Overview of Gemini Hardware Counters

This document describes the Gemini Performance Counters and how to use them to optimize individual applications and system traffic.

Send e-mail to docs@cray.com with any comments that will help us to improve the accuracy and usability of this document. Be sure to include the title and number of the document with your comments. We value your comments and will respond to them promptly.

Accessing network performance counters is desirable for application developers, system library developers (e.g. MPI), and system administrators. Application developers want to improve their application run-times or measure what affect other traffic on the system has on their application. System library developers want to optimize their collective operations. System Administrators want to observe the system, looking for hotspots. Effective with the CrayPat (Cray performance analysis tool) version 5.1 and Cray Linux Environment (CLE) version 3.1 software releases for the Cray XE platform, users can monitor many of the performance counters that reside on the Gemini networking chip.

There are two categories of Gemini performance counters available to users. NIC performance counters record information about the data moving through theNetwork Interface Controller (NIC). On the Gemini ASIC there are two NICs, each attached to a compute node. Thus, the data from the NIC performance counters reflects network transfers beginning and ending on the node. These performance counters are read-only.

Network router tile counters are available on a per-Gemini basis. There are both read-only and read/write tile counters. Each chip has 48 router tiles, arranged in a 6x8 grid. Eight processor tiles connect to each of the two Gemini NICs. Each NIC connects to a different node, running separate Linux instances.

If collection at other points of the application is desired, use the CrayPat API to insert regions as described in the `pat_build` man page. It is recommended that you do not collect any other performance data when collecting network counters. Data collection of network counters is much more expensive than other performance data collection, and will skew other results. At the time the instrumented executable program is launched with the `aprun` command, a set of environment variables, `PAT_RT_NWPC_*`, provide access to the Gemini network performance counters. These environment variables are described in the `intro_craypat` man page.

1.1 Using CrayPat to Monitor Gemini Counters

The CrayPat utility `pat_build` instruments an executable file. One aspect of the instrumentation includes intercepting entries into and returns out of a function. This is known formally as *tracing*. Information such as time stamps and performance counter values are recorded at this time.

CrayPat supports instrumentation of an application binary for collection of Gemini counters. Counter values are recorded at application runtime, and are presented to the user through a table generated by pat_report. The CrayPat user interface to request instrumentation is similar to that for processor performance counters. There is no Gemini counter display available in Cray Apprentice2 at this time. A new display will be available in a subsequent release of the Cray Apprentice2 software.

Although the user interface to request network counters is similar to processor counters, there are some significant differences that must be understood. Depending on the type of counters requested, some are shared across all processors within a node, some are shared between two nodes and some are shared across all applications passing through a chip. Some counters monitor all traffic for your application, even on nodes that are not reserved for your application, and some monitor locally, that is they monitor only traffic associated with nodes assigned to a Gemini chip and no other traffic from the network.

Users should also be aware that access to the network counters is more resource-intensive than access to the processor performance counters. Because Gemini counters are a shared resource, the system software is designed to provide dedicated access whenever possible. This is done through the Application Level Placement Scheduler (ALPS) by ensuring that an application collecting counters is not placed on the same Gemini chip as another application collecting performance counters. It does not prevent a second application from being placed on the same Gemini chip that is not collecting counters however. This compromise assures better system utilization because compute nodes are not left unavailable for use by another application.

The CrayPat 5.1 release focuses on the use of the NIC and ORB counters available within the Gemini chip. The values collected from these counters are local to a node and therefore specific to an application. Traffic between MPI ranks cannot be distinguished through the counters. The event names that CrayPat supports are listed at the end of this document. Network counters are only collected for the MAIN thread. Values are collected at the beginning and end of the instrumented application. Instrumentation overhead is minimal. This gives a high-level view of the program's use of the networking router in terms of the counters specified. Currently the time to access counter data is too expensive to collect more frequently. A future release of CLE will address these performance limitations.

Before attempting the following examples verify that your system has a Gemini network:

```
$ module list
 xtpe-network-gemini
```

Attempting to collect Gemini performance counters on a system that does not have the Gemini network will result in a fatal error:

```
$ aprun -n 16 my_program+pat
CrayPat/X:  Version 5.1 Revision 3329  05/20/10 11:26:16
pat[FATAL][0]: initialization of NW performance counter API failed
[No such file or directory]
```

Example 1. Collect stalls associated with node traffic to and from the network

This example enables tracing of MAIN.

```
$ pat_build -w my_program
$ export PAT_RT_NWPC=GM_ORB_PERF_VC0_STALLED,GM_ORB_PERF_VC1_STALLED
$ aprun my_program+pat
```

Example 2. Display network counter data

```
$ pat_report my_program+pat+11171-41tdot.xf> counter_rpt
```

