Flow Q Occupancy
- **Category:** FlowQ Events
- **Event Code:** 0x1d
- **Max. Inc/Cyc:.** 8

Register Restrictions: 0

- **Definition:**

Table 2-356. Unit Masks for TxC_BL_FLQ_OCCUPANCY

Extension	umask [15:8]	Description
VN0_RSP	bxxxxxxx1	VN0 RSP Messages
VN0_WB	bxxxxxx1x	VN0 WB Messages
VN0_NCB	bxxxxx1xx	VN0 NCB Messages
VN0_NCS	bxxxx1xxx	VN0 NCS Messages
VN1_RSP	bxxx1xxxx	VN1 RSP Messages
VN1_WB	bxx1xxxxx	VN1 WB Messages
VN1_NCB	bx1xxxxxx	VN1_NCS Messages
VN1_NCS	b1xxxxxxx	VN1_NCB Messages

TxC_BL_WB_FLQ_OCCUPANCY

- **Title:** BL Flow Q Occupancy
- **Category:** FlowQ Events
- **Event Code:** 0x1f
- **Max. Inc/Cyc:.** 8

Register Restrictions: 0

- **Definition:**

Table 2-357. Unit Masks for TxC_BL_WB_FLQ_OCCUPANCY

Extension	umask [15:8]	Description
VN0_LOCAL	b00000001	VN0 RSP Messages
VN0_THROUGH	b00000010	VN0 WB Messages
VN0_WRPULL	b00000100	VN0 NCB Messages
VN1_LOCAL	b00010000	VN1 RSP Messages
VN1_THROUGH	b00100000	VN1 WB Messages
VN1_WRPULL	b01000000	VN1_NCS Messages

UPI_PEER_AD_CREDITS_EMPTY

- **Title:** UPI0 AD Credits Empty
- **Category:** EGRESS Credit Events
- **Event Code:** 0x20
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** No credits available to send to Intel UPIs on the AD Ring
- **NOTE:** 2 cases for non-smi3 mode and 3 cases for smi3 mode

Table 2-358. Unit Masks for UPI_PEER_AD_CREDITS_EMPTY

Extension	umask [15:8]	Description
VNA	bxxxxxxx1	VNA
VN0_REQ	bxxxxxx1x	VN0 REQ Messages
VN0_SNP	bxxxxx1xx	VN0 SNP Messages
VN0_RSP	bxxxx1xxx	VN0 RSP Messages
VN1_REQ	bxxx1xxxx	VN1 REQ Messages
VN1_SNP	bxx1xxxxx	VN1 SNP Messages
VN1_RSP	bx1xxxxxx	VN1 RSP Messages

UPI_PEER_BL_CREDITS_EMPTY

- **Title:** UPI0 BL Credits Empty
- **Category:** EGRESS Credit Events
- **Event Code:** 0x21
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** No credits available to send to Intel UPI on the BL Ring (diff between non-SMI and SMI mode)
- **NOTE:** smi and non-smi modes

Table 2-359. Unit Masks for UPI_PEER_BL_CREDITS_EMPTY

Extension	umask [15:8]	Description
VNA	bxxxxxxx1	VNA
VN0_RSP	bxxxxxx1x	VN0 REQ Messages
VN0_NCS_NCB	bxxxxx1xx	VN0 RSP Messages
VN0_WB	bxxxx1xxx	VN0 SNP Messages
VN1_RSP	bxxx1xxxx	VN1 REQ Messages
VN1_NCS_NCB	bxx1xxxxx	VN1 RSP Messages
VN1_WB	bx1xxxxxx	VN1 SNP Messages

UPI_PREFETCH_SPAWN

- **Title:** FlowQ Generated Prefetch
- **Category:** Special Egress Events
- **Event Code:** 0x29
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Count cases where FlowQ causes spawn of Prefetch to iMC/SMI3 target

VN0_CREDITS_USED

- **Title:** VN0 Credit Used
- **Category:** Link VN Credit Events
- **Event Code:** 0x5b
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a VN0 credit was used on the DRS message channel. In order for a request to be transferred across Intel UPI, it must be guaranteed to have a flit buffer on the remote socket to sink into. There are two credit pools, VNA and VN0. VNA is a shared pool used to achieve high performance. The VN0 pool has reserved entries for each message class and is used to prevent deadlock. Requests first attempt to acquire a VNA credit, and then fall back to VN0 if they fail. This counts the number of times a VN0 credit was used. Note that a single VN0 credit holds access to potentially multiple flit buffers. For example, a transfer that uses VNA could use 9 flit buffers and in that case uses 9 credits. A transfer on VN0 will only count a single credit even though it may use multiple buffers.

Table 2-360. Unit Masks for VN0_CREDITS_USED (Sheet 1 of 2)

Extension	umask [15:8]	Description
REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
WB	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).

Table 2-360. Unit Masks for VN0_CREDITS_USED (Sheet 2 of 2)

Extension	umask [15:8]	Description
NCB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
NCS	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.

VN0_NO_CREDITS

- **Title:** VN0 No Credits
- **Category:** Link VN Credit Events
- **Event Code:** 0x5d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of Cycles there were no VN0 Credits

Table 2-361. Unit Masks for VN0_NO_CREDITS

Extension	umask [15:8]	Description
REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
WB	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
NCB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
NCS	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.

VN1_CREDITS_USED

- **Title:** VN1 Credit Used
- **Category:** Link VN Credit Events
- **Event Code:** 0x5c
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of times a VN1 credit was used on the WB message channel. In order for a request to be transferred across QPI, it must be guaranteed to have a flit buffer on the remote socket to sink into. There are two credit pools, VNA and VN1. VNA is a shared pool used to achieve high performance. The VN1 pool has reserved entries for each message class and is used to prevent deadlock. Requests first attempt to acquire a VNA credit, and then fall back to VN1 if they fail. This counts the number of times a VN1 credit was used. Note that a single VN1 credit holds

access to potentially multiple flit buffers. For example, a transfer that uses VNA could use 9 flit buffers and in that case uses 9 credits. A transfer on VN1 will only count a single credit even though it may use multiple buffers.

Table 2-362. Unit Masks for VN1_CREDITS_USED

Extension	umask [15:8]	Description
REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
WB	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
NCB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
NCS	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.

VN1_NO_CREDITS

- **Title:** VN1 No Credits
- **Category:** Link VN Credit Events
- **Event Code:** 0x5e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:** Number of Cycles there were no VN1 Credits

Table 2-363. Unit Masks for VN1_NO_CREDITS

Extension	umask [15:8]	Description
REQ	bxxxxxxx1	REQ on AD Home (REQ) messages on AD. REQ is generally used to send requests, request responses, and snoop responses.
SNP	bxxxxxx1x	SNP on AD Snoops (SNP) messages on AD. SNP is used for outgoing snoops.
RSP	bxxxxx1xx	RSP on AD Response (RSP) messages on AD. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
WB	bxxxx1xxx	RSP on BL Response (RSP) messages on BL. RSP packets are used to transmit a variety of protocol flits including grants and completions (CMP).
NCB	bxxx1xxxx	WB on BL Data Response (WB) messages on BL. WB is generally used to transmit data with coherency. For example, remote reads and writes, or cache to cache transfers will transmit their data using WB.
NCS	bxx1xxxxx	NCB on BL Non-Coherent Broadcast (NCB) messages on BL. NCB is generally used to transmit data without coherency. For example, non-coherent read data returns.

