HMC Gen2 User Guide

MT43A4G40200

1 Chapter 1 – Introduction

This HMC Gen2 User Guide is organized in three distinct chapters as they relate to designing an end product with HMC: Initial Design Guidelines, Board Design Guidelines, and Board Bring-Up, Software, and Register Definitions. The initial design chapter has an overview of the HMC device and provides system-level estimation data, such as performance optimization details and RAS feature descriptions. The board design chapter makes sure that all aspects of designing a board incorporating HMC are considered. It also points to other relevant sources within the guide and, as needed, to external resources. The board bring-up and debug chapter targets both initial bring-up and software/firmware integration for the host, including details of the registers and external request interface (ERI) functions needed for configuration, runtime, and debug. Each chapter includes a checklist to help ensure the success of that design phase and a successful transition to the next.

This guide is informational in nature and is not comprehensive; refer to the HMC Gen2 data sheet for the complete device specification. Also see the HMC FAQ on micron.com to help answer your questions as they come up during each of the design phases.

Contents

List of Figures

List of Tables

2 Chapter 2 – Initial Design Guidelines

The Hybrid Memory Cube (HMC) is a revolutionary memory product that combines multiple DRAM layers with a logic layer, which manages the DRAM, and contains a high-speed serial interface (SerDes) all in the same package. This product offers 160 GB/s (1.28 Tb/s) of peak bandwidth to help users solve their memory sub system challenges.

For detailed information on the SerDes, the packet protocol or operational details of this device refer to the product data sheet.

This chapter aids the user in understanding the HMC Gen2 device capabilities, features and performance, which includes optimizing HMC performance, reliability and serviceability (RAS) features, simulation guidance and an overall initial design checklist.

2.1 Initial Design Checklist

HMC are more complicated memory devices than DDR memories and requires careful design, simulation and layout practices to ensure proper operation. This checklist is intended to act as a reminder for the important actions and their relative order of priority for new HMC designs. Delaying or skipping any of the suggested actions may allow problems to remain hidden until late in the design and bring-up process. Following this checklist and completing the associated actions will help create a more reliable and robust product.

2.1.1 Checklist

1. Use the 2-in-1 model to determine the following:
   a. The number of HMC cubes required to meet bandwidth needs
   b. The number of links to be used per HMC
   c. The link width (full or half)
   d. The SerDes link speed to be used (10, 12.5 or 15 Gb/s)
2. Determine if HMC chaining is needed to meet system memory capacity (Refer to the HMC Gen 2 Data sheet).
3. Reference Optimizing HMC Gen2 Performance (page 10)
4. Reference Reliability Features (page 27)
5. Reference Nonvolatile Memory Usage (page 32)
6. Reference HMC Gen2 Power Calculator (page 54)
   a. Use the HMC power calculator to determine the maximum power requirement per voltage rail.
7. Reference Thermal Modeling (page 61)
   a. Use the available FloTHERM model to determine the requirements for meeting the DRAM and Logic layer T_J specifications.
   b. Analyze and design a thermal heat dissipation solution that will insure the HMC junction temperatures are kept within the operating ranges specified in the data sheet. In most cases this will require a heatsink and fan, even for prototype systems. Contact your local Micron FAE/Sales contact for sockets and heatsinks available from Micron.
8. Do initial signal integrity (SI) analysis and design
   a. SI analysis of HMC SerDes, host SerDes, and board channels using IBIS AMI and S-parameter models available on micron.com

b. SI analysis of HMC low-speed signals (REFCLK, I^{2}C, and so on) using HMC GPIO IBIS model available on micron.com

2.2 2-in-1 Model

The 2-in-1 model is both a performance model and a bus functional model (BFM). Included with the 2-in-1 model available on micron.com is a readme providing additional usage details.

2.3 Optimizing HMC Gen2 Performance

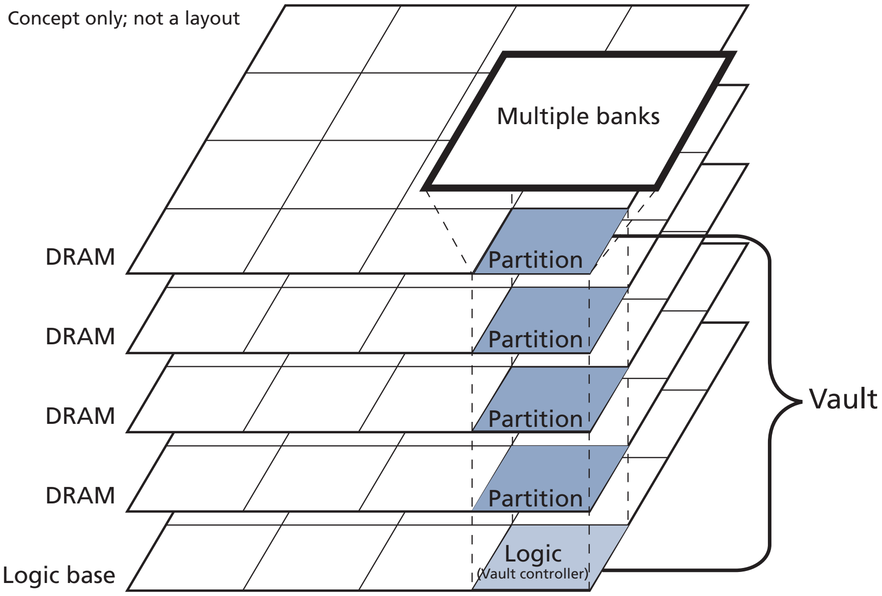
HMC devices provide an unprecedented level of sustainable bandwidth using a fraction of the board space, pin count and power per bit when compared to commodity DRAMs like DDR3 and DDR4. The HMC logic layer abstracts the DRAM layers' functions from the host and manages all the DRAM timing parameters and refresh requirements. The vault controllers have a mechanism to deal with bank conflicts (a request to a bank that has not met its DRAM cycle time requirement). The host controller can simply issue requests and receive responses. This reduces the complexity of the host memory controller and simplifies the system design.

2.3.1 Overview

An HMC is a single package containing four DRAM die and one logic die. The layers are connected using through-silicon via (TSV) technology. The four-high DRAM stack has a density of 2GB (16Gb). Within each cube, memory is organized vertically—portions of each memory die are combined with the corresponding portions of the other memory die in the stack. Each group of memory partitions is combined with a corresponding section of the logic die, forming what is referred to as a vault (shown in the figure below). Each vault has a memory controller in the logic base called a vault controller that manages all memory operations (timing, refresh and command sequencing) within that vault.

An HMC's overall performance is based on the aggregate of its DRAM, SerDes links, and logic layer efficiencies. As each of these areas are discussed examples will be given to highlight the associated impact and tradeoffs.

Figure 1: HMC Logical Block Diagram



2.3.2 Bandwidth Constraint Overview

The HMC Gen2 is a 2GB cube (4Gb x 4-high DRAM stack) containing 128 banks divided into 16 vaults. Each vault contains 8 DRAM banks.

Each vault has a maximum bandwidth of 10 GB/s, for a total of 160 GB/s maximum DRAM bandwidth per cube. This bandwidth can be constrained by many factors. We will discuss the topics that could have the most significant impact on bandwidth. Systems designed with these items in mind will benefit from higher sustainable bandwidth.

- Packet size: HMC supports packet sizes ranging from 16B to 128B, in 16B increments (16, 32, 48, 64, 80, 96, 112 and 128B).
- Packet protocol vs. link rate : The packet-based protocol contains a header and a tail, which adds a varying percentage of overhead depending upon the packet size(s) being used. The HMC can be configured to operate at three different link rates (10, 12.5 or 15 GB/s). A later section will explain how the higher link rates (12.5 and 15 GB/s) can be used to reduce or eliminate the protocol overhead.
- Bank availability: After a bank has been accessed, there is a delay before that bank can be accessed again.
- Local vaults vs. remote vaults: Each link is assigned 4 vaults considered to be local. The other 12 vaults are considered remote and must access the cross bar switch for both the request and response.
- Refresh: The DRAM cells must be refreshed to maintain the programmed data. The refreshes are handled internally by the HMC device. The internal refresh rate depends on the DRAM junction temperature allowed by the system. Some system implementations may see a reduction in system bandwidth due to refresh at high temperatures.
- Token return loop time: In order to maintain link performance, the token return loop time must be less than the amount of time it takes to fill the link input buffer.

- Retry pointer loop time: In order to maintain link performance, the maximum allowable retry pointer loop time must be less than the retry buffer full period (host allowable delay).

2.3.3 Packet Size

HMC supports packet sizes ranging from 16-byte to 128-byte, in 16-byte increments (16, 32, 48, 64, 80, 96, 112 and 128B). The DRAM data bus within each vault is optimized for 32-byte accesses (32, 64, 96, and128B). Inefficient utilization of this bus occurs when starting or ending on a 16-byte boundary (16, 48, 80, and 112B). The table below shows the maximum DRAM bandwidth per vault and per cube for the different packet sizes. The BW values shown in this table, are meant to highlight the reduction in BW associated with 16-byte boundary accesses.

An extreme example of this inefficiency would be the host issuing a series of 16-byte read requests. Each read request would fetch 32 bytes from the DRAM and return half of the data in the response packet, throwing away the other 16 bytes of data. This type of access is supported, but the inefficiencies associated with these type of accesses can reduce system bandwidth depending on the number of active links and whether or not the DRAM BW is limiting the overall BW.

Issuing requests which utilize 32-byte boundary accesses (32, 64, 96, 128B) will help optimize system bandwidth.

Table 1: HMC DRAM Bandwidth vs. Packet Size

	Maximum DRAM Bandwidth	
Packet Size (Bytes)	Per Cube	Unit
16	80	GB/s
32	160	
48	120	
64	160	
80	133.33	
96	160	
112	140	
128	160	

2.3.4 Packet Protocol

Commands and data are transmitted in both directions across a link using a packet-based protocol where the packets consist of 128-bit flow units called FLITS. A FLIT consists of 128b or 16 bytes (16B) of payload or control fields. Packets can range in size from 1 to 9 FLITs. The tables below show the number of FLITs and the associated protocol overhead for READ and WRITE commands for each of the supported packet sizes. Read requests and non-posted write responses are always a single FLIT transaction.

Table 2: Read Request and Response – Packet Protocol Overhead

Packet Size	16B	32B	48B	64B	80B	96B	112B	128B
Read Request FLIT count (overhead)	1	1	1	1	1	1	1	1
Read Response FLIT count (payload)	1	2	3	4	5	6	7	8

Table 2: Read Request and Response – Packet Protocol Overhead (Continued)

Packet Size	16B	32B	48B	64B	80B	96B	112B	128B
Read Response FLIT count (overhead)	1	1	1	1	1	1	1	1
Total FLIT count	3	4	5	6	7	8	9	10
Read Request Protocol Overhead	100%	100%	100%	100%	100%	100%	100%	100%
Read Response Protocol Overhead	50%	33%	25%	20%	16.7%	14.3%	12.5%	11%
Total Protocol Overhead	66%	50%	40%	33%	29%	25%	22%	20%

Table 3: Non-Posted Write Request and Response – Packet Protocol Overhead

Packet Size	16B	32B	48B	64B	80B	96B	112B	128B
Write request FLIT count (overhead)	1	1	1	1	1	1	1	1
Write request FLIT count (payload)	1	2	3	4	5	6	7	8
Write response FLIT count (overhead)	1	1	1	1	1	1	1	1
Total FLIT count	3	4	5	6	7	8	9	10
Write Request protocol overhead	50%	33%	25%	20%	16.7%	14.3%	12.5%	11%
Write Response protocol overhead	100%	100%	100%	100%	100%	100%	100%	100%
Total protocol overhead	66%	50%	40%	33%	29%	25%	22%	20%

Table 4: Posted Write – Packet Protocol Overhead

Packet Size	16B	32B	48B	64B	80B	96B	112B	128B
Write request FLIT count (overhead)	1	1	1	1	1	1	1	1
Write request FLIT count (payload)	1	2	3	4	5	6	7	8
Write response FLIT count (overhead)	0	0	0	0	0	0	0	0
Total FLIT count	2	3	4	5	6	7	8	9
Write Request protocol overhead	50%	33%	25%	20%	16.7%	14.3%	12.5%	11%
Write Response protocol overhead	0%	0%	0%	0%	0%	0%	0%	0%
Total protocol overhead	50%	33%	25%	20%	16.7%	14.3%	12.5%	11%

2.3.5 Link Rate

An HMC has up to four links, each comprised of 16 differential lanes in both transmit (TX) and receive (RX) directions. The links can be configured to all run at a frequency of 10 GB/s, 12.5 GB/s or 15 GB/s. These are exact frequencies, no other frequency rates are allowed. The lanes of a link can be configured to run in full-width (16 lanes) or half-width (8 lanes) configuration. Full-width and half-width configuration is determined in both the transmit (TX) and receive (RX) directions and do not have to match. The following lane configurations are allowed: 16 x 16, 16 x 8, 8 x 16, or 8 x 8 RX/TX lanes. The available link bandwidth per cube is determined by the number of active links, the configured link width and the selected line rate. The three tables below highlight the half-width link bandwidth, the full-width link bandwidth, and compares the link bandwidth at the each link rate (10, 12.5 and 15 GB/s) to the maximum DRAM bandwidth, which shows the percentage of head room (available link bandwidth above the maximum DRAM bandwidth).

Table 5: Half-Width Link Bandwidth

Number of Half width Links	Link Rate			Unit
	10 GB/s	12.5 GB/s	15 GB/s	
1	20	25	30	GB/s
2	40	50	60	
3	60	75	90	
4	80	100	120	

Table 6: Full-Width Link Bandwidth

Number of Full-Width Links	Link Rate			Unit
	10 GB/s	12.5 GB/s	15 GB/s	
1	40	50	60	GB/s
2	80	100	120	
3	120	150	180	
4	160	200	240	

Table 7: Link Bandwidth vs. DRAM Bandwidth

Link Rate	SerDes Bandwidth (4 Full-Width Links)	DRAM Data Bandwidth	Head Room
10 GB/s	160 GB/s	160 GB/s	0%
12.5 GB/s	200 GB/s	160 GB/s	25%
15 GB/s	240 GB/s	160 GB/s	50%

2.3.6 Packet Protocol vs. Link Rate

Which link rate should you use?

Before deciding which link rate to use there are a few key points that need to be understood.

- The link rate is completely independent from the internal DRAM bandwidth. Changing the external link rate does not effect or change the theoretical MAX DRAM bandwidth, because the internal DRAM clock rate does not change when different external link rates are used.
- To reduce the inefficiencies of packet overhead and to optimize the DRAM bandwidth use the faster 12.5 and 15 GB/s link rates.
- The link rate selected may impact the number of links required to support the desired bandwidth. This is a tradeoff between the number of signals to be routed versus the link rate to be supported.