Example output from `pat_report`:

```
NWPC Data by Function Group and Function    Group / Function / Node Id=0='HIDE'
=============================================================================
Total
-----------------------------------------------------------------------------
Time%                         100.0%
Time                        2.476423 secs
GM_ORB_PERF_VC1_STALLED            0
GM_ORB_PERF_VC1_BLOCKED            0
GM_ORB_PERF_VC1_BLOCKED_PKT_GEN    0
GM_ORB_PERF_VC1_PKTS              48
GM_ORB_PERF_VC1_FLITS             48
GM_ORB_PERF_VC0_STALLED          111
GM_ORB_PERF_VC0_PKTS              48
GM_ORB_PERF_VC0_FLITS            201
=============================================================================
```

Example 3. Collect data for a custom group of network counters

In this example a user creates a group of network events in a file called *my_nwpc_groups*, one called `1` and the other called `CQ_AMO`:

```
$ cat my_nwpc_groups
# Group 1: Outstanding Request Buffer
1 =
GM_ORB_PERF_VC1_STALLED,
GM_ORB_PERF_VC1_BLOCKED,
GM_ORB_PERF_VC1_BLOCKED_PKT_GEN,
GM_ORB_PERF_VC1_PKTS,
GM_ORB_PERF_VC1_FLITS,
GM_ORB_PERF_VC0_STALLED,
GM_ORB_PERF_VC0_PKTS,
GM_ORB_PERF_VC0_FLITS

# Group CQ_AMO:
CQ_AMO =
GM_AMO_PERF_COUNTER_EN,
GM_AMO_PERF_CQ_FLIT_CNTR,
GM_AMO_PERF_CQ_PKT_CNTR,
GM_AMO_PERF_CQ_STALLED_CNTR,
GM_AMO_PERF_CQ_BLOCKED_CNTR

$ pat_build -w my_program
$ export PAT_RT_NWPC_FILE=my_nwpc_groups
$ export PAT_RT_NWPC=1,CQ_AMO
$ aprun -n16 my_program+pat
```

Example output from `pat_report`:

```
NWPC Data by Function Group and Function
Group / Function / Node Id=0='HIDE'
=========================================================================
Total
 -------------------------------------------------------------------------
Time%                              100.0%
Time                             2.639046 secs
GM_ORB_PERF_VC1_STALLED             72525
GM_ORB_PERF_VC1_PKTS                50457
GM_AMO_PERF_COUNTER_EN                  0
GM_AMO_PERF_CQ_FLIT_CNTR            11752
GM_AMO_PERF_CQ_PKT_CNTR              5876
GM_AMO_PERF_CQ_STALLED_CNTR          5092
GM_AMO_PERF_CQ_BLOCKED_CNTR            29
=========================================================================
```

Example 4. Suppress instrumented entry points from recording performance data to reduce overhead

This example assumes a NWPC group `FMAS` exists and is available for use. Because the program is traced, the `PAT_RT_TRACE_FUNCTION_NAME` is set to suppress any data collection by already instrumented entry points in *my_program*`+pat`. This means that NWPC values will only be recorded for the MAIN thread at the start and the end of the instrumented program. Instrumentation overhead is minimal.

```
$ pat_build -u -g mpi my_program
$ export PAT_RT_NWPC=FMAS
$ export PAT_RT_TRACE_FUNCITON_NAME=*:0
$ aprun -n32 my_program+pat
```

This gives a high-level view of the program's use of the networking router in terms of what the FMAS group describes. If more details about NWPC use during execution of the program are desired, the `PAT_RT_TRACE_FUNCTION_NAME` environment variable need not be set, but the significant overhead injected by reading the NWPCs may make the resulting performance data inaccurate.

To selectively collect NWPCs and the other performance data for traced functions, add them to the end of `PAT_RT_TRACE_FUNCTION_NAME`:

```
$ export PAT_RT_TRACE_FUNCTION_NAME=0:*,mxm,MPI_Bcast
```

1.2 Gemini NIC Counters

To better understand how to use the NIC counters, you need to understand some of the terminology specific to the Gemini network architecture.

The Block Transfer Engine (BTE)

A Gemini *network packet* typically consists of one or more *flits*, which are the units of flow control for the network. Because flits are usually larger than the physical datapath, they are divided into *phits*, which are the units of data that the network can handle physically. A packet must contain at least two phits, one for the header and one for the cyclical redundancy check (CRC).

The `V0` counters support the request channel and the `V1` counters support the response channel. A flit/pkt ratio can tell the user if the data entering the network was not aligned, eg a ratio greater than 1 indicates misaligned data is being sent across the network. Because there is a bandwidth/pipe size difference between outgoing and incoming (outgoing is smaller), in general you will notice more stalls on the `V0` (request) channel.