WB_OCC_COMPARE

- **Title:**
- **Category:** Writeback Events
- **Event Code:** 0x7e
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-364. Unit Masks for WB_OCC_COMPARE

Extension	umask [15:8]	Description
RT_GT_LOCALDEST_VN0	b0xxxxxx1	
RT_EQ_LOCALDEST_VN0	b0xxxxx1x	
RT_LT_LOCALDEST_VN0	b0xxxx1xx	
RT_GT_LOCALDEST_VN1	b0xx1xxxx	
RT_EQ_LOCALDEST_VN1	b0x1xxxxx	
RT_LT_LOCALDEST_VN1	b01xxxxxx	
BOTHNONZERO_RT_GT_LOCALDEST_VN0	b1xxxxxx1	
BOTHNONZERO_RT_EQ_LOCALDEST_VN0	b1xxxxx1x	
BOTHNONZERO_RT_LT_LOCALDEST_VN0	b1xxxx1xx	
BOTHNONZERO_RT_GT_LOCALDEST_VN1	b1xx1xxxx	
BOTHNONZERO_RT_EQ_LOCALDEST_VN1	b1x1xxxxx	
BOTHNONZERO_RT_LT_LOCALDEST_VN1	b11xxxxxx	

WB_PENDING

- **Title:**
- **Category:** Writeback Events
- **Event Code:** 0x7d
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-365. Unit Masks for WB_PENDING

Extension	umask [15:8]	Description
LOCALDEST_VN0	bxxxxxxx1	
ROUTETHRU_VN0	bxxxxxx1x	
LOCAL_AND_RT_VN0	bxxxxx1xx	
WAITING4PULL_VN0	bxxxx1xxx	
LOCALDEST_VN1	bxxx1xxxx	
ROUTETHRU_VN1	bxx1xxxxx	

Table 2-365. Unit Masks for WB_PENDING

Extension	umask [15:8]	Description
LOCAL_AND_RT_VN1	bx1xxxxxx	
WAITING4PULL_VN1	b1xxxxxxx	

XPT_PFTCH

- **Title:**
- **Category:** XPT Events
- **Event Code:** 0x61
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

Table 2-366. Unit Masks for XPT_PFTCH

Extension	umask [15:8]	Description
ARRIVED	bxxxxxxx1	xpt prefetch message arrived in ingress pipeline
BYPASS	bxxxxxx1x	xpt prefetch message took bypass path
ARB	bxxxxx1xx	xpt prefetch message is making arbitration request
LOST_ARB	bxxxx1xxx	xpt prefetch message lost arbitration
FLITTED	bxxx1xxxx	xpt prefetch message was slotted into flit (non bypass)
LOST_OLD	bxx1xxxxx	xpt prefetch message was dropped because it became too old
LOST_QFULL	bx1xxxxx	xpt prefetch message was dropped because it was overwritten by new message while prefetch queue was full

2.11 PCIe3/DMI Performance Monitoring

The 3rd Gen Intel Xeon Processor Scalable Family integrated PCIe3/DMI root port provides the interface between the component plugged into the PCIe3 / DMI port and the components (in M2IOSF) responsible for translating and managing requests to/ from the device.

2.11.1 PCIe3 Performance Monitoring Overview

Each PCIe3 Box supports event monitoring through four 48b wide counters (PI3n_PCI_PMON_CTR/CTL{3:0}). Each of these four counters can be programmed to count almost any PCIe3 event (see NOTE for exceptions). the PCIe3 counters can increment by a maximum of ??b per cycle.

Note: Only counter 0 can be used for tracking occupancy events. .

2.11.2 PCIe 3.0 Performance Monitoring Events

PCIe 3.0 provides events to capture Link information such as Utilization.

2.11.3 PCIe 3.0 Box Events Ordered By Code

The following table summarizes the directly measured PCIe 3.0 Box events.

Symbol Name	Event Code	Ctrs	Max Inc/ Cyc	Description
UTIL_IN	0x16	0-3	1	Cycles of Inbound Link Utilization
UTIL_OUT	0x17	0-3	1	Cycles of Outbound Link Utilization
LINK_CYCLES	0x18	0-3	1	Cycles a Link is in a power state or busy/idle
LINK_RETRIES	0x1e	0-3	1	Number of Link Retries
CORR_ERR	0x1f	0-3	1	Number of Correctable Errors

2.11.4 PCIe 3.0 Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the PCIe 3.0 Box.

CORR_ERR

- **Title:** Number of Correctable Errors
- **Category:** Misc Events
- **Event Code:** 0x1f
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

LINK_CYCLES

- **Title:** Cycles a Link is in a power state or busy/idle
- **Category:** Link Cycle Events
- **Event Code:** 0x18
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

Table 2-367. Unit Masks for LINK_CYCLES (Sheet 1 of 2)

Extension	umask [15:8]	Description
L0_BUSY.PORT2	bxxxxxxx1	Cycles in L0 where the Link is Busy on Port2
L0_BUSY.PORT3	bxxxxxxx1	Cycles in L0 where the Link is Busy on Port3
L0_BUSY.PORT0	bxxxxxxx1	Cycles in L0 where the Link is Busy on Port0
L0_BUSY.PORT1	bxxxxxxx1	Cycles in L0 where the Link is Busy on Port1
L1.PORT2	bxxxxxx1x	Cycles in L1 on Port2
L1.PORT1	bxxxxxx1x	Cycles in L1 on Port1
L1.PORT0	bxxxxxx1x	Cycles in L1 on Port0
L1.PORT3	bxxxxxx1x	Cycles in L1 on Port3
L0_IDLE.PORT2	bxxxxx1xx	Cycles in L0 where the Link is Idle on Port2
L0_IDLE.PORT1	bxxxxx1xx	Cycles in L0 where the Link is Idle on Port1

Table 2-367. Unit Masks for LINK_CYCLES (Sheet 2 of 2)

Extension	umask [15:8]	Description
L0_IDLE.PORT3	bxxxxx1xx	Cycles in L0 where the Link is Idle on Port3
L0_IDLE.PORT0	bxxxxx1xx	Cycles in L0 where the Link is Idle on Port0

LINK_RETRIES

- **Title:** Number of Link Retries
- **Category:** Misc Events
- **Event Code:** 0x1e
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

UTIL_IN

- **Title:** Cycles of Inbound Link Utilization
- **Category:** Utilization Events
- **Event Code:** 0x16
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Cycles the Inbound Link is Utilized. Utilized is whenever the IP was not idle

Table 2-368. Unit Masks for UTIL_IN

Extension	umask [15:8]	Description
PORT0	bxxxxxxxx	Port0
PORT3	bxxxxxxxx	Port3
PORT2	bxxxxxxxx	Port2
PORT1	bxxxxxxxx	Port1

UTIL_OUT

- **Title:** Cycles of Outbound Link Utilization
- **Category:** Utilization Events
- **Event Code:** 0x17
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of Cycles the Outbound Link is Utilized. Utilized is whenever the IP was not idle

Table 2-369. Unit Masks for UTIL_OUT

Extension	umask [15:8]	Description
PORT2	bxxxxxxxx	Port2
PORT1	bxxxxxxxx	Port1
PORT0	bxxxxxxxx	Port0
PORT3	bxxxxxxxx	Port3

2.12 Power Control (PCU) Performance Monitoring

The PCU is the primary Power Controller for the 3rd Gen Intel Xeon Processor Scalable Family die, responsible for distributing power to core/uncore components and thermal management. It runs in firmware on an internal micro-controller and coordinates the socket's power states.

The PCU algorithmically governs the P-state of the processor, C-state of the core and the package C-state of the socket. It enables the core to go to a higher performance state ("turbo mode") when the proper set of conditions are met. Conversely, the PCU will throttle the processor to a lower performance state when a thermal violation occurs.

Through specific events, the OS and the PCU will either promote or demote the C-State of each core by altering the voltage and frequency. The system power state (S-state) of all the sockets in the system is managed by the server legacy bridge in coordination with all socket PCUs.