Example 1:

- 16B packets

- 1-to-1 read/write ratio
- 60 GB/s (30 GB/s read + 30 GB/s posted write) of DRAM bandwidth required
  - Links must support 150 GB/s due to packet protocol overhead
    - 60 GB/s posted write (minus 50% overhead = 30 GB/s)
    - 90 GB/s read (minus 66% overhead = 30 GB/s)
      - 30 GB/s read requests(100% overhead)
      - 60 GB/s read responses (minus 50% overhead = 30 GB/s)

Options to obtain an aggregate link bandwidth (RX + TX) capable of supporting 150 GB/s:

- 5 full-width links (2 HMC) at 10 GB/s (40 GB/s per link, 200 GB/s total [100 GB/s RX + 100 GB/s TX])
- 4 full-width links (1 HMC) at 12.5 GB/s (50 GB/s per link, 200 GB/s total [100 GB/s RX + 100 GB/s TX])
- 3 full-width links (1 HMC) at 15 GB/s (60 GB/s per link, 180 GB/s total [90 GB/s RX + 90 GB/s TX])

Example 2:

- 64B packets
- 1-to-1 read/write ratio
- 80 GB/s (40 GB/s read + 40 GB/s posted write) of DRAM bandwidth required
  - Links must support 110 GB/s due to packet protocol overhead
    - 50 GB/s posted write (minus 20% overhead = 40 GB/s)
    - 60 GB/s read (minus 33% overhead = 40 GB/s)
      - 10 GB/s read requests (100% overhead)
      - 50 GB/s read responses (minus 20% overhead = 40 GB/s)

Options to obtain an aggregate link bandwidth (RX + TX) capable of supporting 110 GB/s:

- 3 full-width links at 10 GB/s (40 GB/s per link, 120 GB/s total [60 GB/s RX + 60 GB/s TX])
- 3 full-width links at 12.5 GB/s (50 GB/s per link, 150 GB/s total [75 GB/s RX + 75 GB/s TX])
- 2 full-width links at 15 GB/s (60 GB/s per link, 120 GB/s total [60 GB/s RX + 60 GB/s TX])

Example 3:

- 128B packets
- 1-to-1 read/posted write ratio
- 70 GB/s (35 GB/s read + 35 GB/s posted write) of DRAM bandwidth required
  - Links must support 84 GB/s due to packet protocol overhead
    - 40 GB/s posted write (minus 11% overhead = 35 GB/s)
    - 44 GB/s read (minus 20% overhead = 35 GB/s)
      - 4 GB/s read requests (100% overhead)
      - 40 GB/s read responses (minus 11% overhead = 35 GB/s)

Options to obtain an aggregate link bandwidth (RX + TX) capable of supporting 84 GB/s:

- 3 full-width links at 10 GB/s (40 GB/s per link, 120 GB/s total [60 GB/s RX + 60 GB/s TX])
- 2 full-width links at 12.5 GB/s (50 GB/s per link, 100 GB/s total [50 GB/s RX + 50 GB/s TX])
- 2 full-width links at 15 GB/s (60 GB/s per link, 120 GB/s total [60 GB/s RX + 60 GB/s TX])
- 3 half width links at 15 GB/s (30 GB/s per link, 90 GB/s total [45 GB/s RX + 45 GB/s TX])

There are pros and cons to each option, such as the number of routes vs. the link frequency to be supported, the total power consumption, latency and the overall thermal impact. All of these factors need to be carefully considered prior to determining which option is best for your design.

2.3.7 Bank Availability

HMC's optimized DRAM design eliminates most of the common timing constraints (^{t}RRD, ^{t}FAW and ^{t}RFC) which system designers have been dealing with for years. However, there is still a bank availability delay (in other memory products this is referred to as Row Cycle time, ^{t}RC) that must be satisfied between accesses to the same bank. In other words, after a specific bank has been accessed, there is a required delay before that bank can be accessed again. This parameter is not specified in the HMC data sheet, because the vault controllers are designed to deal with bank conflicts (trying to access a bank before it is available). If a vault controller receives a request for a bank that is not available, it will set aside this request and process other requests to available banks. When the unavailable bank becomes available, the vault controller will schedule the request that was set aside to be executed. It is important for designers to understand that this bank availability delay exists so that memory access patterns that can be controlled are controlled in the most efficient manner.

How the DRAM banks are accessed can significantly impact the achievable HMC bandwidth.

For example, a single link accessing four vaults has a maximum DRAM bandwidth of 40 GB/s available. We will look at three different bank access scenarios.

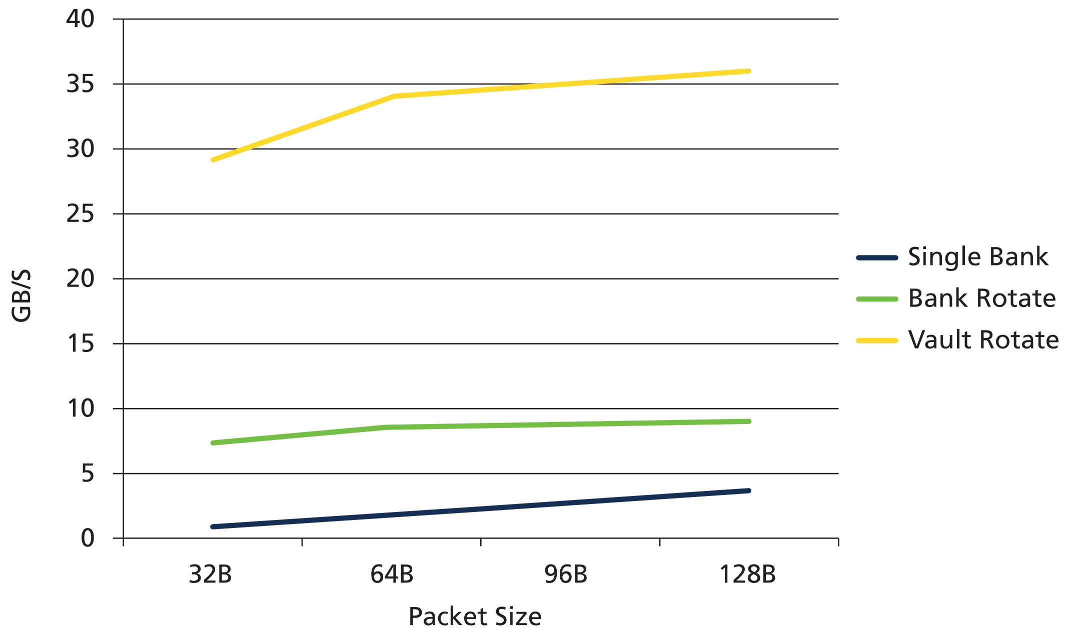
Single Bank: Repeatedly accessing a single bank in a single vault (READ Vault 0, Bank 0, READ Vault 0, Bank 0, READ Vault 0, Bank 0, READ Vault 0, Bank 0, and so on). This scenario only allows one request to be processed each ^{t}RC period.

Bank Rotate: Rotating between all the available banks of one vault and then going on to the next vault and accessing all the banks of this vault (READ Vault 0, Bank 0, READ Vault 0, Bank 1, READ Vault 0, Bank 2, READ Vault 0, Bank 3, and so on). This significantly increases the number of requests that can be processed, because the ^{t}RC delay is avoided by interleaving accesses between multiple banks. However, it does not take full advantage of having requests processing in all four vaults simultaneously.

Vault Rotate: Rotating between the vaults and then rotating between all the banks in each vault (READ Vault 0, Bank 0, READ Vault 1, Bank 0, READ Vault 2, Bank 0, READ Vault 3, Bank 0, READ Vault 0, Bank 1, READ Vault 1, Bank 1, READ Vault 2, Bank 1, READ Vault 3, Bank 1, and so on). This produces the best performance, because not only does it avoid the tRC delay like the bank rotate scheme, it also takes advantage of the internal vault structure, which allows requests to be processed in all vaults simultaneously.

The following figure shows the difference in achievable bandwidth of the three different bank access methods discussed (single bank, bank rotate and vault rotate). The bandwidth information is for a 2GB HMC using only one link running at 15 GB/s and accessing only the four local vaults with 50% reads, 50% posted writes.

Figure 2: Achievable Bandwidth vs. Bank Access Method



The vault controller opens (activates) and closes (precharges) a DRAM row for each HMC packet. The row cannot be left open even if subsequent packets will be to the same page (closed page per packet policy).

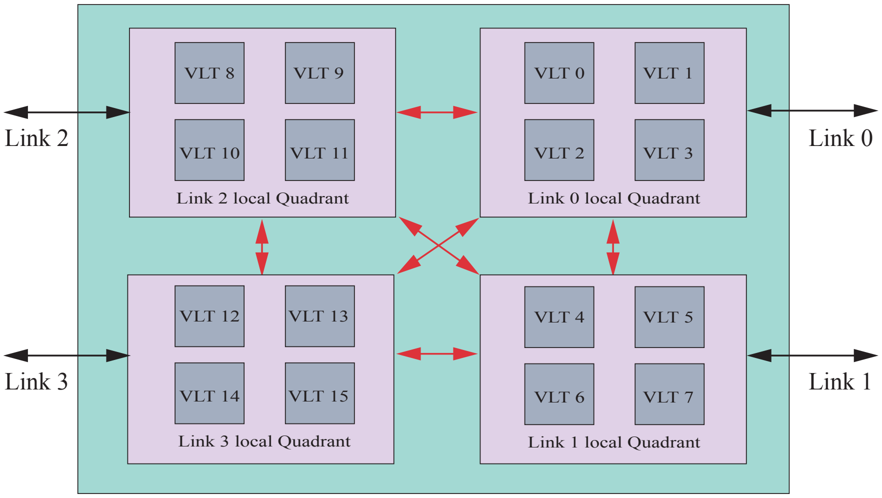
Memory accesses are effected by both packet sizes and block sizes. For example, a host needs a 128B block of data from Vault 0/Bank 0. This data can be requested in a variety of ways. One method could be to use 32-byte packets and 32-byte or larger block size (set in the Address Configuration Register). This method would require the host to issue four consecutive 32-byte data requests to Vault 0/ Bank 0. The first request will be processed normally, but the next three requests will each have to wait for Bank 0 to become available before they can be executed, because all of the requests were to the same vault and bank. These bank availability delays reduce the system throughput and increase the response latency for the requests. The protocol overhead from sending four separate packets can further reduce the system throughput. Instead, it would be more efficient for the host to send a single 128-byte request (128-byte block size is required to prevent wrap around). Sent as a single packet, this request minimizes the protocol overhead and processes the 128B transaction all at one time with no associated bank availability delays.

2.3.8 Local Vaults vs. Remote Vault Accesses

HMC's 16 vaults are divided into four quadrants, each quadrant containing four "local" vaults. Each link has a direct connection to its local quadrant, allowing that link to access the four local vaults without having to access the internal cross bar switch. Every link has access to all quadrants and all vaults of the entire device through a crossbar switch (represented by the red arrows in the two figures below) that requires multiple levels of arbitration. Accesses to vaults outside of the local quadrant are referred to as remote vault accesses. Local vault accesses pass through a local quadrant switch and will have lower latency and may sustain higher bandwidth than remote vault accesses; however, this cannot be guaranteed. The actual latency and bandwidth will depend on how the HMC is accessed.

Each vault controller has a queue used to buffer requests for that vault's memory. The vault controller executes requests within that queue based on transactional efficiency (that is, setting aside a request with a bank conflict) rather than order of arrival. Therefore, responses from vault operations back to the host may be out of order. However, requests from a single external link to the same vault/bank address are executed in order. Requests from different external links to the same vault/bank address are not guaranteed to be executed in a specific order and must be managed by the host controller(s).

Figure 3: HMC 4-Link Package (31 x 31mm) Local Vaults vs. Remote Vaults

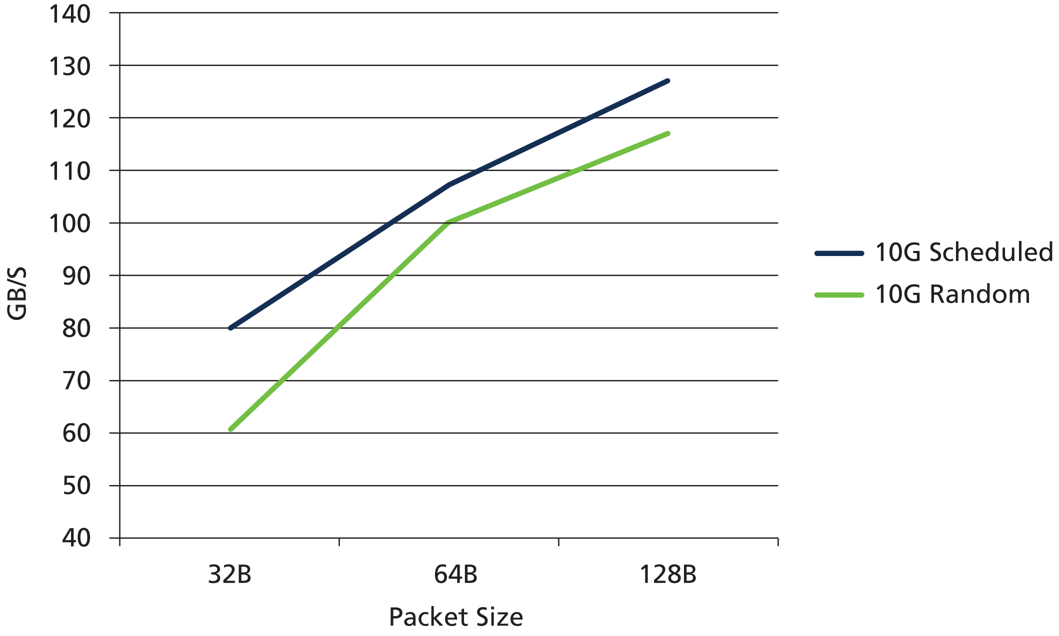


2.3.9 Scheduled vs. Random Requests

Scheduled requests are when the host has the flexibility to control both the address location and order in which requests are executed. A scheduled system should be able to adhere to all or most of the suggestions made in this document, which will help optimize the HMC performance. However, it must be understood that not all requests can be scheduled. Some requests are random in nature. Random requests will normally have lower bandwidth due to accessing data in a nonoptimized fashion. Designers can implement commonly used data storage strategies to minimize the negative impact of random requests on system bandwidth.

The following figure shows the bandwidth of scheduled vs. random requests for a 10 GB/s SerDes. The bandwidth information is for a 2GB HMC using all 4 links, 50% reads, 50% posted writes.

Figure 4: Bandwidth Impact – Scheduled vs. Random Requests



2.3.10 Refresh

The HMC memory array is comprised of volatile memory cells that need to be refreshed in order to retain stored information. REFRESH operations are controlled by each vault controller, eliminating this burden from the external host controller. The refresh rate is adjusted based on the value of the internal temperature sensor. There are two supported refresh ranges <95°C and 95°C–105°C.

In the event that an access is made to a bank that has just begun a refresh, the request would be subject to the bank availability delay discussed in a previous section. The overall impact of refresh on bandwidth depends upon the number of links being used (1, 2, 3 or 4), the size of the accesses (16B–128B) and the temperature range of operation (<95°C or 95°C–105°C). In general, refresh impact to performance is low because of the vault controller's ability to execute requests out of order, maintaining bandwidth.

The following table shows the bandwidth impact based on a device running with a 15 GB/s link speed, doing random addressing to all vaults (50% READs/ 50% WRITEs). This example highlights the relative bandwidth impact based upon the different configuration/usage options. The 2-in-1 model can be used to determine the impact of refresh for specific use cases.

Table 8: Bandwidth Impact Due to Refresh Example

Number of Active Links	Access Size	2GB (4H) DRAM0 – Temp. Sensor (T_J) Value	
		<95°C	95°C–105°C
4	32B	–1.3%	–3.9%
	64B	–1.0%	–2.1%
	128B	–0.7%	–1.4%

Table 8: Bandwidth Impact Due to Refresh Example (Continued)

Number of Active Links	Access Size	2GB (4H) DRAM0 – Temp. Sensor (T_J) Value	
		<95°C	95°C–105°C
3	32B	–1.4%	–3.5%
	64B	0.0%	–0.1%
	128B	0.0%	0.0%
2	32B	0.0%	0.0%
	64B	0.0%	0.0%
	128B	0.0%	0.0%
1	32B	0.0%	0.0%
	64B	0.0%	0.0%
	128B	0.0%	0.0%

2.3.11 Token Return Loop Time

The link input buffer in the link slave provides buffering of received FLITs while tokens are being returned to the transmitter. While tokens are being returned, additional FLITs are being transmitted and filling the link input buffer at a rate up to the maximum link bandwidth. An adequately sized link input buffer will have room to buffer enough FLITs to accommodate the time it takes for the token to return to the transmitting link master and allow it to transmit more FLITs. This is referred to as the "token return loop time."

The total delay for the token return loop time starts from the point in the link master where it is determined that there are enough tokens to transmit the next packet to the link; the first transmitted FLIT includes the following delays:

- Logical and physical block serialization time at the transmitting end of the link
- Physical and logical block deserialization time at the receiving end of the link
- The time for the link slave to forward the received FLIT and generate a token for the FLIT
  (Note that the link input buffer is empty and the FLIT does not get queued behind a previously transmitted FLIT. Any additional queuing in the link input buffer does not get added to the token return loop time.)
- Transfer of the token from the link slave to the local link master
- Time to wait in the link master from just missing embedding the token into the RTC field of the previous packet and waiting for the next tail of the longest packet
- Logical and physical serialization time for the packet with the embedded token in the transmitting link master
- Physical and logical deserialization time for the packet with the embedded token in the receiving link slave
- Time taken for this longest packet's tail to be received in the link slave, to the point of CRC check and token extraction from the RTC field
- Transfer of the extracted token from the link slave to the local link master (the original transmitter in the first delay above)
- Time taken to increment the token counter
- Time taken to determine that the token counter has enough tokens to allow the next packet to be transmitted (which is the same point where the loop started)

In order to maintain link performance, the maximum allowable delay of this token return loop time must be less than the amount of time it takes to fill the link input buffer. If the actual token return loop time is larger than the time it takes to fill the link input buffer, the requester will become starved for tokens, causing the packet stream to be throttled and resulting in a drop of bandwidth across the link.