The following counters are recommended as a way to begin using the Gemini NWPC:

`GM_ORB_PERF_VC0_STALLED`

`GM_ORB_PERF_VC1_STALLED`

`GM_ORB_PERF_VC0_PKTS`

`GM_ORB_PERF_VC1_PKTS`

`GM_ORB_PERF_VC0_FLITS`

`GM_ORB_PERF_VC1_FLITS`

Table 1. Atomic Memory Operations Performance Counters

Name	Description
`GM_AMO_PERF_ACP_COMP_CNTR`	Number of Atomic Memory Operation (AMO) computations that have occurred.
`GM_AMO_PERF_ACP_MEM_UPDATE_CNTR`	Number of AMO logic cache write-throughs that have occurred.
`GM_AMO_PERF_ACP_STALL_CNTR`	Number of AMO logic pipeline stalls that have occurred.
`GM_AMO_PERF_AMO_HEADER_CNTR`	Number of request headers processed by the Decode Logic that have had an AMO computation. Error packets are not counted.
`GM_AMO_PERF_COUNTER_EN`	When set, counting is enabled. When cleared, counting is disabled.
`GM_AMO_PERF_CQ_BLOCKED_CNTR`	Number of cycles the CQ FIFO is blocked.

Name	Description
GM_AMO_PERF_CQ_FLIT_CNTR	Number of flits (network flow control units) that are read from the CQ FIFO.
GM_AMO_PERF_CQ_PKT_CNTR	Number of packets that are read from the CQ FIFO.
GM_AMO_PERF_CQ_STALLED_CNTR	Number of cycles the CQ FIFO is stalled.
GM_AMO_PERF_DONE_INV_CNTR	Number of times a valid cache entry was invalidated because there were no more outstanding AMO requests targeting it and the last request did not have the `cacheable` bit set.
GM_AMO_PERF_ERROR_HEADER_CNTR	Number of request headers processed by the Decode Logic that have had errors.
GM_AMO_PERF_FLUSH_HEADER_CNTR	Number of request headers processed by the Decode Logic that have had a Flush command. Error packets are not counted.
GM_AMO_PERF_FULL_INV_CNTR	Number of times a valid but inactive cache entry was invalidated to make room for a new AMO address. A high value in this counter indicates that there are too many cacheable AMO addresses and that the cache is being thrashed.
GM_AMO_PERF_GET_HEADER_CNTR	Number of request headers processed by the Decode Logic that have had an GET command. Error packets are not counted.
GM_AMO_PERF_MSGCOMP_HEADER_CNTR	Number of request headers processed by the Decode Logic that have had a MsgComplete command. Error packets are not counted.
GM_AMO_PERF_PUT_HEADER_CNTR	Number of request headers processed by the Decode Logic that have had an PUT command. Error packets are not counted.
GM_AMO_PERF_REQLIST_FULL_STALL_CNTR	Number of times an AMO request causes the NRP to stall waiting for a Request List entry to become free.
GM_AMO_PERF_RMT_BLOCKED_CNTR	Number cycles the RMT FIFO is blocked
GM_AMO_PERF_RMT_FLIT_CNTR	Number of flits that are read from the RMT FIFO
GM_AMO_PERF_RMT_PKT_CNTR	Number of packets that are read from the RMT FIFO
GM_AMO_PERF_RMT_STALLED_CNTR	Number cycles the RMT FIFO is stalled

Name	Description
GM_AMO_PERF_TAG_HIT_CNTR	Number of AMO requests that have been processed in the Tag Store and have resulted in a cache hit.
GM_AMO_PERF_TAG_MISS_CNTR	Number of AMO requests that have been processed in the Tag Store and have resulted in a cache miss.
GM_AMO_PERF_TAG_STALL_CNTR	Number of times a GET/PUT request hits in the cache and causes the NRP to stall.

Table 2. Fast Memory Access Performance Counters

Name	Description
GM_FMA_PERF_CQ_PKT_CNT	Number of packets from Fast Memory Access (FMA) to CQ.
GM_FMA_PERF_CQ_STALLED_CNT	Number of clock cycles FMA_CQ was stalled due to lack of credits.
GM_FMA_PERF_HT_NP_REQ_FLIT_CNT	Number of HT NP request flits to FMA.
GM_FMA_PERF_HT_NP_REQ_PKT_CNT	Number of HT NP request packets to FMA.
GM_FMA_PERF_HT_P_REQ_FLIT_CNT	Number of HT P request flits to FMA.
GM_FMA_PERF_HT_P_REQ_PKT_CNT	Number of HT P request packets to FMA.
GM_FMA_PERF_HT_RSP_PKT_CNT	Number of HT response packets from FMA to HT.
GM_FMA_PERF_HT_RSP_STALLED_CNT	Number of clock cycles FMA_HT_RSP was stalled due to lack of credits.
GM_FMA_PERF_TARB_FLIT_CNT	Number of flits from FMA to TARB.
GM_FMA_PERF_TARB_PKT_CNT	Number of packets from FMA to TARB.
GM_FMA_PERF_TARB_STALLED_CNT	Number of clock cycles FMA_TARB was stalled due to lack of credits.