The PCU communicates to all the other units through multiple PMLink interfaces on-die and Message Channel to access their registers. The OS and BIOS communicates to the PCU thru standardized MSR registers and ACPI.

The PCU also acts as the interface to external management controllers via PECI and voltage regulators (NPTM). The DMI2 interface is the communication path from the southbridge for system power management.

Note: Power management is not completely centralized. Many units employ their own power saving features. Events that provide information about those features are captured in the PMON bocks of those units. For example, Intel® UPI Link Power saving states and Memory CKE statistics are captured in the Intel® UPI Perfmon and IMC Perfmon respectively.

2.12.1 PCU Performance Monitoring Overview

The uncore PCU supports event monitoring through four 48-bit wide counters (PCU_MSR_PMON_CTR{3:0}). Each of these counters can be programmed (PCU_MSR_PMON_CTL{3:0}) to monitor any PCU event. The PCU counters can increment by a maximum of 5b per cycle.

Four extra 64-bit counters are provided by the PCU to track P and C-State Residence. Although documented in this manual for reference, these counters exist outside of the PMON infrastructure.

2.12.2 Additional PCU Performance Monitoring

2.12.2.1 PCU PMON Counter Control - Difference from Baseline

The following table defines the difference in the layout of the PCU performance monitor control registers from the baseline presented in Chapter 1, "Introduction."

Since much of the PCU's functionality is provided by an embedded microcontroller, many of the available events are generated by the microcontroller and handed off to the hardware for PMON capture. Among the events generated by the microcontroller

are occupancy events allowing a user to measure the number of cores in a given C-state per-cycle. Given this unique situation, extra control bits are provided to filter the output of the these special occupancy events.

- *.occ_invert* - Changes the *.thresh* test condition to '<' for the occupancy events (when .ev_sel[7] is set to 1)

- *.occ_edge_det* - Rather than accumulating the raw count each cycle (for events that can increment by 1 per cycle), the register can capture transitions from no event to an event incoming for the PCU's occupancy events (when .ev_sel[7] is set to 1).

Figure 2-13. PCU Counter Control Register for 3rd Gen Intel Xeon Processor Scalable Family

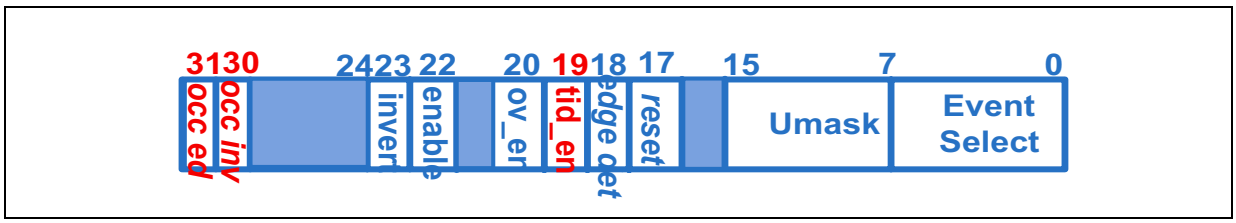


Table 2-370. PCU_MSR_PMON_CTL{3-0} Difference from Baseline – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
occ_edge_det	31	RW-V	0	Enables edge detect for occupancy events (.ev_sel[7] is 1) When set to 1, rather than measuring the event in each cycle it is active, the corresponding counter will increment when a 0 to 1 transition (i.e. rising edge) is detected. When 0, the counter will increment in each cycle that the event is asserted. NOTE: .edge_det is in series following .thresh, Due to this, the .thresh field must be set to a non-0 value. For events that increment by no more than 1 per cycle, set .thresh to 0x1.
occ_invert	30	RW-V	0	Invert comparison against Threshold for the PCU Occupancy events (.ev_sel[7] is 1) 0 - comparison will be 'is event increment >= threshold?'. 1 - comparison is inverted - 'is event increment < threshold?' NOTE: .invert is in series following .thresh, Due to this, the .thresh field must be set to a non-0 value. For events that increment by no more than 1 per cycle, set .thresh to 0x1. Also, if .edge_det is set to 1, the counter will increment when a 1 to 0 transition (i.e. falling edge) is detected.

Context sensitive filtering is provided for through the PCU_MSR_PMON_BOX_FILTER register.

- For frequency/voltage band filters, the multiplier is at 100MHz granularity. So, a value of 32 (0x20) would represent a frequency of 3.2GHz.
- Support for limited Frequency/Voltage Band histogramming. Each of the four bands provided for in the filter may be simultaneous tracked by the corresponding event

Note: Since use of the register as a filter is heavily overloaded, simultaneous application of this filter to additional events in the same run is severely limited

Table 2-371. Additional PCU Performance Monitoring Registers (MSR)

MSR Name	MSR Address	Size (bits)	Description
Fixed (Non-PMON) Counters			
PCU_MSR_CORE_P6_CTR	0x03F9	64	Fixed P-State Residency Counter
PCU_MSR_CORE_P3_CTR	0x03F8	64	Fixed P-State Residency Counter
PCU_MSR_CORE_C6_CTR	0x03FD	64	Fixed C-State Residency Counter
PCU_MSR_CORE_C3_CTR	0x03FC	64	Fixed C-State Residency Counter

Note: Address to the PCU specific filtering register can be found in Chapter 1. But there are some additional pieces of state of relevance to performance monitoring uses.

Table 2-372. PCU_MSR_PMON_BOX_FILTER Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
rsv	63:48	RV	0	Reserved
filt31_24	31:24	RW-V	0	Band 3 - For Voltage/Frequency Band Event
filt23_16	23:16	RW-V	0	Band 2 - For Voltage/Frequency Band Event
filt15_8	15:8	RW-V	0	Band 1 - For Voltage/Frequency Band Event
filt7_0	7:0	RW-V	0	Band 0 - For Voltage/Frequency Band Event

The PCU includes two extra MSRs that track the number of reference cycles a core (any core) is in either the C3 or C6 state. And the PCU includes two extra MSRs that track the number of reference cycles the package is in either the C3 or C6 state. As mentioned before, these counters are not part of the PMON infrastructure so they can't be frozen or reset with the otherwise controlled by the PCU PMON control registers.

Note: To be clear, these counters track the number of cycles **some** core is in C3/6 state. It does not track the total number of cores in the C3/6 state in any cycle. For that, a user should refer to the regular PCU event list.

Table 2-373. PCU_MSR_CORE_{C3,C6,P3,P6}_CTR Register – Field Definitions

Field	Bits	Attr	HW Reset Val	Description
event_count	63:0	RW-V	0	64-bit performance event counter

2.12.3 PCU Performance Monitoring Events

The PCU provides the ability to capture information covering a wide range of the PCU's functionality including:

- Number of cores in a given C-state per-cycle
- Core State Transitions - there are a larger number of events provided to track when cores transition C-state, when the enter/exit specific C-states, when they receive a C-state demotion, and so forth.

- Package State Transitions
- Frequency/Voltage Banding - ability to measure the number of cycles the uncore was operating within a frequency or voltage 'band' that can be specified in a separate filter register.

Note: Given the nature of many of the PCU events, a great deal of additional information can be measured by setting the *.edge_det* bit. By doing so, an event such as "Cycles Changing Frequency" becomes "Number of Frequency Transitions.

On Occupancy Events:

Because it is not possible to "sync" the PCU occupancy counters by employing tricks such as bus lock before the events start incrementing, the PCU has provided fixed occupancy counters to track the major queues.

1. Cores in C0 (4 bits)
2. Cores in C3 (4 bits)
3. Cores in C6 (4 bits)

The PCU perfmon implementation/programming is more complicated than many of the other units. As such, it is best to describe how to use them with a couple examples.