There are two token return loops, one beginning in the host's link master, which corresponds to the HMC's link input buffer size, and the other beginning in the HMC's link master, which corresponds to the host's link input buffer size.

The HMC link input buffer size (in FLITs) is controlled by the starting number of tokens specified in the Link Input Buffer Max Token Count field in the Input Buffer Token Count register. This determines the usable depth of the input buffer and therefore determines the maximum amount of time available for token return loop time. The link input buffer in the HMC has a maximum usable size of 228 FLITs. However, there must be sufficient input buffer depth reserved for one poisoned packet (poisoned packets are described in the Packet Integrity section) in the case of a link retry. If this space is not reserved, there is the potential for a link input buffer overrun if a link retry does occur. For example, if the maximum block (packet) size available to the host is 128B (9 FLITs), the maximum usable tokens are 219 (228–9) FLIT locations. This is the space to be used and tracked with tokens by the requester. This maximum allowable size may be reduced by initializing the Link Input Buffer Max Token Count field (within the Input Buffer Token Count register) to a value smaller than 219 during HMC initialization. The Input Buffer Token Count should be set to cover the maximum total Token Return Loop time. Adding additional tokens will not increase bandwidth, but can increase the latency. The HMC 2-in-1 model, along with system validation should be used to determine the correct number of tokens to use.

Table 9: Token Update Delay for HMC Link Input Buffer

Link		219-FLIT HMC Input Buffer Full Period[1] (ns)	HMC Token Return Delay[2] (ns)	Host Allowable Delay (ns)
Bit Rate	ps/FLIT			
15 Gb/s	533.33	116.7	19.2	97.5
12.5 Gb/s	640	140.2	22.4	117.8
10 Gb/s	800	175.2	23.2	152

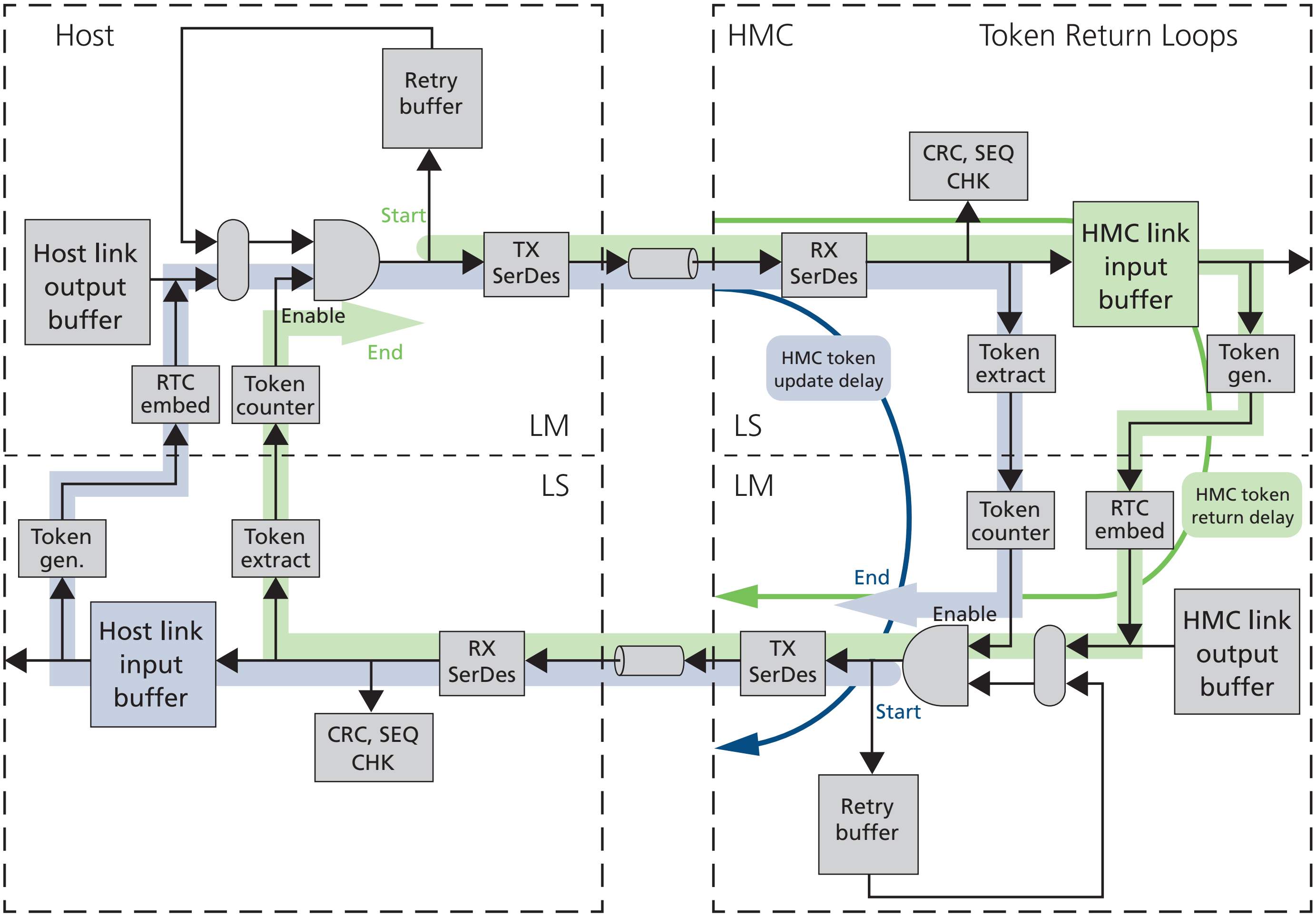
Notes:
1. If the link input buffer size is reduced with a value smaller than 219 in the Link Input Buffer Max Token Count field of the Input Buffer Token Count register, the buffer full period can be recalculated as: Buffer full period in ns = (ps/FLIT × Link Input Buffer Max Token Count)/1000.
2. "HMC Token Return Delay" is the time from when the first bit of a FLIT is received, the HMC token for that FLIT is generated (when the link input buffer is empty), the HMC token is embedded into a single-FLIT packet or TRET, to the first bit out of the HMC. The following additional delays may be added to the HMC Token Return Delay values shown in the table if multi-FLIT packets are in flight on the return link:
   2–4 FLIT packet: 1.6ns
   5–7 FLIT packet: 3.2ns
   8–9 FLIT packet: 4.8ns.

Table 10: Token Update Delay for Host Link Input Buffer

Link		HMC Token Return Delay[1] (ns)
Bit Rate	**ps/FLIT**	
15 Gb/s	533.33	19.2
12.5 Gb/s	640	22.4
10 Gb/s	800	23.2

Note: 1. "HMC Token Update Delay" is the time from when the first bit of single-FLIT packet is received, the host token from the RTC field of that FLIT is extracted, the token counter in the link master is updated, the next FLIT to be transmitted to the SerDes TX is allowed, and the SerDes TX time to the first bit of the FLIT is transmitted. The following additional delays may be added to HMC Token Update Delay values shown in the table if the host token is extracted from the RTC field of multi-FLIT packets being received from the host:
2–4 FLIT packet: 1.6ns
5–7 FLIT packet: 3.2ns
8–9 FLIT packet: 4.8ns.

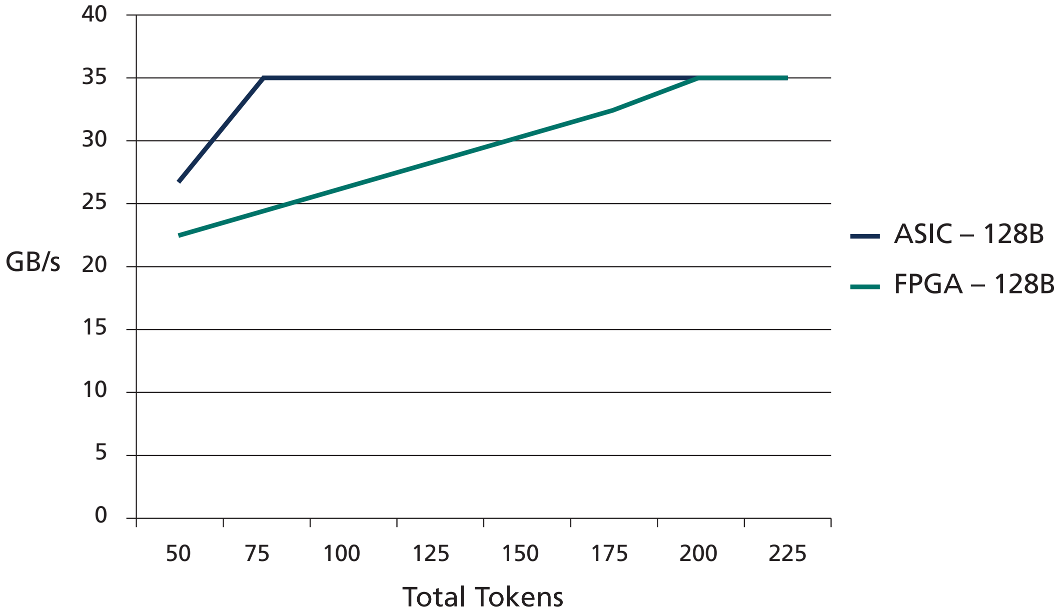
Figure 5: Token Return Loops



2.3.12 Token Count Impact

The following figure shows the effect that limiting the token count has on the DRAM bandwidth. There are two different scenarios shown in the graph: 1) An ASIC-based host using 1 link, with 50% reads, 50% non-posted writes and a 15 Gb/s SerDes rate. 2) An FPGA-based host using 1 link, with 50% reads, 50% non-posted writes and a 10 Gb/s SerDes rate. The FPGA-based host requires more tokens before the bandwidth begins to saturate, because it has a longer host delay than the ASIC-based host.

Figure 6: DRAM Bandwidth vs. Token Count



2.3.13 Retry Pointer Loop Time

Table 11: Retry Pointer Loop Time Implementation Example

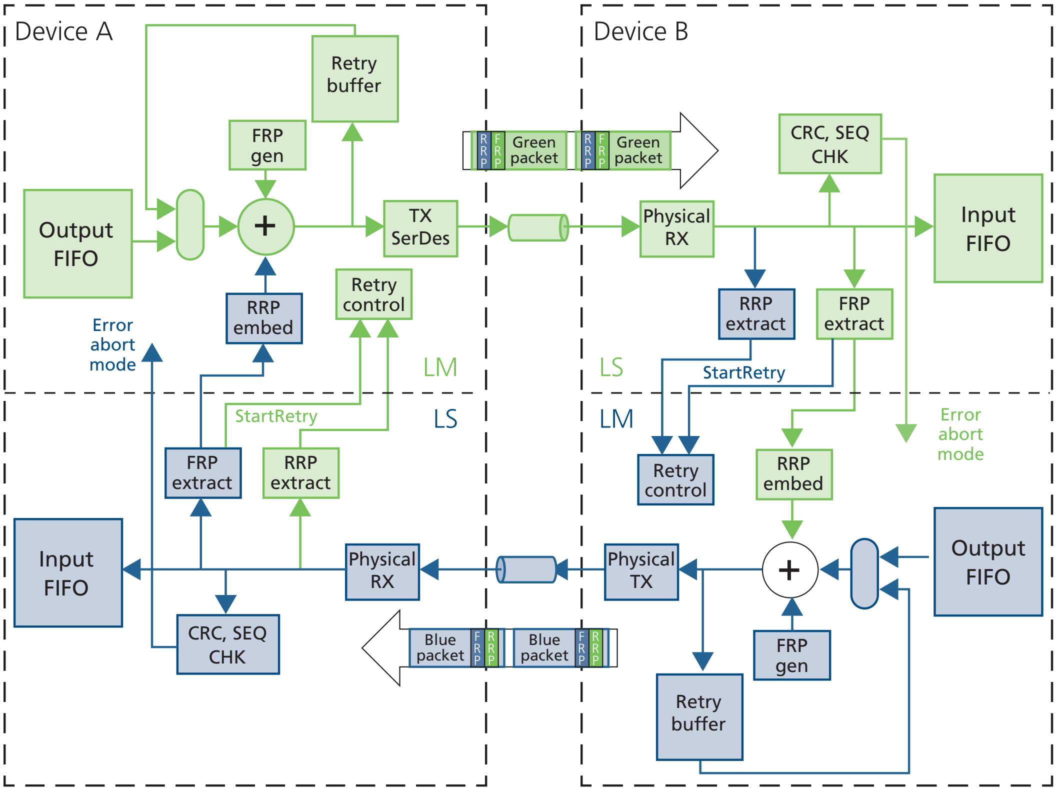
Link		Retry Buffer Full Period (ns)	HMC Delay (ns)	Host Allowable Delay (ns)	Recommended Retry Timer Value (ns)	Recommended Timeout Period Encode Value
Bit Rate	ps/FLIT					
10 Gb/s	800	153.6	26.5	127.1	> 450	4
12.5 Gb/s	640	163.8	25.9	138.0	> 500	4
15 Gb/s	533	136.4	22.3	114.2	> 400	4

The HMC protocol includes retry pointer space to support a retry buffer that can hold up to 256 FLITs. In order to maintain link performance, the maximum allowable retry pointer loop time must be less than the retry buffer full period shown in the table. If the maximum retry pointer loop time was allowed to be greater than the time it takes to fill the retry buffer, the retry buffer full condition would throttle the packet stream. The result would be NULL FLITs sent between transaction packets, degrading the performance of the link. To avoid link throttling, the host's portion of the retry pointer loop delay must be less than the "Host Allowable Delay" shown in the table.

The following implementation example highlights the measurements used to identify the total retry pointer loop time:

- The time it takes for a packet (and its FRP) to travel from a link master to a link slave, including:
  - Serialization time at the transmitting end of the link
  - Deserialization time at the receiving end of the link
- The time for the longest packet's tail to be received in the link slave, to the point of CRC check and FRP extraction
- The transfer of the extracted FRP from the link slave to the local link master
- Time to wait in the link master for just missing embedding the retry pointer into the RRP field of the previous packet and waiting for the next tail of the longest packet
- The time it takes for a packet with the embedded retry pointer to travel back to the source end of the link, including:
  - Serialization time at the transmitting end of the link
  - Deserialization time at the receiving end of the link
- Time for the longest packet's tail to be received in the link slave, to the point of CRC check and RRP extraction
- The transfer of the RRP from the link slave to the local link master (the original transmitter in the first step)

Figure 7: Link Retry Block Diagram (example)



2.3.14 Optimizing HMC Performance Conclusion

HMC is a revolutionary device that can greatly simplify the host controller design and has the capability of sustaining extremely high bandwidth.

The concepts discussed in this section should be used in conjunction with the 2-in-1 model to determine the most optimum settings and addressing scheme to use, to obtain the maximum performance of this device in your system.

2.4 Reliability Features

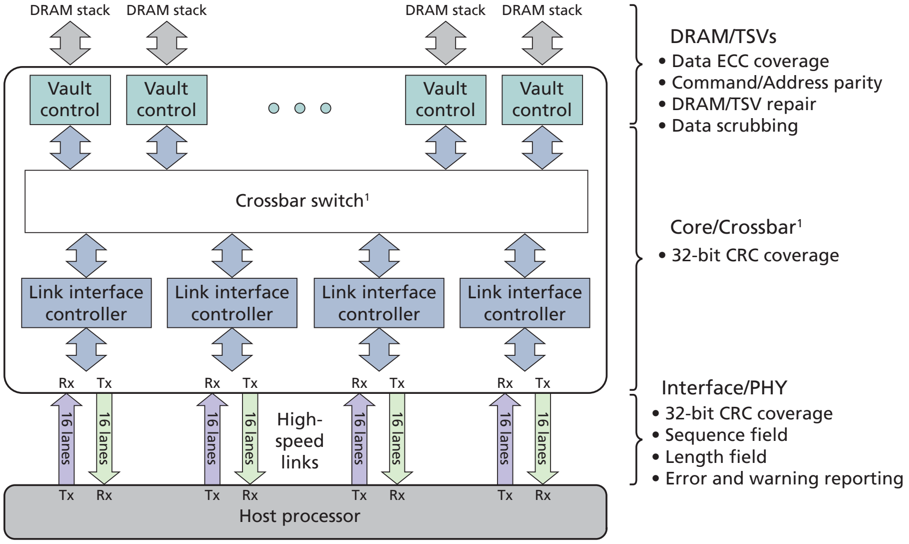
The HMC was designed with a high level of reliability, unsurpassed performance, and uptime at the forefront. HMC confronts high reliability issues and manages memory precisely without losing performance. This section introduces key architectural design aspects and explains how they are best used.