Table 3. Hyper-transport Arbiter Performance Counters

Name	Description
GM_HARB_PERF_AMO_NP_BLOCKED	Number of times AMO Non-Posted Queue has an entry, but is blocked from using the Non-Posted Initiator Request output channel by the BTE Non-Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_NP_FLITS	Number of flits coming out of the AMO Non-Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_NP_PKTS	Number of packets coming out of the AMO Non-Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_NP_STALLED	Number of cycles the AMO Non-Posted Queue is stalled due to a lack credits on the Non-Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_ACP_BLOCKED	Number of times AMO Posted AMO Computation Pipe Queue has an entry, but is blocked from using the Posted Initiator Request output channel by another Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_ACP_FLITS	Number of flits coming out of the AMO Posted AMO Computation Pipe Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).

Name	Description
GM_HARB_PERF_AMO_P_ACP_PKTS	Number of packets coming out of the AMO Posted AMO Computation Pipe Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_ACP_STALLED	Number of cycles the AMO Posted AMO Computation Pipe Queue is stalled due to a lack credits on the Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_NRP_BLOCKED	Number of times AMO Posted New Request Pipe Queue has an entry, but is blocked from using the Posted Initiator Request output channel by another Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_NRP_FLITS	Number of flits coming out of the AMO Posted New Request Pipe Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_NRP_PKTS	Number of packets coming out of the AMO Posted New Request Pipe Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_AMO_P_NRP_STALLED	Number of cycles the AMO Posted New Request Pipe Queue is stalled due to a lack credits on the Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).

Name	Description
GM_HARB_PERF_BTE_NP_BLOCKED	Number of times AMO Non-Posted BTE Queue has an entry, but is blocked from using the Non-Posted Initiator Request output channel by another Non-Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_BTE_NP_FLITS	Number of flits coming out of the AMO Non-Posted BTE Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_BTE_NP_PKTS	Number of packets coming out of the AMO Non-Posted BTE Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_BTE_NP_STALLED	Number of cycles the AMO Non-Posted BTE Queue is stalled due to a lack credits on the Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_BTE_P_BLOCKED	Number of times AMO Posted BTE Queue has an entry, but is blocked from using the Posted Initiator Request output channel by another Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_BTE_P_FLITS	Number of flits coming out of the AMO Posted BTE Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).

Name	Description
GM_HARB_PERF_BTE_P_PKTS	Number of packets coming out of the AMO Posted BTE Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_BTE_P_STALLED	Number of cycles the AMO Posted BTE Queue is stalled due to a lack credits on the Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_COUNTER_EN	When set, counting is enabled. When clear, counting is disabled. This MMR is reset by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_IREQ_NP_FLITS	Number of flits on the non-posted initiator request output of the HARB block. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_IREQ_NP_PKTS	Number of packets on the non-posted initiator request output of the HARB Block. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_IREQ_NP_STALLED	Number of cycles on the non-posted initiator request output of the HARB is stalled due to a lack credits on the Non-Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).

Name	Description
GM_HARB_PERF_IREQ_P_FLITS	Number of flits on the posted initiator request output of the HARB block. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_IREQ_P_PKTS	Number of packets on the posted initiator request output of the HARB Block. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_IREQ_P_STALLED	Number of cycles on the posted initiator request output of the HARB is stalled due to a lack credits on the Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_RAT_P_BLOCKED	Number of times AMO Posted RAT Queue has an entry, but is blocked from using the Posted Initiator Request output channel by another Posted Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_RAT_P_FLITS	Number of flits coming out of the AMO Posted RAT Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).

Name	Description
GM_HARB_PERF_RAT_P_PKTS	Number of packets coming out of the AMO Posted RAT Queue. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).
GM_HARB_PERF_RAT_P_STALLED	Number of cycles the AMO Posted RAT Queue is stalled due to a lack credits on the Posted Initiator Request channel. The Local Block has read/write access to the full counter. Bits 63:48 of this MMR are unimplemented and always return zero. This MMR is reset to all zeros by the chip reset (i_reset), but not by HT reset (i_ht_reset).