- Case 1: Cycles there was a Voltage Transition (Simple Event)
- Case 2: Cores in C0 (Occupancy Accumulation)
- Case 3: Cycles w/ more than 4 cores in C0 (Occupancy Thresholding)
- Case 4: Transitions into more than 4 cores in C0 (Thresholding + Edge Detect)
- Case 5: Cycles a) w/ > 4 Cores in C0 and b) there was a Voltage Transition
- Case 6: Cycles a) w/ <4 Cores in C0 and b) Freq < 2.0 GHz

Table 2-374. PCU Configuration Examples

	Case					
Config	**1**	**2**	**3**	**4**	**5**	**6**
Counter Control 0						
.ev_sel		0x80	0x80	0x80	0x80	0x80
.occ_sel		0x1	0x1	0x1	0x1	0x1
.thresh		0x0	0x5	0x5	0x5	0x4
.invert		0	0	0	0	1
.occ_invert		0	0	0	0	1
.occ_edge_det		0	0	1	0	0
Counter Control 1						
.ev_sel	0x03				0x03	0x0B
Filter	0x00	0x00	0x00	0x00	0x00	0x14

2.12.4 PCU Box Events Ordered By Code

The following table summarizes the directly measured PCU Box events.

Symbol Name	Event Code	Ctrs	Extra Select Bit	Max Inc/ Cyc	Description
CLOCKTICKS	0x00	0-3	0	1	Clockticks of the power control unit (PCU)
FREQ_MAX_LIMIT_THERMAL_CYCLES	0x04	0-3	0	1	Thermal Strongest Upper Limit Cycles
FREQ_MAX_POWER_CYCLES	0x05	0-3	0	1	Power Strongest Upper Limit Cycles
PMAX_THROTTLED_CYCLES	0x06		0	0	
PROCHOT_INTERNAL_CYCLES	0x09	0-3	0	1	Internal Prochot
PROCHOT_EXTERNAL_CYCLES	0x0a	0-3	0	1	External Prochot
PKG_RESIDENCY_C0_CYCLES	0x2a	0-3	0	1	Package C State Residency - C0
PKG_RESIDENCY_C2E_CYCLES	0x2b	0-3	0	1	Package C State Residency - C2E
PKG_RESIDENCY_C3_CYCLES	0x2c	0-3	0	1	Package C State Residency - C3
PKG_RESIDENCY_C6_CYCLES	0x2d	0-3	0	1	Package C State Residency - C6
MEMORY_PHASE_SHEDDING_CYCLES	0x2f	0-3	0	1	Memory Phase Shedding Cycles
DEMOTIONS	0x30	0-3	0	1	
VR_HOT_CYCLES	0x42	0-3	0	1	VR Hot
FREQ_CLIP_AVX256	0x49		0	0	AVX256 Frequency Clipping
FREQ_CLIP_AVX512	0x4a		0	0	AVX512 Frequency Clipping
CORE_TRANSITION_CYCLES	0x60	0-3	0	1	
TOTAL_TRANSITION_CYCLES	0x72	0-3	0	1	Total Core C State Transition Cycles
FREQ_MIN_IO_P_CYCLES	0x73	0-3	0	1	IO P Limit Strongest Lower Limit Cycles
FREQ_TRANS_CYCLES	0x74	0-3	0	1	Cycles spent changing Frequency
FIVR_PS_PS0_CYCLES	0x75	0-3	0	1	Phase Shed 0 Cycles
FIVR_PS_PS1_CYCLES	0x76	0-3	0	1	Phase Shed 1 Cycles
FIVR_PS_PS2_CYCLES	0x77	0-3	0	1	Phase Shed 2 Cycles
FIVR_PS_PS3_CYCLES	0x78	0-3	0	1	Phase Shed 3 Cycles
POWER_STATE_OCCUPANCY	0x80	0-3	0	8	Number of cores in C-State

2.12.5 PCU Box Common Metrics (Derived Events)

The following table summarizes metrics commonly calculated from PCU Box events.

Symbol Name: Definition	Equation
PCT_CYC_FREQ_POWER_LTD: Percentage of Cycles the Max Frequency is limited by power	FREQ_MAX_POWER_CYCLES / CLOCKTICKS

2.12.6 PCU Box Performance Monitor Event List

The section enumerates 3rd Gen Intel Xeon Processor Scalable Family performance monitoring events for the PCU Box.

CLOCKTICKS

- **Title:** Clockticks of the power control unit (PCU)
- **Category:** PCLK Events
- **Event Code:** 0x00
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** The PCU runs off a fixed 1 GHz clock. This event counts the number of pclk cycles measured while the counter was enabled. The pclk, like the Memory Controller's dclk, counts at a constant rate making it a good measure of actual wall time.

CORE_TRANSITION_CYCLES

- **Title:**
- **Category:** CORE_C_STATE_TRANSITION Events
- **Event Code:** 0x60
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

DEMOTIONS

- **Title:**
- **Category:** CORE_C_STATE_TRANSITION Events
- **Event Code:** 0x30
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:**

FIVR_PS_PS0_CYCLES

- **Title:** Phase Shed 0 Cycles
- **Category:** FIVR Events
- **Event Code:** 0x75
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Cycles spent in phase-shedding power state 0

FIVR_PS_PS1_CYCLES

- **Title:** Phase Shed 1 Cycles
- **Category:** FIVR Events
- **Event Code:** 0x76
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Cycles spent in phase-shedding power state 1

FIVR_PS_PS2_CYCLES

- **Title:** Phase Shed 2 Cycles
- **Category:** FIVR Events
- **Event Code:** 0x77
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Cycles spent in phase-shedding power state 2

FIVR_PS_PS3_CYCLES

- **Title:** Phase Shed 3 Cycles
- **Category:** FIVR Events
- **Event Code:** 0x78
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Cycles spent in phase-shedding power state 3

FREQ_CLIP_AVX256

- **Title:** AVX256 Frequency Clipping
- **Category:** Frequency Clipping Events
- **Event Code:** 0x49
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

FREQ_CLIP_AVX512

- **Title:** AVX512 Frequency Clipping
- **Category:** Frequency Clipping Events
- **Event Code:** 0x4a
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

FREQ_MAX_LIMIT_THERMAL_CYCLES

- **Title:** Thermal Strongest Upper Limit Cycles
- **Category:** FREQ_MAX_LIMIT Events
- **Event Code:** 0x04
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of cycles any frequency is reduced due to a thermal limit. Count only if throttling is occurring.

FREQ_MAX_POWER_CYCLES

- **Title:** Power Strongest Upper Limit Cycles
- **Category:** FREQ_MAX_LIMIT Events
- **Event Code:** 0x05
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when power is the upper limit on frequency.

FREQ_MIN_IO_P_CYCLES

- **Title:** IO P Limit Strongest Lower Limit Cycles
- **Category:** FREQ_MIN_LIMIT Events
- **Event Code:** 0x73
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when IO P Limit is preventing us from dropping the frequency lower. This algorithm monitors the needs to the IO subsystem on

both local and remote sockets and will maintain a frequency high enough to maintain good IO BW. This is necessary for when all the IA cores on a socket are idle but a user still would like to maintain high IO Bandwidth.

FREQ_TRANS_CYCLES

- **Title:** Cycles spent changing Frequency
- **Category:** FREQ_TRANS Events
- **Event Code:** 0x74
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the system is changing frequency. This can not be filtered by thread ID. One can also use it with the occupancy counter that monitors number of threads in C0 to estimate the performance impact that frequency transitions had on the system.