2.4.1 RAS Features

The HMC incorporates numerous reliability, availability and serviceability (RAS) features. Highlights of the RAS features available in HMC include: Cyclic redundancy check (CRC) error-detecting code protecting packet data integrity end-to-end, serialized/deserialized (SerDes) link retry (triggered by CRC errors), error correcting code (ECC) and parity protection in the DRAM, ECC-detected DRAM error scrubbing, BIST testing, and innovative DRAM and TSV repair capabilities. These RAS features work together to increase the uptime, high performance, and robustness of the HMC devices. System designs requiring high uptime, high performance, data integrity, and reliable HMC access and availability will benefit from implementing these optimized RAS features.

The figure below shows the robust end-to-end error protection mechanisms available in the HMC device. Starting at the link, communication to and from the HMC occurs through highly reliable SerDes interfaces with a topology optimized for high-speed signal integrity.

Figure 8: End-to-End Error Protection



Note: Vaults associated to each Link are considered "local vaults" and go through a local quadrant switch. Transactions on local vaults will go directly from Link interface controller through the local quadrant switch to the Vault controller. Transactions on remote vaults will use the crossbar switch to access the correct remote vault controller.

Interface/PHY

The HMC physical link (PHY) is equipped with equalization features on both the transmitter and receiver. The HMC transmitter uses feed forward equalization (FFE) with a programmable 3-tap, baud-spaced finite impulse response (FIR) driver. The HMC receiver is equipped with both an automatic gain control (AGC) amplifier with dynamic peaking control and a decision feedback equalizer (DFE). In the rare case that a data or command packet bit error occurs in the link or incoming transactional layer, it is detected by the 32-bit CRC. After the error is detected, normal packet transmission is halted. To prevent memory contamination, the associated command or data is not executed; instead, the packet is poisoned or dropped, and a retry is requested.

The retry sequence ensures both ends of the failing link are functional before retransmitting all packets in the retry buffer. This ensures that no packets are dropped during a link failure.

Other transactional layer mechanisms ensuring packet integrity across the link include duplicate packet lengths and sequence checks. Within the header of each packet is a length (LNG) and duplicate length (DLN) field. The LNG field is set according to the number of FLow control digITs (FLITs) within the packet being transferred (1 to 9). This enables the parser at the receiving side of the link to identify the correct location of various fields within the packet. Because this function is crucial to proper operation of the link, the DLN field is included to provide additional packet integrity. If the LNG and DLN fields are not equal at the receive side of the link, the packet is dropped and a retry sequence is initiated.

The error status bits within the tail of each response packet provide error and warning reporting to the host. This gives the host visibility to the health of the HMC. Some of these status updates are temperature alerts, DRAM vault array errors, protocol errors, and fatal errors where execution to the HMC is halted.

Core/Crossbar

Transactional packets received through the external protocol will include unique identification numbers called TAGs to track each transaction. After external packets have been verified as complete and valid, the packet contents are routed to the applicable destination within the HMC. Packets with a destination to a local vault (vaults found in the same quadrant as the link which initiated the transactional packet) will travel from the Link interface controller through a local quadrant switch to the corresponding local Vault controller. While packets with a destination to a remote vault (all vaults in the quadrants which are outside the quadrant of the link which initiated the transactional packet) will travel from the link interface controller through the crossbar switch to the corresponding remote vault controller. As packets are parsed and routed internally, each will incorporate a CRC code to ensure transactions continue without errors.

Before access to the memory location occurs at the vault controller, a final CRC check is performed to ensure no transient errors corrupted the packet on its path through the logic chip. Upon a good CRC check, the vault controller issues a command to the memory location. The command and address signals are also protected by parity bits. Detection of a parity error at the destination bank initiates a retry of the command by the vault controller. For response packets that are to be sent back to the host, the vault controller will generate a new CRC code prior to transmitting packets back to the link master. The link master will then verify the CRC for the packet from the vault controller, embed the token return and the retry pointer fields, and generate a new CRC code to be included in the response packet returning to the host. The process is designed to immediately halt on any error and prevent faults from propagating.

DRAM ECC and Scrubbing

Each 16-byte data word written to the DRAM and read from the DRAM is protected by a 16-bit ECC field that provides correction of single-bit DRAM errors (SBE), detection of double-bit DRAM errors (MUE), and single bit errors on the TSV bus between the DRAM and logic layer.

There are three main ways in which ECC is used. First, during every READ command to the HMC, the ECC of the DRAM contents is checked by the vault controller, and if an SBE is detected the vault controller will correct the data in the read response packet back to the host. If an MUE is detected by ECC during a READ cycle at the vault controller the response packet of the specific demand request will indicate that an MUE has occurred and the data in the response packet is not valid. Second, during that same READ cycle, if Demand Scrubbing is enabled, the contents of the DRAM will similarly be corrected to reduce the risk of additional soft errors causing the SBE to become an multiple uncorrectable error (MUE). Third, the patrol scrubbing feature, when enabled will gradually parse through all the DRAM during REFRESH cycles, checking the ECC and correcting the DRAM contents if SBEs are detected. If data ECC correction disable is optionally set in the Vault Control Configuration register settings, neither demand scrubbing nor patrol scrubbing will be enabled. As long as patrol scrubbing is enabled, SBE failing locations are tracked for persistence. If SBEs are detected at a particular DRAM address and are beyond a specific threshold count, they are considered a persistent or hard error and will be logged for repair. At that point, the HMC will dynamically allocate a replacement resource for the hard error (see the SBE Repair section below for more details). Operation of the HMC can continue without the permanent repair portion of the SBE repair (requiring a power/reset cycle); however, there are some limitations to the number of dynamically allocated resources available for SBE errors before a power or cold-reset cycle would be required to enable the NVM logged permanent SBE repairs and free up the dynamically allocated repair resource.

2.4.2 Repair

Proprietary DRAM repair at the factory is not a new concept and has become ever more important as DRAM array sizes increase and DRAM process geometries shrink. However, the ability to make repairs in the field is new and innovative. Each HMC device will go through the same comprehensive factory testing, burn-in, and reliability monitors as traditional DRAM. One of the most advanced features and innovative concepts of the HMC is the way it detects and repairs weak or faulty DRAM cells, repairs/replaces unreliable TSV connections, and provides ongoing HMC health status to the host controller.

While the ECC and parity checks discussed above detect and correct errors that are correctable, it is also necessary to repair reoccurring correctable SBE errors. SBE errors, if not corrected can lead to an MUE error if a second SBE error occurs within the same word. It is also important to have a means to recover from MUE error events. HMC provides four different ways to repair a faulty element: BIST with repair, SBE repair, MUE repair, and manual repair. See Repair Usage Table below for settings to enable the appropriate repair features needed.

2.4.3 DRAM Built-In Self Test and Repair

DRAM built-in self test (BIST) consists of using built-in algorithms implemented in hardware and software within the device to test the integrity of the DRAM array and TSVs. Testing is under autonomous control after the operation is initiated by the host through the use of External Request Interface (ERI) commands. Micron determines the breadth of the test suite and the range of algorithms included within the DRAM BIST ERIs.

Use of the DRAM BIST ERI commands is optional but recommended, particularly after significant temperature fluctuation, such as that experienced during manufacturing and attachment to a printed circuit board. The DRAM BIST ERI commands are runtime commands and may only be loaded with the I^{2}C or JTAG sideband interface after the start bit of the INIT Continue ERI has been cleared (0). Access with in-band mode read requests must occur after the transaction layer has been initialized. The DRAM BIST ERI commands can be executed with a repair option enabled or not enabled.

When the DRAM BIST with repair ERI sequence is executed and faults are found, if the Disable NVM Write bit is not set, repairs will be performed. See Built-In Self Test and Repair in the Register Definitions section for an in-depth description of the modes, options, and uses of DRAM BIST.

2.4.4 SBE Repair

SBE repair is a RAS feature meant to ensure longevity and longer uninterrupted runtime of the HMC devices. Single bit errors (SBEs) are automatically corrected by ECC and the corrected data is sent in the read response packet. When an SBE repeats at the same address multiple times and exceeds a rate threshold the HMC will begin an SBE repair. There are two aspects of an SBE repair, SRAM remapping and NVM based DRAM repairs. The HMC remaps the SBE address to a temporary SRAM resource, storing the last contents at the newly allocated SRAM resource. This is accomplished without stopping customer system traffic. The HMC records the SBE address into the repair table and the NVM log. After a subsequent power-on or cold reset cycle, a fully tested redundant DRAM resource is used in place of the SBE address DRAM, freeing up the dynamic SRAM resource to be used again if necessary.

If the Disable NVM Write bit is set when the SBE repair mechanism is writing to NVM, the write to the repair table and the log will both be silently discarded.

2.4.5 MUE Repair

MUE repair is a RAS feature that replaces existing bad DRAM elements in the HMC device with good DRAM elements. When a multi-bit uncorrectable error (MUE) is detected during a READ operation, the host is notified via the MUE ERRSTAT in the read response packet. When an MUE is detected during patrol scrubbing, the host is notified via the MUE ERRSTAT in an error response packet. If MUE repairs are enabled, either event will trigger a repair to be entered in the repair table and NVM log. After a subsequent power-on or cold reset cycle, a fully tested redundant DRAM resource is used in place of the faulty resource. This is known as MUE repair.

If the Disable NVM Write bit is set when the MUE repair mechanism is writing to NVM, the write to the repair table and the log will both be silently discarded.

2.4.6 Manual Repair

If the address of a DRAM array defect is known, it is possible to manually repair the DRAM array by using the Manual DRAM Repair ERI. A manual repair may be desirable if testing outside of BIST has indicated a failure that has not been repaired by SBE repair, MUE repair, or DRAM BIST. Prior to executing the Manual DRAM Repair ERI the user should verify the following:

1. Whether or not a repair has already been done at this address location by reading the NVM repair log.
2. That there are repair elements available by executing the DRAM Repair Health Status ERI.

3. That the Disable NVM Write bit is set to 0.

If the failing address has already been repaired, or there are no repair elements available or if the Disable NVW Write bit is set to 1, it will block a repair from taking place. After completion of the Manual DRAM Repair ERI sequence, a cold reset or power cycle is required before any other operations can be carried out. See the DRAM Repair Health Status ERI sequence found in the Register Definitions section.

2.4.7 Repair Combinations and Usage Cases

The combination table below, Repair Usage Table, highlights which configuration register bit settings determine the optimal HMC repair state and which HMC repair features are being used to obtain the desired results. ECC is used to detect MUE and SBE events. Detection of MUE and SBE is required in order for MUE and SBE repairs to take place. SBE repairs are prevented when patrol scrubbing is disabled. See the Vault Controller Register definition in the Register Definitions section.

Table 12: Repair Usage Table

Demand Scrub	Patrol Scrub	SBE Repair	MUE Repair	Scrub	Repair
1	1	1	1	Demand and Patrol	Both SBE and MUE
0	1	1	1	Patrol	Both SBE and MUE
1	0	1	1	Demand	MUE
0	0	1	1	None	MUE
1	1	0	1	Demand and Patrol	MUE
0	1	0	1	Patrol	MUE
1	0	0	1	Demand	MUE
0	0	0	1	None	MUE
1	1	1	0	Demand and Patrol	SBE
0	1	1	0	Patrol	SBE
1	0	1	0	Demand	None
0	0	1	0	None	None
1	1	0	0	Demand and Patrol	None
0	1	0	0	Patrol	None
1	0	0	0	Demand	None
0	0	0	0	None	None

Note: 1. The recommended setting is to enable demand scrubbing, patrol scrubbing, SBE repair and MUE repair.

2.5 Nonvolatile Memory Usage

The HMC uses nonvolatile memory (NVM) to contain the internal firmware (FW) and the factory DRAM repair data specific to each individual HMC. The HMC's internal processor also writes to this NVM during normal run time to track SBE repairs, MUE repairs, temperature logs, error logs, and so on. The NVM Log Read ERI (see Register Definitions for details) can be used to read out the data that has been written in this fashion. Since the NVM space is one-time programmable (OTP) per word, it is important that the NVM not become corrupted. The information in this section is meant to aid in understanding how the NVM is used during runtime and how to protect the NVM from corruption.

In the field, NVM will be written for the following reasons:

- Storing MUE repairs and permanent SBE repairs
- Storing repairs from DRAM BIST With Repair ERI
- Storing repairs from Manual DRAM Repair ERI
- FW patching from the Patch Apply ERI
- Logging of all repairs and over-temperature events in the log table
- Logging of fatal error events

2.5.1 Logging Maximum Temperature Events

The initial temperature thresholds for logging a maximum temperature event are upper limits of the operating junction temperature ranges. (105°C DRAM/110°C logic). If the temperature sensor on the logic or DRAM die are ever measured above those levels, the measured values are stored in the NVM log space. After each logged event, the temperature that was logged plus 2°C becomes the new threshold for subsequent logging. This means that the minimum step size between logged values will be 2°C; however, if the temperature ramps quickly enough, the step size can be larger.

2.5.2 Avoiding NVM Corruption in HMC

If the HMC is writing to NVM at the same time that power is removed from the device or a cold reset occurs (asserting P_RST_N), there is a remote possibility that the two events coincide and this could result in corruption of the record being written to the NVM. The HMC's internal processor attempts to detect a power loss and prevent this from occurring. The Shutdown ERI (see Register Definitions for details) avoids this entirely by ensuring that all such events have completed safely before indicating that it is safe to cold reset or power down.

Do not power-down the HMC or assert P_RST_N during DRAM BIST With Repair, Manual DRAM Repair or Patch Apply ERIs. Powering down or cold resetting the HMC during a Patch Apply ERI can corrupt the device firmware, potentially rendering the part unuseable.

2.5.3 NVM Corruption Risk Evaluation

Risks of NVM corruption occuring under these circumstances:

- Shutdown ERI run prior to power-down or cold reset (asserting P_RST_N): **No Risk**
- Allow the HMC to detect a power loss during normal operation without ERIs being executed: **Low Risk**
- Asserting P_RST_N without ERIs running and without executing Shutdown ERI: **Low Risk**

- Allow the HMC to detect a power loss during execution of BIST, Manual Repair, Patch Apply ERIs: **High Risk**, though deterministic and easy to avoid.

2.5.4 Potential Consequences of an NVM Corruption

NVM corruption events have different consequences depending on which NVM space they occur in. NVM corruption events can occur in the Log space, the repair space or the patch space. If the NVM is corrupted in the log space, there is a risk that one or more log entries will not be decoded appropriately when executing the Log Read ERI. It is expected that the device will otherwise be able to operate normally. If the NVM is corrupted in the DRAM repair space, there is a risk that an address may be falsely repaired, while the address intended for repair would not. It is expected that the device will otherwise be able to operate normally, and future repairs would happen normally.

In the case of a Patch Apply ERI being interrupted with a power-down or P_RST_N event, the risk is far more severe. A firmware patch is essentially an extension of the firmware itself so any corruption of a patch would be like a corruption of the firmware. The consequences of this are not deterministic and include fatal errors, non-responsive devices, and devices that are not able to be further patched in the future.

2.5.5 Alternative Method to Avoid NVM Corruption

The following is not the recommended approach but is presented as an alternative workaround option.

This alternative method to eliminate risk of NVM corruption due to power-down or cold-reset when the Shutdown ERI cannot be run, is to disable writing to the NVM altogether. This can be accomplished using the NVM write disable configuration register, which can be set/cleared by the host during initialization. Again, changing this value during run time is possible but not recommended.

If NVM write is changed from enabled to disabled in the middle of writing to NVM, the write would complete writing the word in flight to NVM, but not write a second word or subsequent writes. This can stop an FW Patch Apply ERI part way through a patch, typically resulting in a device that cannot accept any future patches and may not be functional after a cold reset. This also can cause a two-word NVM log entry to have only the first word written, causing some potential confusion in the decoding of the NVM Log Read ERI results, which impacts the logging of a patrol scrub or demand scrub during repair but not the repair itself.