Table 4. Network Address Translation Performance Counters

Name	Description
GM_NAT_PERF_BTE_BLOCKED	Number of cycles a BTE translation is blocked due to arbitration loss.
GM_NAT_PERF_BTE_STALLED	Number of cycles a BTE translation is stalled due to MMR access.
GM_NAT_PERF_BTE_TRANSLATIONS	Number of translations performed for the BTE interface.
GM_NAT_PERF_COUNTER_EN	When set, counting is enabled. When cleared, counting is disabled.
GM_NAT_PERF_REQ_BLOCKED	Number of cycles a REQ translation is blocked due to arbitration loss.
GM_NAT_PERF_REQ_STALLED	Number of cycles a REQ translation is stalled due to MMR access.
GM_NAT_PERF_REQ_TRANSLATIONS	Number of translations performed for the REQ interface.
GM_NAT_PERF_RSP_BLOCKED	Number of cycles a RSP translation is blocked due to arbitration loss.
GM_NAT_PERF_RSP_STALLED	Number of cycles a RSP translation is stalled due to MMR access.
GM_NAT_PERF_RSP_TRANSLATIONS	Number of translations performed for the RSP interface.
GM_NAT_PERF_TRANS_ERROR0	Number of translations that failed due to error 0 (Uncorrectable error in translation).

Name	Description
GM_NAT_PERF_TRANS_ERROR1	Number of translations that failed due to error 1 (VMDH table invalid entry).
GM_NAT_PERF_TRANS_ERROR2	Number of translations that failed due to error 2 (MDDT/MRT invalid or illegal entry).
GM_NAT_PERF_TRANS_ERROR3	Number of translations that failed due to error 3 (Protection tag violation).
GM_NAT_PERF_TRANS_ERROR4	Number of translations that failed due to error 4 (memory bounds error).
GM_NAT_PERF_TRANS_ERROR5	Number of translations that failed due to error 5 (write permission error)

Table 5. Netlink Performance Counters

Name	Description
GM_NL_PERF_ALL_LCBS_REQS_TO_NIC_0_STALLED	Number of ticks all LCBs requests have stalled to NIC 0.
GM_NL_PERF_ALL_LCBS_REQS_TO_NIC_1_STALLED	Number of ticks all LCBs requests have stalled to NIC 1.
GM_NL_PERF_ALL_LCBS_RSP_TO_NIC_0_STALLED	Number of ticks all LCBs responses have stalled to NIC 0.
GM_NL_PERF_ALL_LCBS_RSP_TO_NIC_1_STALLED	Number of ticks all LCBs responses have stalled to NIC 1.
GM_NL_PERF_CNTRL	Controls the performance counters. Writing a 1 to the Start field starts the counters. Writing a 1 to the Stop field stops the counters. Writing a 1 to the Clear field clears the counters.
GM_NL_PERF_LCB_*n*_REQ_CMP_22	Decompressed request data to two phit LCB_*n*, where *n* is a value from 0 to 7 that specifies the LCB.
GM_NL_PERF_LCB_*n*_REQ_CMP_44	Decompressed request data to one phit LCB_*n*, where *n* is a value from 0 to 7 that specifies the LCB.
GM_NL_PERF_LCB_*n*_REQ_TO_NIC_0	Number of requests from LCB_*n* to NIC 0.
GM_NL_PERF_LCB_*n*_REQ_TO_NIC_0_STALLED	Number of ticks LCB_*n* requests are blocked to NIC 0.
GM_NL_PERF_LCB_*n*_REQ_TO_NIC_1	Number of requests from LCB_*n* to NIC 1.

Name	Description
GM_NL_PERF_LCB_*n*_REQ_TO_NIC_1_STALLED	Number of ticks LCB_*n* requests are blocked to NIC 1.
GM_NL_PERF_LCB_*n*_REQ_TO_PHITS	Number of request phits received on LCB_*n*.
GM_NL_PERF_LCB_*n*_REQ_TO_PKTS	Number of request packets received on LCB_*n*.
GM_NL_PERF_LCB_*n*_RSP_CMP_22	Decompressed response data to two phit LCB_*n*
GM_NL_PERF_LCB_*n*_RSP_TO_NIC_1	Number of responses from LCB_*n* to NIC 1.
GM_NL_PERF_LCB_*n*_RSP_TO_NIC_1_STALLED	Number of ticks LCB_*n* responses are blocked to NIC 1.
GM_NL_PERF_NIC_0_REQ_STALLED_TO_ALL_LCBS	Number of ticks NIC_0 requests are blocked to all LCBs.
GM_NL_PERF_NIC_0_REQ_TO_LCB_*n*	Number of requests from NIC_0 LCB_ *n*.
GM_NL_PERF_NIC_0_REQ_TO_LCB_*n*_STALLED	Number of ticks NIC_0 requests are blocked to LCB_*n*.
GM_NL_PERF_NIC_0_RSP_STALLED_TO_ALL_LCBS	Number of ticks NIC_0 responses are blocked to all LCBs.
GM_NL_PERF_NIC_0_RSP_TO_LCB_*n*	Number of responses from NIC_0 LCB_ *n*.
GM_NL_PERF_NIC_0_RSP_TO_LCB_*n*_STALLED	Number of ticks NIC_0 responses are blocked to LCB_*n*.
GM_NL_PERF_NIC_1_REQ_STALLED_TO_ALL_LCBS	Number of ticks NIC_0 requests are blocked to all LCBs.
GM_NL_PERF_NIC_1_REQ_TO_LCB_*n*	Number of requests from NIC_1 to LCB_ *n*.
GM_NL_PERF_NIC_1_REQ_TO_LCB*n*_STALLED	Number of ticks NIC_1 requests are blocked to LCB_*n*.
GM_NL_PERF_NIC_1_RSP_STALLED_TO_ALL_LCBS	Number of ticks NIC_1 responses are blocked to all LCBs.
GM_NL_PERF_NIC_1_RSP_TO_LCB_*n*	Number of responses from NIC_1 LCB_ *n*.
GM_NL_PERF_NIC_1_RSP_TO_LCB_*n*_STALLED	Number of ticks NIC_1 responses are blocked to LCB_*n*.