MEMORY_PHASE_SHEDDING_CYCLES

- **Title:** Memory Phase Shedding Cycles
- **Category:** MEMORY_PHASE_SHEDDING Events
- **Event Code:** 0x2f
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles that the PCU has triggered memory phase shedding. This is a mode that can be run in the iMC physicals that saves power at the expense of additional latency.
- **NOTE:** Package C1

PKG_RESIDENCY_C0_CYCLES

- **Title:** Package C State Residency - C0
- **Category:** PKG_C_STATE_RESIDENCY Events
- **Event Code:** 0x2a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the package was in C0. This event can be used in conjunction with edge detect to count C0 entrances (or exits using invert). Residency events do not include transition times.

PKG_RESIDENCY_C2E_CYCLES

- **Title:** Package C State Residency - C2E
- **Category:** PKG_C_STATE_RESIDENCY Events
- **Event Code:** 0x2b
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the package was in C2E. This event can be used in conjunction with edge detect to count C2E entrances (or exits using invert). Residency events do not include transition times.

PKG_RESIDENCY_C3_CYCLES

- **Title:** Package C State Residency - C3
- **Category:** PKG_C_STATE_RESIDENCY Events
- **Event Code:** 0x2c
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the package was in C3. This event can be used in conjunction with edge detect to count C3 entrances (or exits using invert). Residency events do not include transition times.

PKG_RESIDENCY_C6_CYCLES

- **Title:** Package C State Residency - C6
- **Category:** PKG_C_STATE_RESIDENCY Events
- **Event Code:** 0x2d
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles when the package was in C6. This event can be used in conjunction with edge detect to count C6 entrances (or exits using invert). Residency events do not include transition times.

PMAX_THROTTLED_CYCLES

- **Title:**
- **Category:** FREQ_MAX_LIMIT Events
- **Event Code:** 0x06
- **Max. Inc/Cyc:.** 0

Register Restrictions:

- **Definition:**

POWER_STATE_OCCUPANCY

- **Title:** Number of cores in C-State
- **Category:** POWER_STATE_OCC Events
- **Event Code:** 0x80
- **Max. Inc/Cyc:.** 8

Register Restrictions: 0-3

- **Definition:** This is an occupancy event that tracks the number of cores that are in the chosen C-State. It can be used by itself to get the average number of cores in that C-state with thresholding to generate histograms, or with other PCU events and occupancy triggering to capture other details.

Table 2-375. Unit Masks for POWER_STATE_OCCUPANCY

Extension	umask [15:8]	Description
CORES_C0	b01000000	C0 and C1
CORES_C3	b10000000	C3
CORES_C6	b11000000	C6 and C7

PROCHOT_EXTERNAL_CYCLES

- **Title:** External Prochot
- **Category:** PROCHOT Events
- **Event Code:** 0x0a
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles that we are in external PROCHOT mode. This mode is triggered when a sensor off the die determines that something off-die (like DRAM) is too hot and must throttle to avoid damaging the chip.

PROCHOT_INTERNAL_CYCLES

- **Title:** Internal Prochot
- **Category:** PROCHOT Events
- **Event Code:** 0x09
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Counts the number of cycles that we are in Internal PROCHOT mode. This mode is triggered when a sensor on the die determines that we are too hot and must throttle to avoid damaging the chip.

TOTAL_TRANSITION_CYCLES

- **Title:** Total Core C State Transition Cycles
- **Category:** CORE_C_STATE_TRANSITION Events
- **Event Code:** 0x72
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of cycles spent performing core C state transitions across all cores.

VR_HOT_CYCLES

- **Title:** VR Hot
- **Category:** VR_HOT Events
- **Event Code:** 0x42
- **Max. Inc/Cyc:.** 1

Register Restrictions: 0-3

- **Definition:** Number of cycles that a CPU SVID VR is hot. Does not cover DRAM VRs

§

3 Reference for PMON Filtering

3.1 Packet Matching Reference(s)

3.1.1 Reference for CHA Packet Matching

In the CHA, the component responsible for managing the last-level cache and maintaining coherency, the performance monitoring infrastructure allows a user to filter IDI packet traffic tracked in the TOR according to certain fields. The Message Class/ Opcode fields have been summarized in the following tables.

Note that the TOR is comprised of different logical queues managing different classes of requests. The opcodes relevant to each logical queue class are presented in separate tables.

The following tables list the IDI opcodes, broken down by queue, that can be matched on with the above filter. The two opcode match fields operate independently and the results are ORed together. It is not possible to measure events filtered by opcode that match in different fields

IDI opcodes relevant to the IRQ (Ingress Request Queue) - From iA Cores

Table 3-1. Opcode Match by IDI Packet Type (relevant to IRQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 1 of 3)

opc Value	Opcode	Defn
0x100	RFO	Demand Data RFO - Full cache line read requests from agent for lines to be cached in any writeable state
0x110	RFO_Pref	RFO Prefetch - Read for Ownership request sent as prefetch from agent
0x101	CRd	Demand Code Read - Full cache-line read requests from core for lines to be cached in S, typically for code
0x105	CRd_UC	Uncacheable Code Read - Full cache-line read request from agent for lines not meant to be cached
0x111	CRd_Pref	Code Read Prefetch - Full cache-line read reqeusts from core lines to be cached in S, typically from code. Treated as prefetch (i.e. can be droppped)
0x102	DRd	Demand Data Read - Full cache-line read requests from core for lines to be cached in S or E, typically for data
0x112	DRd_Pref	Demand Data Read Prefetch - Full cache-line read requests from core for lines to be cached in S or E, typically for data. Treated as prefetch (i.e. can be dropped)
0x104	DRd_Opt	Optimized Demand Data Read - Acts like DRd Prefetch except does not send LLC Miss response
0x114	DRd_Opt_Pref	Optimized Demand Data Read Prefetch - Acts like DRd except does not send LLC Miss response

Table 3-1. Opcode Match by IDI Packet Type (relevant to IRQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 2 of 3)

opc Value	Opcode	Defn
0x106	DRdPTE	Demand Data Read for Page Walks - Full cache-line read requests from core for lines to be cached in S or E, for page walks
0x107	PRd	Partial Reads (UC) - Partial read requests of 0-32B (IIO can be up to 64B). Uncacheable.
0x10C	WCiLF	Streaming Store - Full - Write invalidate for full cache line of write combining stores
0x10D	WCiL	Streaming Store - Write invalidate for write combining stores
0x10E	UCRdF	Uncacheable Reads - Full - Full-line uncachable read requests.
0x10F	WiL	Write Invalidate Line - Partial
0x118	CLFlush	Cacheline Flush - Invalidate cache line. All other agents cache lines must also be invalidated
0x11A	CLFlushOpt	Optimized Cacheline Flush - Invalidate cache line. All other agents cache lines must also be invalidated
0x11C	CLWB	Cacheline Flush WB - Invalidate cache line. All other agents cache lines must also be invalidated. Only writes modified data back to memory leaving cacheline in E if it was modified.
0x11E	PCIRdCur	Read current - Read Current requests from IIO. Used to read data without changing state.
0x13C	CLCleanse	Cacheline Cleanse - Only from IO
0x1A4	WbPushHint	 - Only from IO
0x184	WbMtoI	Request writeback Modified invalidate line - Evict full M-state cache line from core. Guarantees core has no cached copies.
0x185	WbMtoE	Request writeback Modified set to Exclusive - Evict full M-state cache line from core.
0x186	WbEFtoI	Request "clean" (E or F -state line) writeback - Core guarantees it will no longer retain ownership of the line when the write-back completes
0x187	WbEFtoE	Request "clean" (E or F -state line) writeback - Core may retain ownership of the line when the write-back completes.
0x18C	WbStoI	Request writeback. Shared to Invalidate - Clean line is being dropped. Allows snoop filter updates
0x188	ItoM	Request Invalidate Line - Request Exclusive Ownership of cache line. Agent guarantees entire cache line will be modified.
0x1A8	ItoMCacheNear	 - Only from IO
0x18A	SpecItoM	Speculatively Request Invadlidate Line - Request Exclusive Ownership of cache line. If speculation is correct, Agent guarantees entire cache line will be modified.
0x198	LlcPrefRFO	LLC Prefetch RFO - Uncore will first look up the line in the LLC; for a cache hit, the LRU will be updated, on a miss, the RFO will be initiated