NVM write disable blocks all write functions to the HMC NVM, including:

1. Storing MUE repairs and permanent persistent SBE repairs
2. Storing repairs from DRAM BIST With Repair ERI
3. Storing repairs from Manual DRAM Repair ERI
4. FW patching from the Patch Apply ERI
5. Logging of all repair events, and over-temperature events in the log table
6. Logging of fatal error events

One technique to gain back the lost functionality of number 1 listed above is to enable NVM writes only during the BIST algorithm and to run the BIST algorithm periodically throughout the life of the product, using either the HMC internal BIST engine or an externally generated pattern generator memory test engine. Use the Shutdown ERI prior to cold resetting the HMC to turn off the NVM write.

Another technique to gain back the lost functionality of number 1 listed above is to enable NVM writes and allow patrol scrub to function on typical transactions for 10 minutes, then disable NVM writes until the next cold reset. Disabling NVM write has a small probability of occurring in the middle of a multi-word NVM log entry, causing future confusion in the interpretation of the NVM log entry.

For number 2 in the list above, performing BIST With Repair at least once after board build/reflow as part of the manufacturing flow is strongly recommended.

The loss of function of numbers 2, 3, and 4 (and the repair logging in number 5) above could be made functionally negligible if the system temporarily enables NVM write immediately prior to those events and disables it immediately afterwards.

For number 6, there is no mitigation — turning off NVM write will disable all logging of fatal errors, which in turn may make debugging problems difficult.

3 Chapter 3 – Board Design Guidelines

Managing a complex device such as the HMC requires careful attention to logical design practices for thermal, power, and routing strategies. This section contains a best-practices checklist that we strongly recommend following for efficient integration of the HMC, includes references to related information in other chapters, and discusses the design tools available from Micron, such as the power calculator, thermal models, and power distribution network (PDN) simulation models.

3.1 Board Design Checklist

3.1.1 Schematic Creation

1. Create HMC schematic symbol using the 4-link pin map files available on micron.com
2. Reference Pin Connection Guide (page 37)
   a. Ensure that all HMC signal connections are compliant with this section
3. Reference I2C Interface (page 99)
   a. Ensure that I^2C design is compliant to NXP I^2C-bus specification (UM10204)
   b. An I^2C header is required to accommodate the connection of Micron's debug cable (Sidewinder)
4. Design initial power delivery network (PDN)
   a. Design the initial decoupling capacitor networks
   b. Reference PDN Simulation Kit (page 44).
   c. Verify the V_{DD}, V_{DDM}, V_{TR} and V_{TT} supply designs using the PDN simulation kit available on micron.com.
      1. Adjust board/decoupling designs as necessary in order to meet the required impedance mask for each supply
   d. Verify that V_{CCP} and V_{DDPLL} supplies are decoupled as recommended in the PDN simulation kit instructions
   e. Verify that the voltage rails ramp up in the sequence specified in the data sheet

3.1.2 Board Layout

1. Design signal routing and layout
   a. Reference Board Routing Guidelines (page 68)
2. Finalize signal integrity (SI) analysis and design
   a. Complete SI analysis of HMC SerDes, host SerDes, and channels using IBIS AMI and S-parameter models available on micron.com
   b. Complete SI analysis of HMC low-speed signals (REFCLK, I^2C, and so on) using the HMC GPIO IBIS model available on micron.com
3. Finalize PDN design by including the PCB layout extractions in the PDN simulations
4. Reference TN-00-15: Recommended Soldering Parameters
   a. Solder balls are SAC405 and typical industry soldering conditions for Pb-free assemblies are acceptable
   b. Vapor phase or convection reflow is acceptable
   c. The thermal mass of HMC is significantly larger than other Micron products, and the thermal reflow profile must be developed to account for this in-

creased mass along with the requirements of all other components on the assembly.

3.2 Pin Connection Guide

This section provides a consolidated list of HMC pins and their associated connection recommendations. This information should be used when creating or reviewing schematic designs in order to ensure proper connectivity between the HMC and other devices on the printed circuit board.

For complete pinout information, see the HMC Gen-2 data sheet.

3.2.1 Schematic Review Checklist and Pin Connection Guidelines

Table 13: Schematic Review Checklist and Pin Connection Guidelines

Pin/Group Name	Pin Type	Pin Description	Connection Guidelines
Link Interface			
LxRXP[n], LxRXN[n]	Differential input	Receiving lanes. *x* is defined as the link number, and *n* is defined as the lane number within a link.	Connect to corresponding host transmit lanes. Lane polarity can be inverted within an RX pair, and lane order can be reversed within a link if needed for routing requirements, because both lane polarity and lane order are detected during initialization. See the HMC data sheet for details. Receiver inputs are internally AC coupled, but can tolerate 0.1µF external AC coupling capacitors if desired. Note that if using two of four links, Micron recommends avoiding the use of both links on the same side of the device (that is, avoid using Link0 and Link2 together or Link1 and Link3 together). Unused lanes should be left unconnected (floating) on the PCB. If using half-width links, lanes 0–7 must be used, and lanes 8–15 are considered DNU.
LxTXP[n], LxTXN[n]	Differential output	Transmitting lanes. *x* is defined as the link number, and *n* is defined as the lane number within a link.	Connect to corresponding host receiver lanes. Coupling requirements are a function of receiving device requirements, based upon common mode voltage and internal AC coupling capabilities of the receiver. Unused lanes should be left unconnected (floating) on the PCB. If using half-width links, lanes 0–7 must be used, and lanes 8–15 are considered DNU.

Table 13: Schematic Review Checklist and Pin Connection Guidelines (Continued)

Pin/Group Name	Pin Type	Pin Description	Connection Guidelines
LxRXPS	Input: 1.5V V_{DDK} referenced	Power-reduction input. *x* is defined as the link number.	Unused or disabled links should have corresponding LxRXPS driven LOW or pulled LOW with 1k–10kΩ pull-down resistors to V_{SS}. If LxRXPS is actively driven to manage power during normal operation, refer to the Power State Management section of the HMC data sheet for important logic details. For host to single cube topologies, LxRXPS should be driven by the host at V_{DDK} levels (see the HMC data sheet for absolute maximum levels). For pass-through link (chained) topologies, LxRXPS should connect to the corresponding link partner's LxTXPS pins on a per-link basis. All LxRXPS pins must be valid prior to the INIT CONTINUE command.
LxTXPS	Output: 1.5V V_{DDK} referenced	Power-reduction output. *x* is defined as the link number.	For host to single cube (nonchained) topologies, LxTXPS should connect to the host to track the power save status of the cube. For pass-through link (chained) topologies, LxTXPS must connect to the corresponding link partner's LxRXPS pins on a per-link basis. Unused or disabled links may have corresponding LxTXPS pin floating or connected to another device input for status monitoring.
FERR_N	Output: Open drain	Fatal error indicator.	Connect to the host for detection of fatal errors using a 2.2k–10kΩ pull-up resistor to V_{DDK}. It can also be connected to LED for visual detection of errors. FERR_N can be wire-OR'd among multiple HMC devices. $R_{ON} \sim 300\Omega$; R_{OFF} ~ 10kΩ. Use the HMC Gen2 GPIO IBIS model if needed to run SI simulations and/or to determine actual pull-down drive.
Clocks and Reset			

Table 13: Schematic Review Checklist and Pin Connection Guidelines (Continued)

Pin/Group Name	Pin Type	Pin Description	Connection Guidelines
REFCLKP, REFCLKN	Differential input	Reference clock.	Connect to a low jitter reference clock source. It can be AC coupled LVDS or LVPECL, or DC coupled differential HSTL. Ensure that REFCLKSEL is set properly to select input standard and coupling methodology. Ensure that jitter specifications are adhered to as stated in the HMC data sheet. Ensure that the clock source (oscillator) is identical (0 PPM offset) in frequency to the reference clock used by the host SerDes. If a frequency other than those allowed by the HMC is required for host SerDes, a PLL may be used to generate the required frequency as long as the original clock source (oscillator) is shared between host and HMC. Ensure that the clock source is toggling at the proper frequency prior to de-assertion of P_RST_N. The frequency can be 125 MHz, 156.25 MHz, or 166.67 MHz, but REFCLK_BOOT must be set appropriately. Note that REFCLK is internally terminated in the device and should not be terminated on the board.
REFCLK_ BOOT[1:0]	Input: 1.5V V_{DDK} referenced	Bootstrapping pins that enable autonomous PLL configuration by selecting the proper reference clock frequency.	Tie the pins DC HIGH or DC LOW (optionally with 1k–10k pull resistors) to match the reference clock frequency being used. REFCLK_BOOT[1:0] reference clock frequency: 00 = 125 MHz, 01 = 156.25 MHz, 10 = 166.67 MHz.
REFCLKSEL	Input: 1.5V V_{DDK} referenced	Determines on-die termination for REFLCLKP/N.	Connect through a 1k–10kΩ pull-up resistor to V_{DDK} if REFCLKP/N are AC coupled. Connect to V_{SS} or through a 1k–10kΩ pull-down resistor to V_{SS} if REFCLKP/N are DC coupled.
P_RST_N	Input: 1.5V V_{DDK} referenced	HMC cold reset.	Connect to the host controller in order to control reset. The pin must have a 1k–10kΩ pull-down resistor in order to ensure logic LOW during power system ramp up, but must be driven HIGH by the host after REFCLK is valid.
JTAG Interface			
The JTAG interface can be left unused if desired, though either JTAG or I^2C is required for side-band transactions. Connecting either the JTAG or I^2C interface to a cable-accessible header is required for effective debugging. The JTAG interface can be accessed after ^tINIT.			
TMS	Input: V_{DDK} referenced	JTAG test mode select.	Can be connected directly to a debug header or as part of larger chain. If in a chain, connect them to other TMS signals in the chain. A buffer may be required when many devices are chained together and driven by a single TMS source. Be sure to verify that input thresholds and timing will be met (refer to the HMC data sheet for specifications). Connect with a 1k–10kΩ pull-up resistor to V_{DDK} if used, otherwise it can be left open.

Table 13: Schematic Review Checklist and Pin Connection Guidelines (Continued)

Pin/Group Name	Pin Type	Pin Description	Connection Guidelines
TCK	Input: V_{DDK} referenced	JTAG test mode clock.	Can be connected directly to a debug header or as part of larger chain. If in a chain, connect them to other TMS signals in the chain. A buffer may be required when many devices are chained together and driven by a single TMS source. Be sure to verify that input thresholds and timing will be met (refer to the HMC data sheet for specifications). Connect with a 10kΩ pull-down resistor to V_{SS} if the JTAG interface is used. If the JTAG interface is not used, the pull-down resistor can be 1k–10kΩ.
TDI	Input: V_{DDK} referenced	JTAG test data-in.	Can be connected directly to a debug header or as part of a larger chain. Connect to the TDO of the previous device in the chain or directly to the debug header if the HMC device is the only or first device in the chain. Connect with a 10kΩ pull-up resistor to V_{DDK} if used, otherwise it can be left open.
TDO	Output: V_{DDK} referenced	JTAG test data-out.	Can be connected directly to a debug header or as part of a larger chain. Connect to the TDI of the next device in the chain or directly to the debug header if the HMC device is the only or last device in the chain. If JTAG connections are not used, it can be left open.
TRST_N	Input: V_{DDK} referenced	JTAG test reset.	Connect to the debug header or JTAG controller. TRST_N should have a 1k–10kΩ pull-down resistor to V_{SS} in order to assure a logic LOW value during power system ramp up. The board JTAG controller will need to be able to drive the pin HIGH to de-assert JTAG reset for JTAG operations. If the JTAG interface is always unused in the system, you may tie TRST_N directly to V_{SS}.
I^2C Interface			
The I^2C interface can be left unused if desired, though either JTAG or I^2C is required for side-band transactions. Connecting either the JTAG or I^2C interface to a cable-accessible header is required for effective debugging. The I^2C interface can be accessed after ^tINIT.			
SCL	Input: V_{DDK} referenced	I^2C clock.	Connect to I^2C master or debug header. The pin must be pulled up to V_{DDK}; the resistor value will depend upon bus loading and desired frequency. Consider the loading of the cable to be used, including whether or not pull-ups are present on cable adapter.
SDA	Bidirectional: Open drain V_{DDK} referenced	I^2C data.	Connect to I^2C master or debug header. The pin must be pulled up to V_{DDK}; the resistor value will depend upon bus loading and desired frequency. The HMC Gen2 GPIO IBIS model may be used to derive the appropriate value through SI simulations. Consider the loading of the cable to be used, including whether or not pull-ups are present on cable adapter.

Table 13: Schematic Review Checklist and Pin Connection Guidelines (Continued)

<table>
<tr><th>Pin/Group Name</th><th>Pin Type</th><th>Pin Description</th><th colspan="2">Connection Guidelines</th></tr>
<tr><td rowspan="11">CUB[2:0]</td><td rowspan="11">Input</td><td rowspan="11">User-assigned HMC identification to enable the host to map the unique location of an HMC device in a system for both I^{2}C and host-to-cube packet routing.</td><td colspan="2">The I^{2}C slave address offset is equivalent to the state of the CUB[2:0] bits. Each HMC on the I^{2}C bus must have a unique address and therefore a unique setting for CUB. The pin settings equate to the following I^{2}C addresses.</td></tr>
<tr><th>CUB[2:0]</th><th>I^{2}C Address</th></tr>
<tr><td>000</td><td>0x20</td></tr>
<tr><td>001</td><td>0x22</td></tr>
<tr><td>010</td><td>0x24</td></tr>
<tr><td>011</td><td>0x26</td></tr>
<tr><td>100</td><td>0x28</td></tr>
<tr><td>101</td><td>0x2A</td></tr>
<tr><td>110</td><td>0x2C</td></tr>
<tr><td>111</td><td>0x2E</td></tr>
<tr><td colspan="2">The HMC device will also use this value as the default value for the CUB field of the request packet header. All HMC devices in a linked chain must also have CUB pins set uniquely.
This signal should be pulled up or down through a 1k–10kΩ resistor to V_{DDK} or V_{SS} as appropriate.</td></tr>
<tr><td colspan="5"><b>Analog Pins</b></td></tr>
<tr><td>EXTRESTP/
EXTRESTN</td><td>–</td><td>Pins used to connect upper (top) precision termination calibration.</td><td colspan="2">Connect a precision (0.1%) 200Ω resistor between the EXTRESTP/EXTRESTN pins.</td></tr>
<tr><td>EXTRESBP/
EXTRESBN</td><td>–</td><td>Pins used to connect upper (bottom) precision termination calibration.</td><td colspan="2">Connect a precision (0.1%) 200Ω resistor between the EXTRESBP/EXTRESBN pins.</td></tr>
<tr><td colspan="5"><b>Reserved Pins</b></td></tr>
<tr><td>DNU</td><td>–</td><td>Do not use.</td><td colspan="2">Leave floating, do not connect to voltage or ground.</td></tr>
<tr><td>TST_V_{SS}</td><td>–</td><td>Test pins.</td><td colspan="2">Connect to V_{SS}.</td></tr>
<tr><td colspan="5"><b>Supply Pins</b></td></tr>
<tr><td colspan="5">See the PDN Simulation Kit section, for pin-specific decoupling, filtering, and bypass requirements and the HMC data sheet for supply initialization timing, sequencing, and absolute maximum requirements.</td></tr>
<tr><td>V_{DD}</td><td>Supply</td><td>Logic supply.</td><td colspan="2">Connect to a filtered/isolated 0.9V supply (see the HMC data sheet DC Electrical Characteristics section for further requirements).</td></tr>
</table>

Table 13: Schematic Review Checklist and Pin Connection Guidelines (Continued)

Pin/Group Name	Pin Type	Pin Description	Connection Guidelines
V_{DDPLLA}[3:0], V_{DDPLLB}[3:0], V_{DDPLLR}	Supply	PLL supplies.	1.2V (see the HMC data sheet DC Electrical Characteristics section for further requirements). V_{DDPLL} pins can share a common supply, but proper filtering is required. Ferrite beads may be used for this purpose. Sharing a common supply with the other 1.2V pins, including V_{TT}, V_{TR}, and V_{DDM} pins is not recommended. **Note:** V_{DDPLLA} and V_{DDPLLB} will experience higher current draw prior to link configuration than during normal operation. Please see the HMC data sheet for guidance on absolute maximum current, which should be accounted for when selecting power supplies for these pins.
V_{TT}, V_{TR}	Supply	Link transmit (V_{TT}) and link receive V_{TR} termination supplies.	1.2V (see the HMC data sheet DC Electrical Characteristics section for further requirements). Can be shared with other 1.2V pins, and use of Ferrite beads for filtering are not recommended.
V_{DDM}	Supply	Memory supply.	1.2V (see the HMC data sheet DC Electrical Characteristics section for further requirements). Can be shared with other 1.2V pins, but proper filtering is required. Ferrite beads may be used for this purpose.
V_{CCP}	Supply	DRAM wordline boost supply.	2.5V (see the HMC data sheet DC Electrical Characteristics section for further requirements).
V_{DDK}	Supply	NVM, I^{2}C, JTAG, and power management pin supply.	1.5V (see the HMC data sheet DC Electrical Characteristics section for further requirements).
V_{SS}	Supply	Ground.	