Table 6. NPT Performance Counters

Name	Description
GM_NPT_PERF_ACP_BLOCKED_CNTR	Number of cycles the ACP FIFO is blocked.
GM_NPT_PERF_ACP_FLIT_CNTR	Number of flits that are read from the ACP FIFO.
GM_NPT_PERF_ACP_PKT_CNTR	Number of packets that are read from the ACP FIFO.
GM_NPT_PERF_ACP_STALLED_CNTR	Number of cycles the ACP FIFO is stalled.
GM_NPT_PERF_BTE_RSP_PKT_CNTR	Number of packets that are sent to the Netlink as Get or Flush responses.
GM_NPT_PERF_COUNTER_EN	Provides the count enable.
GM_NPT_PERF_FILL_RSP_PKT_CNTR	Number of packets that are sent to the AMO block as fill responses.
GM_NPT_PERF_HTIRSP_ERR_CNTR	Number of packets that are received from the HT cave and have an error status.
GM_NPT_PERF_HTIRSP_FLIT_CNTR	Number of flits that are received from the HT cave.
GM_NPT_PERF_HTIRSP_PKT_CNTR	Number of packets that are received from the HT cave.
GM_NPT_PERF_LB_BLOCKED_CNTR	Number of cycles the LB FIFO is blocked.
GM_NPT_PERF_LB_FLIT_CNTR	Number of flits that are read from the LB FIFO.
GM_NPT_PERF_LB_PKT_CNTR	Number of packets that are read from the LB FIFO.
GM_NPT_PERF_LB_STALLED_CNTR	Number of cycles the LB FIFO is stalled.
GM_NPT_PERF_NL_RSP_PKT_CNTR	Number of packets that are sent to the AMO block as fill responses.
GM_NPT_PERF_NPT_BLOCKED_CNTR	Number of cycles the NPT FIFO is blocked.
GM_NPT_PERF_NPT_FLIT_CNTR	Number of flits that are read from the NPT FIFO.
GM_NPT_PERF_NPT_PKT_CNTR	Number of packets that are read from the NPT FIFO.
GM_NPT_PERF_NPT_STALLED_CNTR	Number of cycles the NPT FIFO is stalled.
GM_NPT_PERF_NRP_BLOCKED_CNTR	Number of cycles the NRP FIFO is blocked.
GM_NPT_PERF_NRP_FLIT_CNTR	Number of flits that are read from the NRP FIFO.
GM_NPT_PERF_NRP_PKT_CNTR	Number of packets that are read from the NRP FIFO.
GM_NPT_PERF_NRP_STALLED_CNTR	Number of cycles the NRP FIFO is stalled.

Table 7. ORB Performance Counters

Name	Description
GM_ORB_PERF_VC0_FLITS	Number of flits to come into the TX Input Queue from the SSID.
GM_ORB_PERF_VC0_PKTS	Number of packets to come into the TX Input Queue from the SSID.
GM_ORB_PERF_VC0_STALLED	Number of packets not given access to the TX Control Logic because there is not enough credits available from the NL Block, or there are no available memory locations from the ORD RAM, or a tail flit has not been received in the ORB Input Queue when performing store-and-forward.
GM_ORB_PERF_VC1_BLOCKED	Number of packets not given access to the RX Control Logic because the read address and write address into the ORD RAM are attempting to access the same bank of the ORD RAM or because there is a read access to the ORD RAM from the Local Block.
GM_ORB_PERF_VC1_BLOCKED_PKT_GEN	Number of times the RX Response FIFO is blocked because a packet in the RX Control Logic is being translated into the format used by the rest of the NIC.
GM_ORB_PERF_VC1_FLITS	Number of flits to come into the Receive Response FIFO from the network.
GM_ORB_PERF_VC1_PKTS	Number of packets to come into the Receive Response FIFO from the network.
GM_ORB_PERF_VC1_STALLED	Number of packets not given access to the RX Control Logic because there is not enough credits available from the RAT.