Table 3-1. Opcode Match by IDI Packet Type (relevant to IRQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 3 of 3)

opc Value	Opcode	Defn
0x199	LlcPrefCode	LLC Prefetch Code - Uncore will first look up the line in the LLC; for a cache hit, the LRU will be updated, on a miss, the CRd will be initiated
0x19A	LlcPrefData	LLC Prefetch Data - Uncore will first look up the line in the LLC; for a cache hit, the LRU will be updated, on a miss, the DRd will be initiated
0x1D9	IntLog	Interrupt (Logically Addressed)
0x1DA	IntPhy	Interrupt (Physically Addressed)
0x1DB	IntPriUp	Interrupt Priority Update
0x1DE	SplitLock	SplitLock - Request to start split lock sequence
0x1DF	Lock	Lock - Request to start IDI lock sequence

IDI Opcodes relevant to the ISMQ (Ingress Subsequent Message Queue) - Responses to Ingress Requests

Table 3-2. Opcode Match by IDI Packet Type (relevant to ISMQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 1 of 2)

opc Value	Opcode	Defn
0x000	RspI	Response I - Cache is in I
0x001	RspS	Response S - Cache is in S
0x026	RspV	Response V - Cache state unknown - For cases cache state does not need to be known.
0x002	RspDataM	Response Data M
0x003	RspIFwdM	Response I Forward M
0x033	RspIFwdMPtl	Response I Forward M Partial - Only from IO
0x004	PullData	Pull Data
0x034	PullDataPtl	Pull Data Partial - Only from IO
0x005	PullDataBogus	Pull Data Bogus
0x006	Cmp	Completion - Only from Intel UPI
0x007	CmpFwdCode	Completion Forward Code - Only from Intel UPI
0x008	CmpFwdInvItoE	Completion Forward Invalidate I to E - Only from Intel UPI
0x009	CmpPullData	Completion Pull Data - Only from Intel UPI
0x00B	CmpFwdInvOwn	Completion Forward Invalidate Own - Only from Intel UPI
0x00C	DataC_Cmp	Data Coherent Completion

Table 3-2. Opcode Match by IDI Packet Type (relevant to ISMQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 2 of 2)

opc Value	Opcode	Defn
0x01B	Victim	Victim - Only generated by CHA
0x01E	DataNc	Data NonCoherent - Only from Intel UPI
0x020	DataC	Data Complete - Only from Intel UPI
0x023	RspIFwdFE	Response S Forward F or E
0x024	RspSFwdFE	Response S Forward F or E
0x025	FwdCnflt	Forward Conflict
0x031	LLCVictim	LLC Victim - Only generated by CHA

IDI opcodes relevant to the IPQ (Ingress Probe Queue) - Remote Socket Snoops

Table 3-3. Opcode Match by IDI Packet Type (relevant to IPQ) for Cn_MSR_PMON_BOX_FILTER.opc

opc Value	Opcode	Defn
0x700	SnpCur	Snoop Current - Snoop to get uncacheable 'snapshot' of data
0x701	SnpCode	Snoop Code - Snoop requests from the uncore for lines intended to be cached in S at requester
0x702	SnpData	Snoop Data - Snoop requests from the uncore for lines intended to be cached in S or E state at the requester (the E state can be cached at requester if all cores respond with RspI)
0x703	SnpDataMig	Snoop Data Migratory - Snoop to get data in M, E, or S
0x704	SnpInvOwn	Snoop Invalidate Own - Snoop Invalidate Own - get data in M or E
0x705	SnpInvItoE	Snoop Invalidate - Snoop requests from the uncore for lines intended to be cached in E state at the requester

IDI opcodes relevant to the RRQ (Remote Request Queue) - Read Requests from Remote Socket

Table 3-4. Opcode Match by IDI Packet Type (relevant to RRQ) for Cn_MSR_PMON_BOX_FILTER.opc

opc Value	Opcode	Defn
0x500	RdCur	Read Current - Request cache line in I. Typically issued by I/O proxy entities, RdCur is used to obtain a coherent snapshot of an uncached line
0x501	RdCode	Read Code - Read cache line in S
0x502	RdData	Read Data - Request cache line in either E or S. The choice between S and E is determined by whether or not per caching agent has cache line in S state
0x503	RdDataMig	Read Data Migratory - Same as RdData, except that peer cache can forward requested cache line in M state without any writeback to memory.
0x504	InvOwn	Read Invalidate Own - Read Invalidate Own requests a cache line in M or E state. M or E is determined by whether requester is forwarded an M copy by a peer caching agent or sent an E copy by home agent
0x505	InvXtoI	Invalidate X to I -
0x087	InvItoE	Invalidate I to E -
0x50C	RdInv	Read Invalidate - Request cache line in E from the home agent; any modified copy is committed to memory before receiving the data.
0x050	InvItoM	Invalidate I to M -

IDI opcodes relevant to the WBQ (Writeback Queue) - Write Requests from Remote Sockets

Table 3-5. Opcode Match by IDI Packet Type (relevant to WBQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 1 of 2)

opc Value	Opcode	Defn
0x400	WbMtoI	Writeback M to I - Evict full M-state cache line from core. Guarantees core has no cached copies. Write a cache line in M state back to memory and invalidate the line in the cache.
0x401	WbMtoS	Writeback M to S - Write a cache line in M state back to memory and transition its state to S.
0x402	WbMtoE	Writeback M to E - Evict full M-state cache line from core. Write a cache line in M state back to memory and transition its state to E
0x403	NonsnpWr	Non-Snoop Write - Write a line to memory.
0x404	WbMtoIPtl	Writeback M to I Partial - Write a cache line in M state back to memory, according to a byte-enable mask, and transition its state to I.

Table 3-5. Opcode Match by IDI Packet Type (relevant to WBQ) for Cn_MSR_PMON_BOX_FILTER.opc (Sheet 2 of 2)

opc Value	Opcode	Defn
0x406	WbMtoEPtl	Writeback M to E Partial - Write a cache line in M state back to memory, according to a byte-enable mask, transition the line to E, and clear the line's mask in the cache.
0x407	NonsnpWrPtl	Non-Snoop Write Partial - Write a line to memory according to byte-enable mask.
0x408	WbPushMtoI	Writeback Push M to I - Push cache line in M state to the HA; HA may push data to a local cache (in M state) or write the data to memory. Transition cache line to I.
0x40B	WbFlush	Writeback Flush - Hint for flushing writes in memory hierarchy. No data is sent with the request.
0x40C	EvctCln	Evict Clean - Notification to home that a cache line in E state was invalidated in the cache
0x40D	NonSnpRd	Non-Snoop Read - Request a read only line (i.e. an uncacheable 'snapshot') from memory.

3.1.2 Reference for Intel® Ultra Path Interconnect (Intel® UPI) LL Packet Matching

In the Intel® UPI Link Layer, the component responsible for transmitting and receiving traffic crossing between sockets in a multi-socket machine, the performance monitoring infrastructure allows a user to filter Intel UPI packet traffic according to certain fields. A couple common fields, the Message Class/Opcode fields, have been summarized in the following tables.