3.2.2 High Speed Serial Lanes Coupling Capacitors

This section provides additional details on the appropriate selection of AC or DC coupling topologies for the high speed serial lanes differential pairs. The HMC transmitter and receiver can be used with AC or DC coupling between the transmitter and the receiver.

HMC TX DC coupling to the host RX is a supported configuration if the host RX has a mode that uses a capacitor load at the center of the receiver termination. This is shown in the "HMC TX Driving Host RX - Example #2" figure in the HMC data sheet. If the host RX does not have this option, instead using a DC bias to establish the common-mode voltage, that common-mode voltage must be compatible to the voltage range defined in the HMC data sheet as shown in the "HMC TX Driving Host RX - Example #1" figure. This latter configuration with a common-mode bias at the host RX comes with a reduced maximum junction temperature (T_J) of the logic die to 100°C, lowered from the normal maximum of 105°C. This is because common-mode voltage mismatches can cause DC current flow between the devices, which could be a reliability risk for the HMC TX circuit at higher temperatures.

HMC TX AC coupling to the host receiver is preferred and will not reduce the operating temperature range the way DC coupling will (see above). The HMC transmitter can connect to the host RX through AC coupling. For more information, refer to the "HMC TX Driving Host RX with External AC Coupling Present" figure in the HMC data sheet.

HMC RX has on-die AC coupling caps so that AC coupling caps at the TX or on the board between the host TX and HMC RX are not necessary. Because external DC coupling is the default mode for HMC RX, this will work correctly by default. The PHY Configuration ERI uses a default selection of external DC coupling mode in the HMC RX circuit. Refer to the Host TX Driving HMC RX Without External AC Coupling figure in the HMC data sheet for information on DC coupling between host TX and HMC RX.

HMC RX supports AC coupling mode for host designs where AC coupling caps are required. Because external AC coupling is not the default, this must be configured correctly via the Phy Configuration ERI. Use the PHY Configuration ERI to select the external AC coupling mode for HMC RX. Refer to the HMC data sheet figure titled "Host TX Driving HMC RX With External AC Coupling" for AC coupling between host TX and HMC RX.

3.3 PDN Simulation Kit

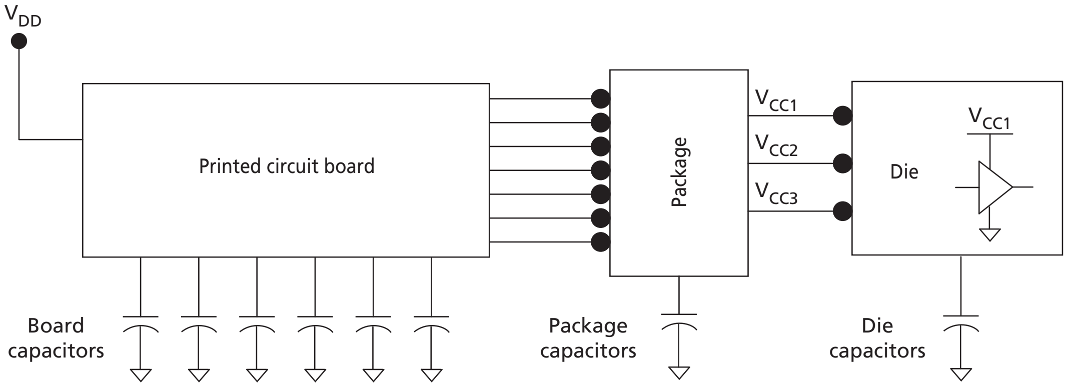
To ensure the power distribution network (PDN) design meets the device requirements, Micron provides a PDN simulation kit (available on micron.com). This section explains running the PDN simulation kit in the Keysight Advanced Design System (ADS) and will take the customer through the process of testing their own board design, analyzing the impedance profile, and comparing the results to the target impedance mask.

3.3.1 Package Considerations

The short-reach HMC Gen2 package size is a 31mm x 31mm with 1mm ball pitch. The package supports up to four links with various link configurations (up to 16 TX lanes and 16 RX lanes per link).

The many package details to consider include use placement of capacitors under and around the HMC, the in-package capacitance, the number of cubes that share the same source, the number and type of board-level decoupling capacitors, the layout of the individual power rails, and whether these power rails are shared between different device nodes. Figure 9 shows the typical PDN components discussed in this section.

Figure 9: Typical PDN Components (Board, Package, and Die)



A successful PDN provides the needed current at all frequencies without violating the specified voltage droop limits. The PDN should handle the DC average demand as well as all high-frequency demand. All current eventually comes from the regulator; however, the high-frequency portion of the current typically comes from various decoupling capacitors that have low impedance and/or low inductance and are closer to the die. All capacitors are then slowly recharged through the rest of the PDN, upstream through the PCB. Voltage fluctuations may be generated from many different sources, including the die, package, and board; interference from other parts; and inconsistencies within the power source. As shown in Figure 9, every PDN component (board, package, and die) removes some margin from the total power budget.

3.3.2 Target Impedance

The goal of PDN simulation is to achieve an overall impedance profile that meets the target impedance mask within a predefined frequency range. Each power rail will have its own set of target impedance masks based on a different number of active links and/or a different number of DRAM heights. Figure 10 shows the target impedance masks for V_{DD}. Figure 11 shows the target impedance masks for V_{DDM}. Figure 12 and Figure 13 show the target impedance for V_{TT} and V_{TR}, respectively. Figure 14 and Figure

15 show the target impedance for V_{DDPLLA} and V_{DDPLLB}, respectively. For V_{TT}, V_{TR}, V_{DDPLLA} and V_{DDPLLB}, the impedance profile should be simulated on a per link basis. Only one mask is required for V_{TT} in each link and one mask for V_{TR} in each link.

Figure 10: Target Impedance for V_{DD}

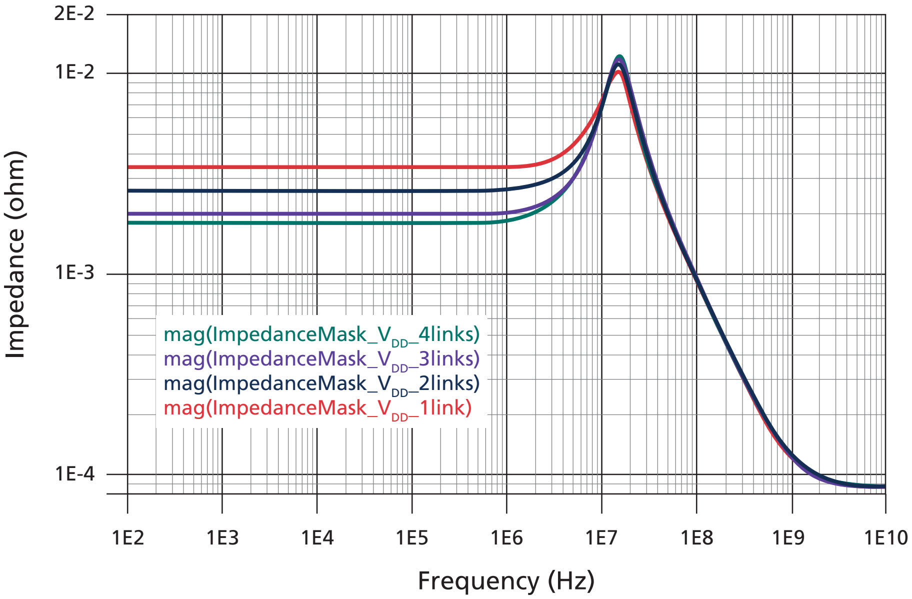


Figure 11: Target Impedance for V_{DDM}

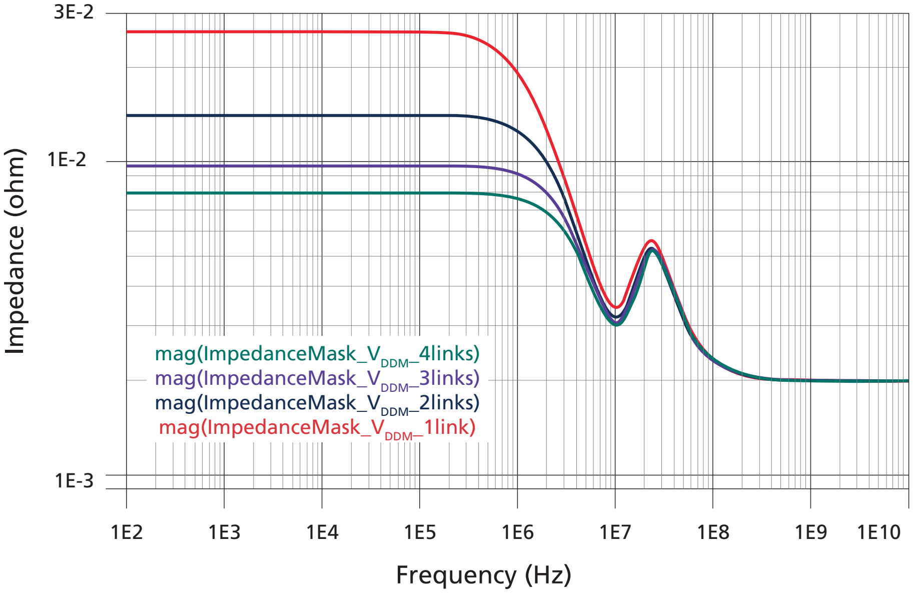


Figure 12: Target Impedance for V_{TT}

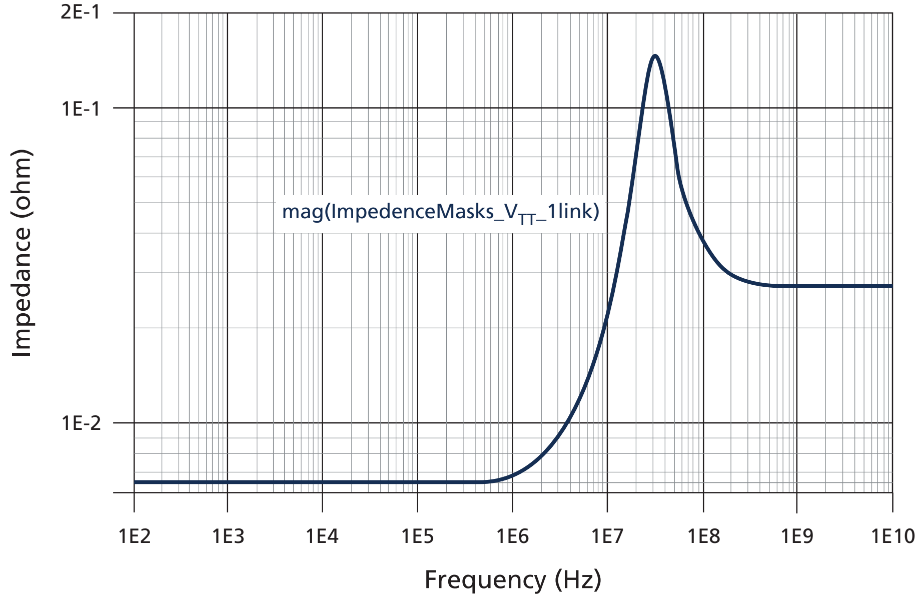


Figure 13: Target Impedance for V_{TR}

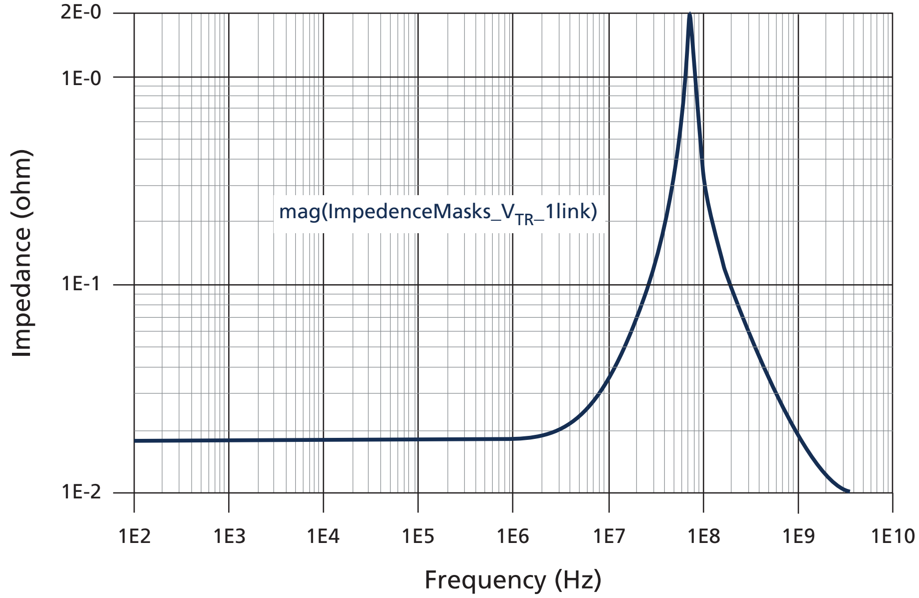


Figure 14: Target Impedance for V_{DDPLLA}

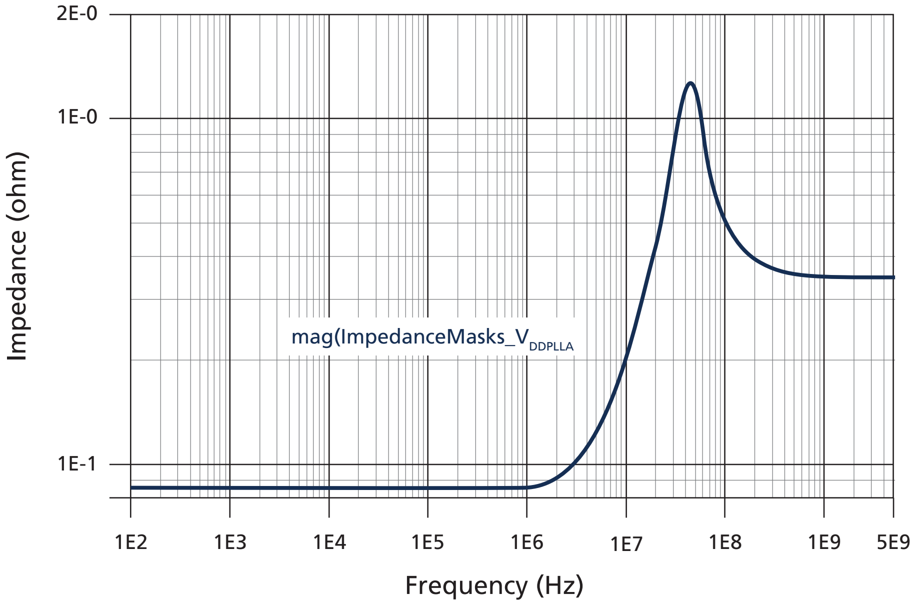
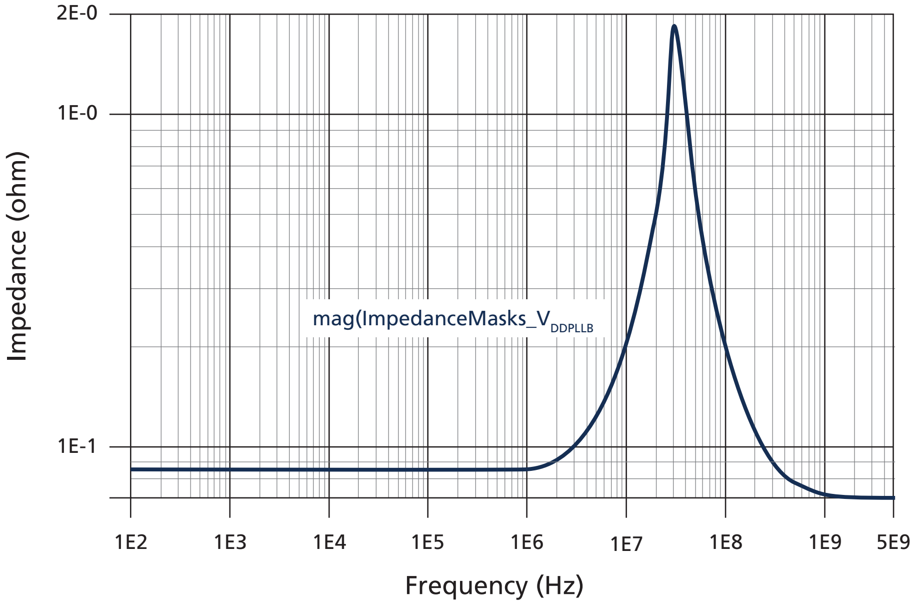