Table 8. RAT Performance Counters

Name	Description
GM_RAT_PERF_COUNTER_EN	Enables the performance counters.
GM_RAT_PERF_DATA_FLITS_VC0	Number of data flits received on VC0 (request pipeline).
GM_RAT_PERF_DATA_FLITS_VC1	Number of data flits received on VC1 (request pipeline).
GM_RAT_PERF_HEADER_FLITS_VC0	Number of header flits received on VC0 (request pipeline).
GM_RAT_PERF_HEADER_FLITS_VC1	Number of header flits received on VC1 (request pipeline).
GM_RAT_PERF_STALLED_CREDITS_VC0	Number of cycles VC0 (request pipeline) is stalled due to insufficient credits.
GM_RAT_PERF_STALLED_CREDITS_VC1	Number of cycles VC1 (request pipeline) is stalled due to insufficient credits.
GM_RAT_PERF_STALLED_TRANSLATION_VC0	Number of cycles VC0 (request pipeline) is stalled due to unavailable translation data.
GM_RAT_PERF_STALLED_TRANSLATION_VC1	Number of cycles VC1 (request pipeline) is stalled due to unavailable translation data.
GM_RAT_PERF_TRANSLATION_ERRORS_VC0	Number of translation errors seen on VC0 (request pipeline).
GM_RAT_PERF_TRANSLATION_ERRORS_VC1	Number of translation errors seen on VC1 (request pipeline).
GM_RAT_PERF_TRANSLATIONS_VC0	Number of translations requested on VC0 (request pipeline).
GM_RAT_PERF_TRANSLATIONS_VC1	Number of translations requested on VC1 (request pipeline).

Table 9. RMT Performance Counters

Name	Description
GM_RMT_PERF_PUT_BYTES_RX	Tally of bytes received in all PUT packets that had the RMT Enable field set that entered and exited the RMT with OK status.
GM_RMT_PERF_PUT_CAM_EVIT	PUT sequences evicted from the CAM.
GM_RMT_PERF_PUT_CAM_FILL	New PUT sequence packet arrived and successfully allocated in the CAM.
GM_RMT_PERF_PUT_CAM_HITS	Packet for PUT sequence currently stored in RMT arrived and successfully located entry in CAM.
GM_RMT_PERF_PUT_CAM_MISS	New PUT sequence packet arrived, but did not allocate because CAM was full.
GM_RMT_PERF_PUT_PARITY	Number of sequences evicted from CAM due to uncorrectable parity errors.
GM_RMT_PERF_PUT_RECV_COMPLETE	Number of MsgRcvComplete packets received which evicted a CAM entry.
GM_RMT_PERF_PUT_TIMEOUTS	Number of sequences evicted from CAM due to timeout.
GM_RMT_PERF_SEND_BYTES_RX	Tally of bytes received in all SEND packets that had the RMT Enable field set and entered and exited the RMT with OK status.
GM_RMT_PERF_SEND_CAM_EVIT	SEND sequences evicted from the CAM.
GM_RMT_PERF_SEND_CAM_FILL	New SEND sequence packet arrived and successfully allocated in the CAM.
GM_RMT_PERF_SEND_CAM_HITS	Packet for SEND sequence currently stored in RMT arrived and successfully located entry in CAM.
GM_RMT_PERF_SEND_CAM_MISS	New SEND sequence packet arrived, but did not allocate because CAM was full.
GM_RMT_PERF_SEND_PARITY	Number of sequences evicted from CAM due to uncorrectable parity errors.
GM_RMT_PERF_SEND_ABORTS	Number of SEND sequences that were aborted.
GM_RMT_PERF_SEND_TIMEOUTS	Number of sequences evicted from CAM due to timeout.