Table 3-6. Intel UPI Interconnect Packet Message Classes

Code	Name	Definition
b0000	REQ	Requests
b0001	SNP	Snoop
b0010	RSP - NoData	Non-Data Responses
b0011	---	
b0100	RSP - Data	Data Response
b0101	WB	Writebacks
b0110	NCB	Non-Coherent Bypass
b0111	NCS	Non-Coherent Standard

Table 3-7. Intel UPI Opcode Match by Message Class (Sheet 1 of 2)

Opc	REQ	SNP	RSP2- NoData	
0000	RdCur	SnpCur	CmpU	
0001	RdCode	SnpCode	P2PCmpU	
0010	RdData	SnpData	RspI	
0011	RdDataMig	SnpDataMig	RspS	
0100	RdInvOwn	SnpInvOwn	RspFwd	

Table 3-7. Intel UPI Opcode Match by Message Class (Sheet 2 of 2)

Opc	REQ	SNP	RSP2- NoData	
0101	InvXtoI	SnpInv	RspFwdI	
0110	---	---	RspFwdS	
0111	InvItoE	---	---	
1000	---	SnpFCur	MirCmpU	
1001	---	SnpFCode	---	
1010	---	SnpFData	RspCnflt	
1011	---	SnpFDataMig	---	
1100	RdInv	SnpFInvOwn	CmpO	
1101	---	SnpFInv	FwdCnfltO	
1110	---	---	---	
1111	InvItoM	---	---	
Opc	**RSP4 - Data**	**WB**	**NCB**	**NCS**
0000	Data_M	WbMtoI	NcWr	NcRd
0001	Data_E	WbMtoS	WcWr	IntAck
0010	Data_SI	WbMtoE	---	---
0011	---	NonSnpWr	---	---
0100	Data_M_CmpO	WbMtoIPtl	---	NcRdPtl
0101	Data_E_CmpO	---	---	NcCfgRd
0110	Data_SI_CmpO	WbMtoEPtl	---	NcLTRd
0111	---	NonSnpWrPtl	---	NcIORd
1000	---	WbPushMtoI	NcMsgB	NcMsgS
1001	---	---	IntLogical	NcCfgWr
1010	RspFwdIWb	---	IntPhysical	NcLTWr
1011	RspFwdSWb	WBFlush	IntPrioUpd	NcIOWr
1100	RspIWb	EvctCln	NcWrPtl	---
1101	RspSWb	NonSnpRd	WcWrPtl	---
1110	---	---	---	---
1111	DebugData	---	NcP2PB	NcP2PS

Note: Opcodes in Italics aren't implemented in Snow Ridge.

Table 3-8. Intel UPI Opcodes (Alphabetical Listing) (Sheet 1 of 4)

Name	Opc	Msg Class	Gen By	Desc
CmpO	1100	RSP2		Completion message with no ordering requirements
CmpU	0000	RSP2		Completion message that must be ordered with forward responses.
DataE	0001	RSP4		Data in E
DataE_CmpO	0101	RSP4		Data in E with an ordered completion response
DataM	0000	RSP4		Data in M
DataM_CmpO	0100	RSP4		Data in M with an ordered completion response
DataSI	0010	RSP4		Depending on request, data in S or uncacheable 'snapshot' of data

Table 3-8. Intel UPI Opcodes (Alphabetical Listing) (Sheet 2 of 4)

Name	Opc	Msg Class	Gen By	Desc
DataSI_CmpO	0110	RSP4		Depending on request, data in S or uncacheable 'snapshot' of data; with an ordered completion response
DebugData	1111	RSP4		Debug Data
EvctCln	1100	SNP		Notification to home that a cache line in E state was invalidated in the cache
FwdCnfltO	1101	WB		Ordered response from home agent to resolve conflict situation and let receiver properly process original snoop request. There is always a pre-allocated resource to sink the FwdCnfltO in the coherence agent.
IntAck	0001	NCS		Interrupt acknowledge to legacy 8259 interrupt controller
IntLogical	1001	NCB		Logical mode interrupt to processor
IntPhysical	1010	NCB		Physical mode interrupt to processor
IntPrioUpd	1011	NCB		Interrupt priority update message to source interrupt agents.
InvItoE	0111	REQ		Invalidate to E state. Requests exclusive ownership of a cache line without receiving data.
InvItoM	1111	REQ		Invalidate to M state. Requests exclusive ownership of a cache line without receiving data and with the intent of performing a writeback soon afterword.
InvXtoI	0101	REQ		Flush a cache line from all caches (that is, downgrade all clean copies to I and cause any dirty copy to be written back to memory). Requesting agent must invalidate the line in its cache before issuing this request
NcCfgRd	0101	NCS		Configuration read from configuration space
NcCfgWr	1001	NCS		Configuration write to configuration space
NcIORd	0111	NCS		Read from legacy I/O space
NcIOWr	1011	NCS		Write to legacy I/O space
NcMsgB	1000	NCB		Non-coherent Message (non-coherent bypass channel)
NcMsgS	1000	NCS		Non-coherent Message (Non-coherent standard channel)
NcP2PB	1111	NCB		Peer-to-peer transaction between I/O entities (non-coherent bypass channel)
NcP2PS	1111	NCS		Peer-to-peer transaction between I/O entities. (Non-coherent standard channel)
NcRd	0000	NCS		Read from non-coherent memory mapped I/O space
NcRdPtl	0100	NCS		Partial read from non-coherent memory mapped I/O space
NcWr	0000	NCB		Write to non-coherent memory mapped I/O space
NcWrPtl	1100	NCB		Partial write to non-coherent memory mapped I/O space
NonSnpRd	1101	WB		Request a read only line (i.e. an uncacheable 'snapshot') from memory.
NonSnpWr	0011	WB		Write a line to memory.
NonSnpWrPtl	0111	WB		Write a line to memory according to byte-enable mask.
P2PCmpU	0001	RSP2		Peer-to-peer codmpletion message that must be ordered with forward responses.

Table 3-8. Intel UPI Opcodes (Alphabetical Listing) (Sheet 3 of 4)

Name	Opc	Msg Class	Gen By	Desc
RdCode	0001	REQ		Read cache line in S
RdCur	0000	REQ		Request cache line in I. Typically issued by I/O proxy entities, RdCur is used to obtain a coherent snapshot of an uncached line.
RdData	0010	REQ		Request cache line in either E or S. The choice between S and E is determined by whether or not per caching agent has cache line in S state.
RdDataMig	0011	REQ		Same as RdData, except that peer cache can forward requested cache line in M state without any writeback to memory.
RdInv	1100	REQ		Request cache line in E from the home agent; any modified copy is committed to memory before receiving the data.
RdInvOwn	0100	REQ		Read Invalidate Own requests a cache line in M or E state. M or E is determined by whether requester is forwarded an M copy by a peer caching agent or sent an E copy by home agent.
RspCnflt	1010	RSP2		Peer has outstanding request to same address, is requesting an ordered forward response, and has allocated a resource for the forward.
RspFwd	0100	RSP2		Copy of cache line was sent to requesting agent, cache state did not change
RspFwdI	0101	RSP2		Copy of cache line was sent to requesting agent, cache state was downgraded to I
RspFwdIWb	1010	RSP4		Modified line is being implicitly written back to memory, a copy of cache line was sent to requesting agent and the line was downgraded to I
RspFwdS	0110	RSP2		Copy of cache line was sent to requesting agent, cache state was downgraded to S
RspFwdSWb	1011	RSP4		Modified line is being implicitly written back to memory, a copy of cache line was sent to requesting agent and the line was downgraded to S
RspI	0010	RSP2		Cache is in I
RspIWb	1100	RSP4		Modified line is being implicitly written back to memory, cache line was downgraded to I
RspS	0011	RSP2		Cache is in S
RspSWb	1101	RSP4		Modified line is being implicitly written back to memory, cache line was downgraded to S
SnpCode	0001	SNP		Snoop Code - get data in S
SnpCur	0000	SNP		Snoop to get uncacheable ‘snapshot’ of data
SnpData	0010	SNP		Snoop Data - get data in E or S
SnpDataMig	0011	SNP		Snoop to get data in M, E, or S
SnpFCode	1001	SNP		Snoop Code - get data in S; Routing layer will handle distribution to all fanout peers
SnpFCur	1000	SNP		Snoop to get uncacheable ‘snapshot’ of data; Routing layer will handle distribution to all fanout peers
SnpFData	1010	SNP		Snoop Data - get data in E or S; Routing layer will handle distribution to all fanout peers
SnpFDataMig	1011	SNP		Snoop to get data in M, E, or S; Routing layer will handle distribution to all fanout peers
SnpFInv	1101	SNP		Snoop to invalidate peer’s cache, flushing any M copy to memory; Routing layer will handle distribution to all fanout peers