Figure 15: Target Impedance for V_{DDPLLB}

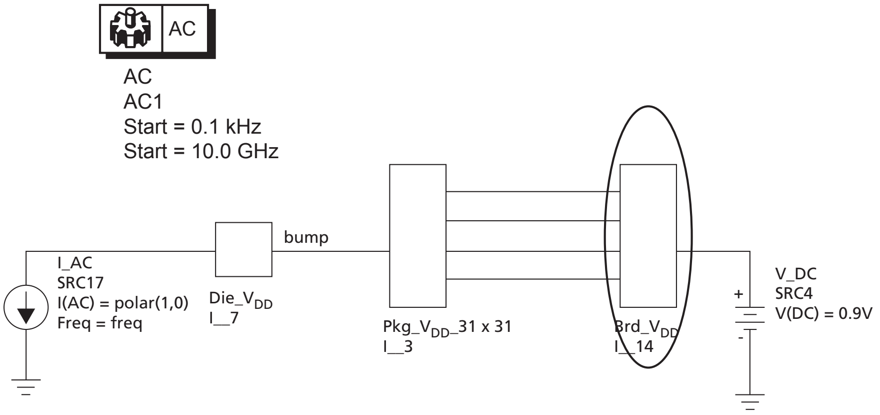


3.3.2.2 Schematics

For each power rail there is an associated schematic to run a simulation example with the corresponding PCB, the package, and the on-die decoupling.

Each schematic is composed of several blocks: AC current source, die model, package model, and the board model for the voltage rail.

Figure 16: Schematic for HMC Gen2 31mm x 31mm Package V_{DD} PDN Simulation



The AC current source is used as the stimulus to generate the impedance profile. The die model has the on-die decoupling components, and the package model contains the on-package decoupling components.

The board model is the s-parameter extraction of the PCB layout on the target board between the power source and the HMC power pin. This simulation kit uses a sample HMC board, whose extraction (s-parameter files) for each specific set of power rails is used in each schematic. ***Users of this simulation kit should replace this block with their own board model before running the simulation.***

To replace the board s-parameter file used in each schematic, select the "Brd_Vxx" block in the schematics, right-click and select "push into hierarchy," then double-click the s-parameter block shown in the new schematic and change the s-parameter file name/directory to the target one.

The sample board trace for V_{DD} rails is extracted to have 4 ports connecting to 4 BGA balls on the package model. When extracting board s-parameter models, Micron recommends grouping the package connections into 4 ports the same way the package model was extracted.

For V_{DD}, the package model ports P2, P3, P4, and P5 group the BGA balls as follows:

- P2 = N9, N10, N12, N14, R12, R14
- P3 = N16, N18, N20, N22, R16, R18
- P4 = T9, T10, T13, T15, V9, V11, V13, V15
- P5 = T17, T19, T21, T23, V17, V19, V21

For V_{DDM}, the package model ports P2, P3, P4, and P5 group the BGA balls as follows:

- P2 = M9, M11, M13, M15, P11, P13, P15
- P3 = M17, M19, M21, M23, P17, P19, P21, P23
- P4 = U8, U10, U12, U14, W8, W10, W12, W14
- P5 = U16, U18, U20, U22, W16, W18, W20, W22

For V_{TT}, the package model ports P1, P2, P5, and P6 group the BGA balls as follows:

- P1 = J10, J13, K9, K11, L10, L14
- P2 = J17, K21, K23, L18
- P5 = AA8, AA10, AB14, Y13

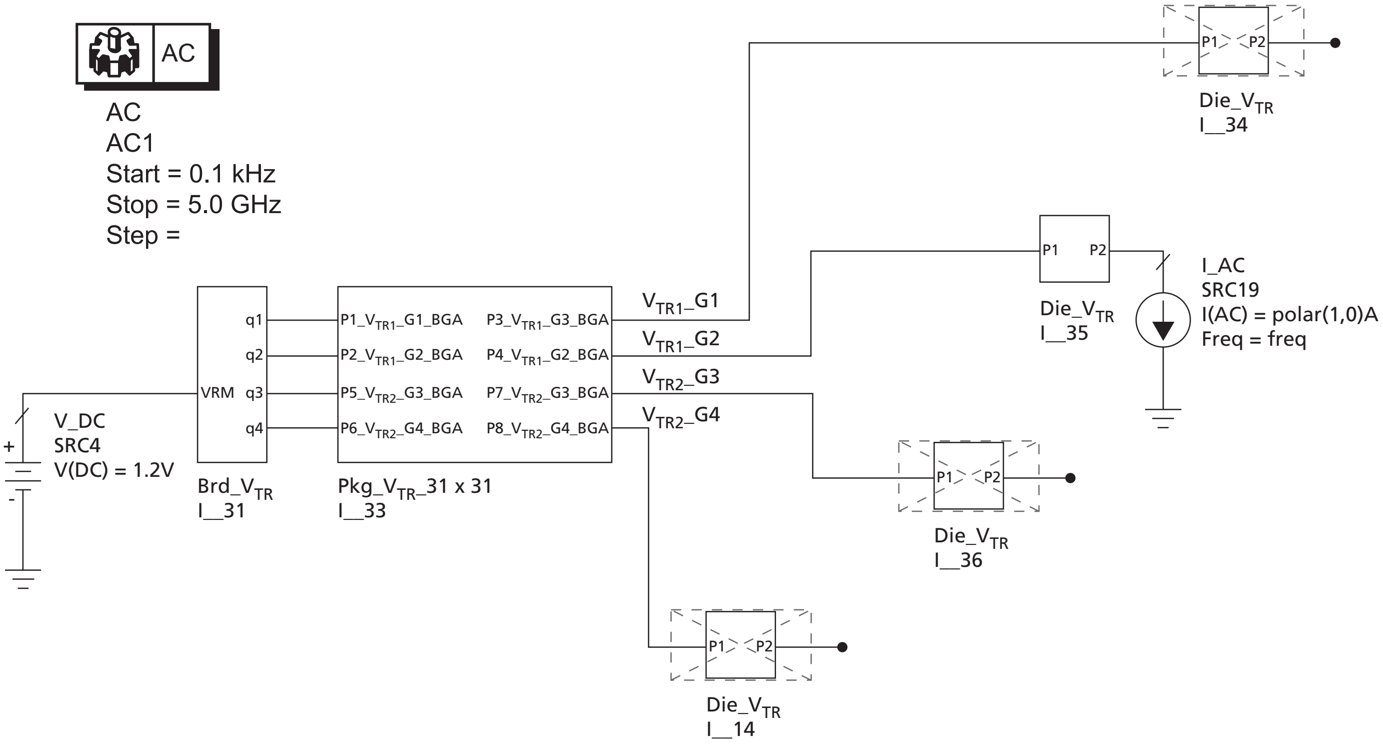
- P6 = AA20, AA22, AB18, AB21, Y17, Y21

For V_{TR}, the package model ports P1, P2, P5, and P6 group the BGA balls as follows:

- P1 = J8, J11, J15, L8, L12
- P2 = J20, J22, L16, L20, L22
- P5 = AB9, AB11, Y9, Y11, Y15
- P6 = AB16, AB20, AB23, Y19, Y23

For V_{TT} and V_{TR}, the simulation will be targeting one specific link while the simulation schematics will include the setup for all links; disable the other links when enabling the target link in the schematics.

Figure 17: V_{TR} Sample Schematics for Simulation Targeting One Link



When extracting board s-parameter models, Micron recommends grouping the package connections into 4 ports the same way the package model was extracted.

3.3.2.3 Results

After the models are ready, run the AC simulation by clicking on the "Gear" button in the schematic window. The workspace will show the simulation results in a pop-up window. The following figures show the simulation results for V_{DD} and V_{DDM}, respectively, with the sample board and the 31 mm x 31 mm package.

Figure 18: V_{DD} PDN Sample Simulation Results

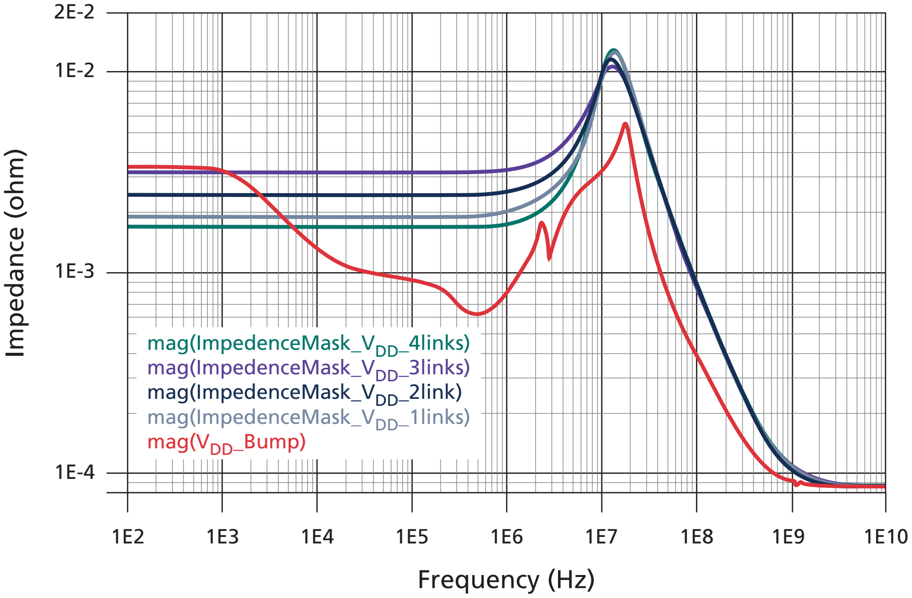
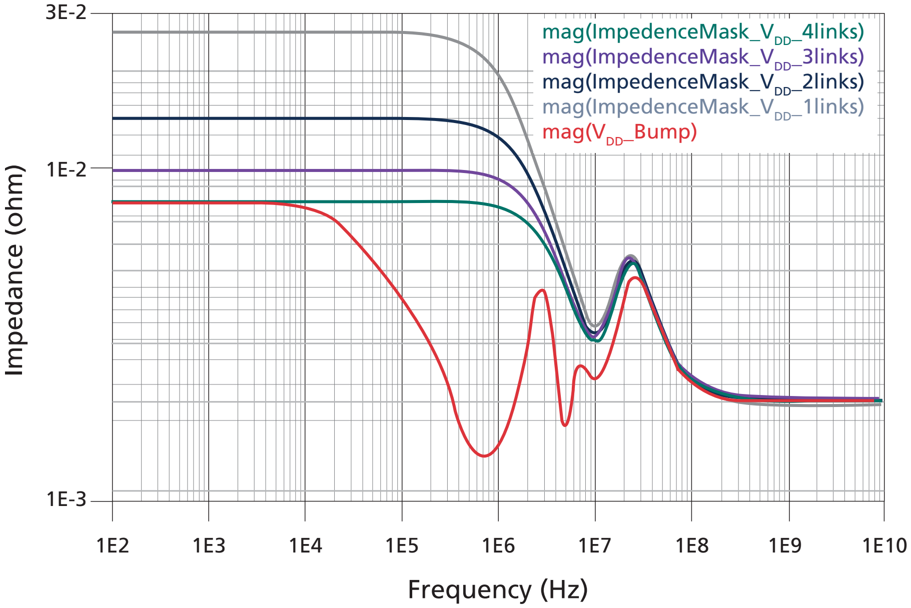


Figure 19: V_{DDM} PDN Sample Simulation Results



Note that each V_{TT}, V_{TR}, V_{DDPLLA} or V_{DDPLLB} simulation targets a specific link. When reviewing the simulation results, it is important to plot the correct impedance profile for the correct link. In the first two figures that follow, the sample simulation shown for V_{TT} is Link 3, and the sample simulation for V_{TR} is Link 1.