Table 10. SSID Performance Counters

Name	Description
GM_SSID_PERF_COMPLETION_COUNT_1	Provides a count of completed request packet sequences. The type of sequence completions counted by this register is controlled by the SSID Performance – Completion Count Selector Register.
GM_SSID_PERF_COMPLETION_COUNT_2	Provides a count of completed request packet sequences. The type of sequence completions counted by this register is controlled by the SSID Performance – Completion Count Selector Register.
GM_SSID_PERF_COMPLETION_COUNT_SELECTOR	Specifies the types of completion events that are counted in the SSID Performance – Completion Count 1 Register (bits 3-0) and the SSID Performance – Completion Count 2 Register (bits 11-8). See the table of SSID_PerfCompletionCountSelect Encoding values for encoding of these fields.
GM_SSID_PERF_OUT_STALLED_DURATION	The accumulated number of cycles of cclk for which the SSID had a valid flit available to send to the ORB but sending of the flit had to be stalled while waiting for a credit from the ORB. This value is cleared by writing any value to this register.
GM_SSID_PERF_OUTOFSSIDS_COUNT	The number of Allocate SSID requests that have been received for which processing of the request had to be stalled for one or more clock cycles because a free SSID was not immediately available to service the request. This value is cleared by writing any value to this register.
GM_SSID_PERF_OUTOFSSIDS_DURATION	The accumulated number of cycles of cclk for which processing of Allocate SSID requests has been stalled because a free SSID is not available to service the request. This value is cleared by writing any value to this register.

Name	Description
GM_SSID_PERF_SSID_ALLOCATE_COUNT	The total number of Allocate SSID requests that have been received, across all channels (all FMA descriptors and all BTE VCs), because this register was last cleared, and that resulted in a SSID actually being allocated. Allocate SSID requests that do not result in a SSID being allocated (i.e. redundant Allocate requests) are not counted. This value is cleared by writing any value to this register.
GM_SSID_PERF_SSIDS_IN_USE	Bits 7-0 specify the number of SSIDs currently in use across all Request Channels. This value is not affected by writes to this register. This field is initialized to its reset value by a full reset and by an ht reset. Bits 23-16 specify the maximum number of SSIDs that have been in use simultaneously, across all channels (all FMA descriptors and all BTE Vcs), since this register was last initialized. This value is initialized to CurrentSSIDsInUse by writing any value to this register. This field is initialized to its reset value by a full reset.

Table 11. Transmit Arbiter Performance Counters

Name	Description
GM_TARB_PERF_BTE_BLOCKED	Transmit Arbiter (TARB) Performance BTE Blocked Count
GM_TARB_PERF_BTE_FLITS	TARB Performance BTE Flit Count
GM_TARB_PERF_BTE_PKTS	TARB Performance BTE Packet Count
GM_TARB_PERF_BTE_STALLED	TARB Performance BTE Stalled Count
GM_TARB_PERF_FMA_BLOCKED	TARB Performance FMA Blocked Count
GM_TARB_PERF_FMA_FLITS	TARB Performance FMA Flit Count
GM_TARB_PERF_FMA_PKTS	TARB Performance FMA Packet Count
GM_TARB_PERF_FMA_STALLED	TARB Performance FMA Stalled Count
GM_TARB_PERF_LB_BLOCKED	TARB Performance LB Blocked Count
GM_TARB_PERF_LB_FLITS	TARB Performance LB Flit Count
GM_TARB_PERF_LB_PKTS	TARB Performance LB Packet Count

Name	Description
GM_TARB_PERF_LB_STALLED	TARB Performance LB Stalled Count
GM_TARB_PERF_OUT_FLITS	TARB Performance Output Flit Count
GM_TARB_PERF_OUT_PKTS	TARB Performance Output Packet Count
GM_TARB_PERF_OUT_STALLED	TARB Performance Output Stalled Count

1.3 Gemini Tile MMRs

The Gemini network consists of 48 tiles, arranged in 6 rows of 8 columns. Within each tile there are memory-mapped registers associated with the LCB and with the rest of the tile. The local block has shared connections to each row of tiles.

By default, when only the name of the MMR is used, an event is counted on all 48 tiles. To address an individual tile, append the row (`0-5`) and column (`0-7`) to the name, as shown in the table.

Table 12. Description of Gemini Tile MMRs

Name	Description
GM_TILE_PERF_VC0_PHIT_CNT:*n*:*m*	Number of `vc0` phits read from inq buffer
GM_TILE_PERF_VC1_PHIT_CNT:*n*:*m*	Number of `vc1` phits read from inq buffer
GM_TILE_PERF_VC0_PKT_CNT:*n*:*m*	Number of `vc0` packets read from inq buffer
GM_TILE_PERF_VC10_PKT_CNT:*n*:*m*	Number of `vc1` packets read from inq buffer
GM_TILE_PERF_INQ_STALL:*n*:*m*	Number of clock periods a valid reference is blocked from the routing pipeline.
GM_TILE_PERF_CREDIT_STALL:*n*:*m*	Number of clock periods a valid reference is stalled in the column buffers, waiting on transmissions credits.

Version 1.0 Published July 2010 Supports CrayPat release 5.1 and CLE release 3.1 running on Cray XT systems.