Table 3-8. Intel UPI Opcodes (Alphabetical Listing) (Sheet 4 of 4)

Name	Opc	Msg Class	Gen By	Desc
SnpFInvOwn	1100	SNP		Snoop Invalidate Own - get data in M or E; Routing layer will handle distribution to all fanout peers
SnpInv	0101	SNP		Snoop to invalidate peer's cache, flushing any M copy to memory
SnpInvOwn	0100	SNP		Snoop Invalidate Own - get data in M or E
SnpInvXtoI	1100	SNP		Snoop Invalidate Writeback M to I state. To invalidate peer caching agent, flushing any M state data to home.
WBFlush	1011	WB		Hint for flushing writes in memory hierarchy. No data is sent with the request.
WbMtoE	0010	WB		Write a cache line in M state back to memory and transition its state to E.
WbMtoEPtl	0110	WB		Write a cache line in M state back to memory, according to a byte-enable mask, transition the line to E, and clear the line's mask in the cache.
WbMtoI	0000	WB		Write a cache line in M state back to memory and invalidate the line in the cache.
WbMtoIPtl	0100	WB		Write a cache line in M state back to memory, according to a byte-enable mask, and transition its state to I.
WbMtoS	0001	WB		Write a cache line in M state back to memory and transition its state to S.
WbPushMtoI	1000	WB		Push cache line in M state to the HA; HA may push data to a local cache (in M state) or write the data to memory. Transition cache line to I.
WcWr	0001	NCB		Write combinable write to non-coherent memory mapped I/O space
WcWrPtl	1101	NCB		Partial write combinable write to non-coherent memory mapped I/O space

3.1.3 Reference for M2M Packet Matching

In M2M, the component that connects the Memory Controller to the Mesh, the performance monitoring infrastructure allows a user to filter SMI3 packet traffic according to certain fields. The Message Class/Opcode fields have been summarized in the following tables.

Table 3-9. SMI3 Opcode Match by Message Class (Sheet 1 of 2)

Opc	REQ	SNP	RSP2- NoData	
0000	MemRd	---	CmpU	
0001	MemSpecRd	---	---	
0010	MemRdData	---	---	
0011	---	---	---	
0100	MemRdXtoS	---	---	
0101	MemRdXtoI	---	---	
0110	MemRdXtoA	---	---	
0111	---	---	---	
1000	MemInv	---	---	

Table 3-9. SMI3 Opcode Match by Message Class (Sheet 2 of 2)

Opc	REQ	SNP	RSP2- NoData	
1001	MemInvXtoI	---	---	
1010	MemInvXtoA	---	---	
1011	---	---	---	
1100	MemInvItoX	---	---	
1101	---	---	---	
1110	---	---	---	
1111	---	---	MultCmpUd	
Opc	**RSP4 - Data**	**WB**	**NCB**	**NCS**
0000	---	MemWr	---	---
0001	---	---	---	---
0010	---	---	---	---
0011	---	MemWrNI	---	---
0100	---	MemWrPtl	---	---
0101	---	---	---	---
0110	MemData	---	---	---
0111	---	MemWrPtlNI	---	---
1000	---	---	---	---
1001	---	---	---	---
1010	---	---	---	---
1011	---	MemWrFlush	---	---
1100	---	---	---	---
1101	---	---	---	---
1110	---	---	---	---
1111	---	---	---	---

Note: Opcodes in Italics aren't implemented in Snow Ridge.

Table 3-10. SMI3 Opcodes (Alphabetical Listing) (Sheet 1 of 2)

Name	Opc	Msg Class	Gen By	Desc
CmpU	0000	RSP2		Completion for MemWr, MemWrNI, MemWrPtl, MemWrPtlNI and all MemInv* commands. For write completions, it is sent when the write becomes globally visible.
MemRd	0000	REQ		Normal memory read. Directory information is left unmodified.
MemSpecRd	0001	REQ		Speculative Read. This is a hint to the memory controller that a read to this address will likely occur. It allows the memory controller to get a head start on the read to reduce average latency. If the demand read is received by the time the data comes back. It will be used. Otherwise the data may be dropped.
MemRdData	0010	REQ		Memory Read, rewrite with directory (if necessary) as follows: I to A S has no change A has no change. If a change is required from A, the host must specifically change it.

Table 3-10. SMI3 Opcodes (Alphabetical Listing) (Sheet 2 of 2)

Name	Opc	Msg Class	Gen By	Desc
MemRdXtoS	0100	REQ		Memory read, directory result is Shared. If the directory indicates anything other than S, it is written back as S
MemRdXtoI	0101	REQ		Memory read, Directory result is Invalid. If the directory indicates anything other than I, it is written back as I
MemRdXtoA	0110	REQ		Memory read, Directory result is Any. If the directory indicates anything other than A, it is written back as A
MemInv	1000	REQ		Memory Invalidate. The command reads and returns the directory state. No directory update is done.
MemInvXtoI	1001	REQ		Memory Invalidate. Directory result is Invalid. No read data is returned. The memory controller reads the cache line and rewrites it with the directory set to I if not already that way.
MemInvXtoA	1010	REQ		Memory Invalidate. Directory result is Any. No read data is returned. The memory controller reads the cache line and rewrites it with the directory set to A if not already that way.
MemInvItoX	1100	REQ		Memory controller reads Near Memory. Tag is not updated in Intel NM regardless of hit or miss If it an Intel NM miss: - No Far Memory read is performed - if Intel NM was Modified, then the data is written to Far Memory since a later write to a different address within this set may overwrite it. If it is an Intel NM hit: - No change to Intel NM or FM
MultCmpU	1111	RSP2		Multiple Completion for MemWr, MemWrNI, MemWrPtl, MemWrPtlNI and all MemInv* commands. Up to 8 completions can be sent in this flit.
MemData	0110	RSP4		Memory Read Data. Returned in response to all MemRd commands (but not MemSpecRd). These commands may also use the headerless data return, however there are cases where the headered version must be used.
MemWr	0000	WB		Memory Write. Used for full line writes in 1LM and for Inclusive full line writes in 2LM
MemWrNI	0011	WB		Memory Write Non-Inclusive. Write checks Intel NM tag and if it is a hit in Intel NM, then updates data/ directory Intel NM. If a miss in Intel NM then writes to FM only.
MemWrPtl	0100	WB		Memory Write Partial. Used for partial line writes in 1LM and for Inclusive full line writes in 2LM
MemWrPtlNI	0011	WB		Memory Write Partial Non-Inclusive. Write checks Intel NM tag and if it is a hit in Intel NM, then updates data/directory Intel NM. If a miss in Intel NM then writes to FM only. Not used for 1LM.
MemWrFlush	1011	WB		Memory Write Flush. This command is sent as a result of a PCommit (Persistent Commit) in the host. The memory controller should flush all writes to persistent memory before returning the completion.

§