Figure 20: V_{TT} PDN Sample Simulation Results

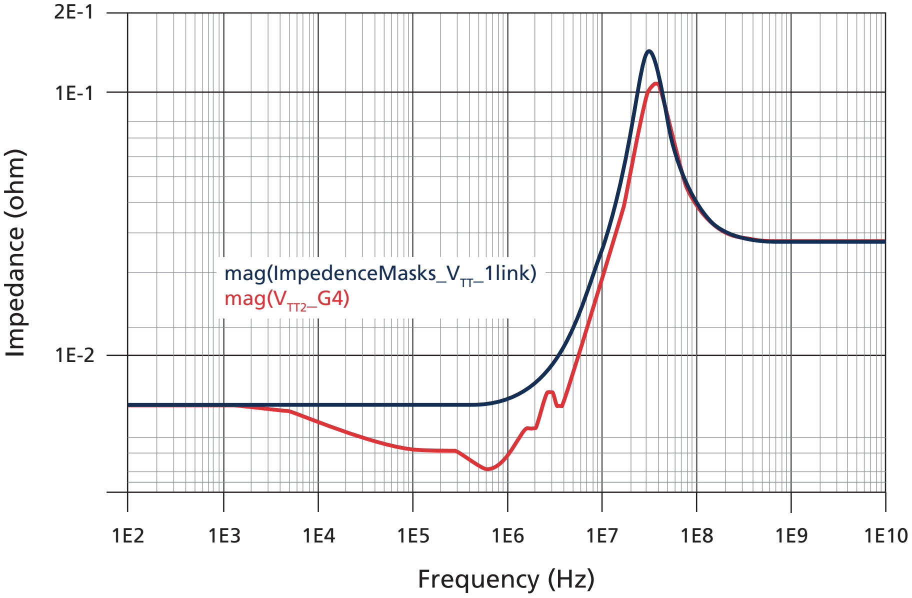


Figure 21: V_{TR} PDN Sample Simulation Results

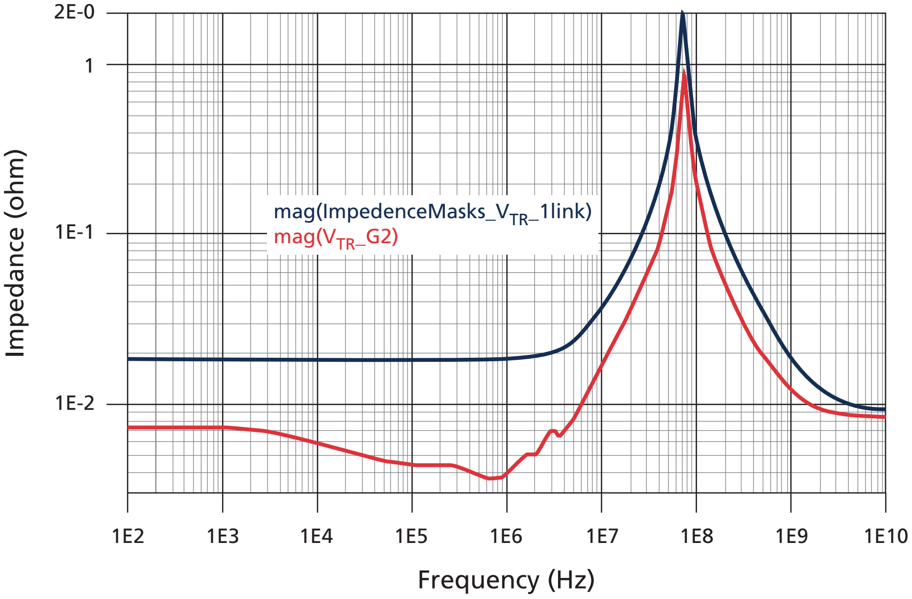


Figure 22: V_{DDPLLA} PDN Sample Simulation Results

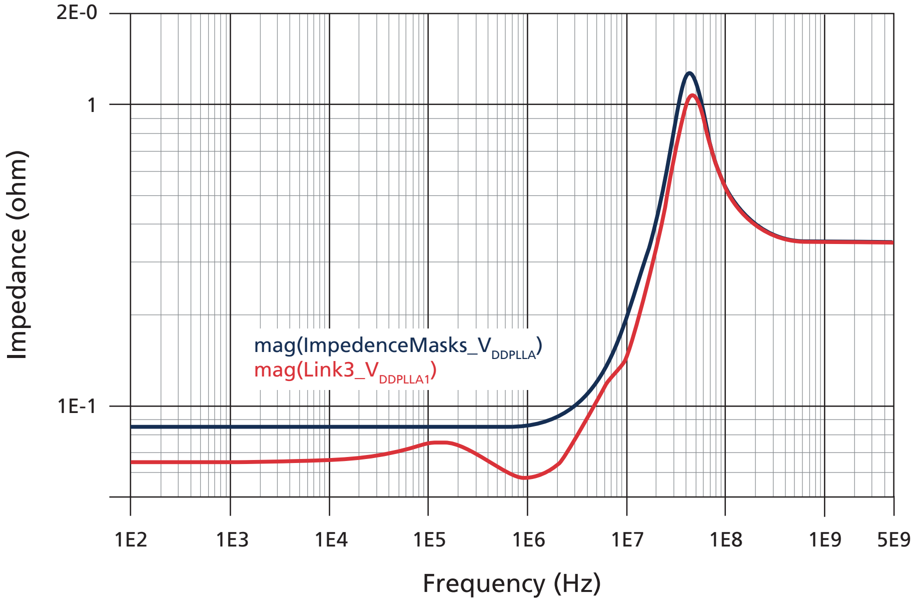
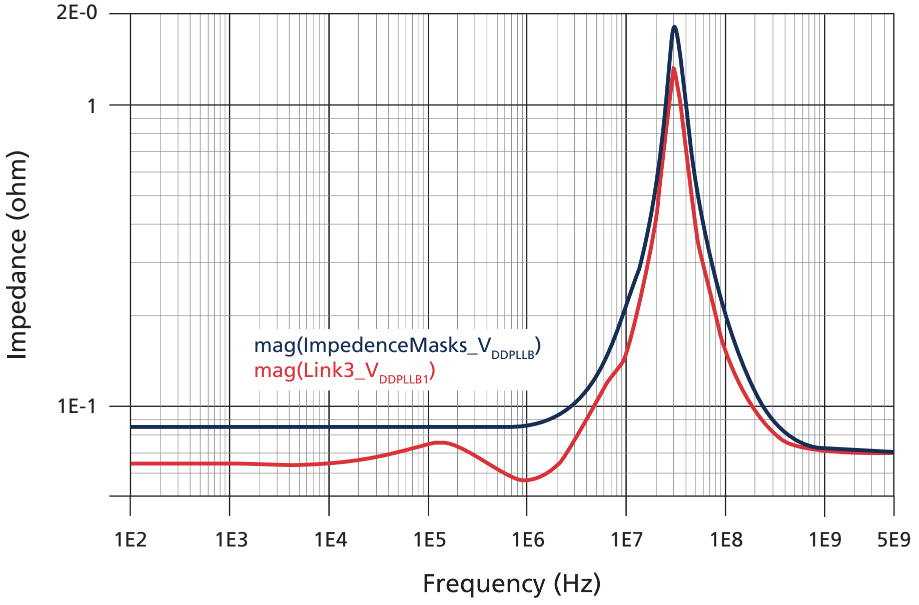


Figure 23: V_{DDPLLB} PDN Sample Simulation Results



If the simulated impedance with the target board stays below the target impedance mask, we can conclude that the on-board decoupling setup is suitable for this power rail; otherwise, the power delivery network may be an issue for the system because of the violation.

For the specific sample simulation of V_{DD} (Figure 18), toward higher frequency, the simulated impedance (red curve) meets the requirement for 1/2/3/4/ active links (see Figure 10). However, at the lower frequency, the impedance will go above the target impedance masks (with violations starting at different frequencies for different numbers of active links). In this situation, a regulator with sense line and feedback that can compensate the low frequency violation (up to the frequency where the violation happens) is required to ensure the voltage to the device will meet the requirement.

3.3.3 V_{DDK} Decoupling

For the 1.5V V_{DDK} voltage rail, the DC current draw is fairly low and transients are small. As a result, the PDN design is greatly simplified and simulation may not be required for a typical design. For the HMC device, Micron recommends using the information in the following table for the board-level decoupling capacitors.

Table 14: V_{DDK} Board Decoupling Capacitor Recommendation

Capacitance (µF)	SMT Package Body	Quantity	Location
22–47	0805 or smaller	3	<10 inches from V_{DDK} pins
10–47	0603 or smaller	3	<1 inches from HMC footprint periphery
1–4.7	0402	5	Within HMC footprint, opposite board side

Note: 1. Ceramic chip capacitors, 2.5V+ voltage rating, and X5R or better temperature coefficient should be used.

3.3.4 PDN Simulation Kit Summary

The HMC PDN simulation kit includes an ADS workspace with all the required PDN components for impedance profile simulation. There is a dedicated schematic for each power rail for a sample simulation setup. An s-parameter model of an HMC sample board, with onboard decoupling capacitors embedded, is included for each sample simulation.

The process of analyzing and simulating the board PDNs can be an iteritive process. In cases where the simulated PDN impedance profile does not meet the simulation target mask, board designers can improve the PDN design through various techniques, such as changing appropriate decoupling capacitors at the correct locations, using full power planes, selecting switching regulators with sense/feedback and high switching frequencies, and so on.

3.4 HMC Gen2 Power Calculator

The HMC Gen2 Power Calculator is used to determine the requirements on the power supplies for the operational current and to develop an accurate thermal model of the HMC in a system. Power supply designs must also meet the specifications in the Gen 2 data sheet.

The power consumption of the HMC is dependant on the junction temperature, which is in turn dependant on the thermal design of the system. The following steps are required for both power calculation and thermal modeling of the HMC:

1. Obtain the HMC Gen2 Power Calculator and the HMC Gen2 FloTHERM models from micron.com.
2. Estimate the Dram 0 junction temperature of the HMC (or use 105°C).
3. Enter the estimated Dram 0 junction temperature and the use case for the specific configuration in the HMC Gen2 Power Calculator.
4. Use the power dissipation values provided by the HMC Gen2 Power Calculator as inputs to the FloTHERM thermal model.
5. Add system parameters to the FloTHERM model (heat sink, airflow, and so forth).
6. Use FloTHERM to run the model and simulate the thermal condition.
7. Use the result of the FloTHERM simulation estimated value of the Dram 0 junction temperature.
8. Repeat from step 2.

This is an iterative process: The junction temperature modifies the power consumption and dissipation, and the power dissipation modifies the junction temperature.

The following example explains the steps for properly using the HMC Gen 2 Power Calculator to generate the power dissipation quantities for use in the FloTHERM model.

3.4.1 HMC Gen2 Power Calculator Outputs

The HMC Gen2 Power Calculator is a spreadsheet that generates output to the DataSheet tab based on the selected parameters. The output of this spreadsheet shows the current for each voltage rail, the total power, and the individual power components for each of the thermal model power sources. The outputs are listed according to three use cases where the user can select the meaning of each use case depending on the inputs. Cases 1, 2 and 3 are provided to allow the user to directly compare various input cases, also labeled Case 1, Case 2 and Case 3. The inputs used by the power calculator are described in more detail later in this document.

The power calculator is useful for thermal modeling and power supply design. Margin should be included in power supply designs above the currents listed in this document, and the designs must meet all parameters listed in the HMC Gen2 data sheet. Power supply designers should take the power consumption of the HMC into consideration when the voltages are marginally high as shown by the power calculator.

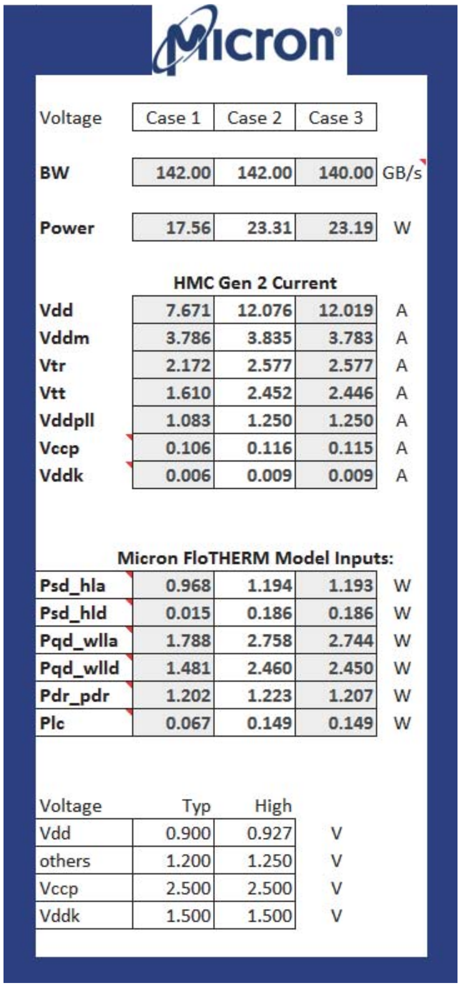
The HMC Gen2 Power Calculator DataSheet tab includes a region dedicated to providing the power dissipation terms for use in the FloTHERM model and a region that provides the RMS values of the individual currents for each of the voltage power rails. The values in this region are updated when the inputs change, including the junction temperature of Dram 0. Dram 0, also called the bottom DRAM, is the DRAM closest to the logic die.

The temperature of Dram 0 (bottom DRAM) is obtainable in the system by using the appropriate ERI command, which are documented in the HMC Gen2 data sheet.

The following figure shows the HMC Gen2 Power Calculator outputs from the Data-Sheet tab.

The example used here is a three link system with each link operating at 15 Gb/s, with a total estimated bandwidth of 142 GB/s. The example is provided only to demonstrate the three different modes of the HMC Gen2 Power Calculator.

Figure 24: HMC Power Calculator Outputs



Voltage	Case 1	Case 2	Case 3	
BW	142.00	142.00	140.00	GB/s
Power	17.56	23.31	23.19	W
	HMC Gen 2 Current			
Vdd	7.671	12.076	12.019	A
Vddm	3.786	3.835	3.783	A
Vtr	2.172	2.577	2.577	A
Vtt	1.610	2.452	2.446	A
Vddpll	1.083	1.250	1.250	A
Vccp	0.106	0.116	0.115	A
Vddk	0.006	0.009	0.009	A

Micron FloTHERM Model Inputs:

Psd_hla	0.968	1.194	1.193	W
Psd_hld	0.015	0.186	0.186	W
Pqd_wlla	1.788	2.758	2.744	W
Pqd_wlld	1.481	2.460	2.450	W
Pdr_pdr	1.202	1.223	1.207	W
Plc	0.067	0.149	0.149	W

Voltage	Typ	High	
Vdd	0.900	0.927	V
others	1.200	1.250	V
Vccp	2.500	2.500	V
Vddk	1.500	1.500	V

The inputs used by the power calculator depend on which of three calculator modes is used: Basic, Enhanced or Advanced. Basic is the simplest to use and Advanced is the most complex. All three modes may be used as the design progresses and additional details become available. Each mode uses various parameters to describe each of the use cases. Descriptions of each parameter are included in the spreadsheet as popup comments.

Up to 40% of the power of the HMC is related to the bandwidth associated with the use case. It is important, therefore, to enter the correct quantity of bandwidth for each use case. It is best to use the actual bandwidth as achieved in a real system; however, early in the design the bandwidth as determined by a system simulation may be used. Very early on in the design process using the bandwidth limits implemented in the HMC Gen2 Power Calculator as an indicator of the maximum power point is acceptable.

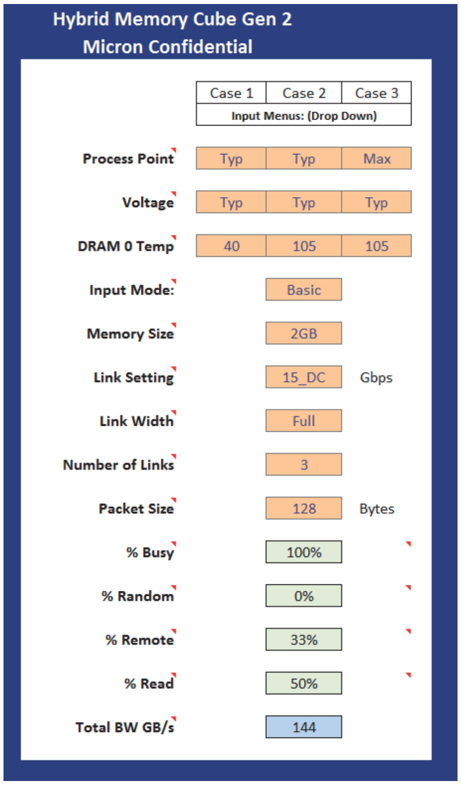
Bandwidth estimates are intended for the power calculator only and may not reflect actual system results.

3.4.2 HMC Gen2 Power Calculator Basic Input Mode

Basic is the simplest calculator mode to use and is selected from the Input Mode pull down menu. In this mode, the user selects the following three parameters individually for each case: Process Point, Voltage and Dram 0 Temp. The user also selects the following four parameters common to all three cases: Link Speed, Link Width, Number of Links and Packet Size. Based on these parameters the power calculator generates an estimate of bandwidth and the resulting current and power values.

In Basic mode the maximum theoretical bandwidth with 50% Write, 50% Read (Non-posted writes and random addressing is not used). The value of % Remote is adjusted based on the number of links selected and the model using the entire HMC memory equally on all active links.

Figure 25: HMC Gen2 Power Calculator in Basic Input Mode



Hybrid Memory Cube Gen 2
Micron Confidential

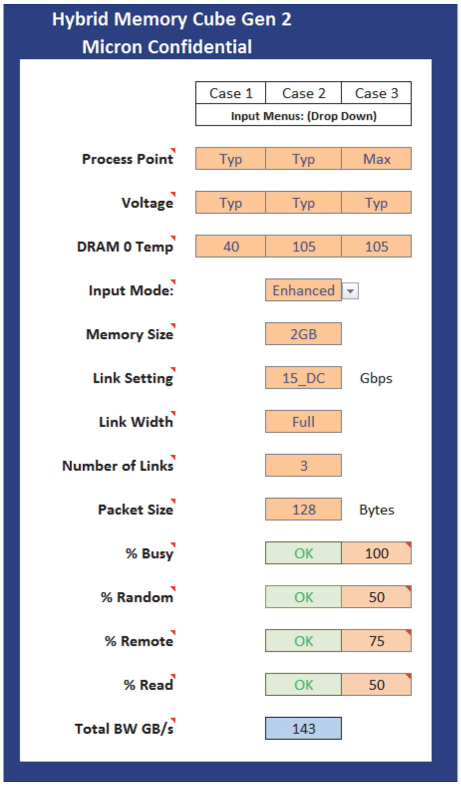
	Case 1	Case 2	Case 3
	Input Menus: (Drop Down)		
Process Point	Typ	Typ	Max
Voltage	Typ	Typ	Typ
DRAM 0 Temp	40	105	105
Input Mode:		Basic	
Memory Size		2GB	
Link Setting		15_DC	Gbps
Link Width		Full	
Number of Links		3	
Packet Size		128	Bytes
% Busy		100%	
% Random		0%	
% Remote		33%	
% Read		50%	
Total BW GB/s		144	

3.4.3 HMC Gen2 Power Calculator Enhanced Mode

The Enhanced calculator mode builds on the features of the Basic mode by adding the following parameters: % Random, % Remote and % Read. Each of these parameters is described in more detail in pop-up notes in the spreadsheet.

In Enhanced mode the % Busy parameter is used to set the bandwidth to the system simulation or system measurements. This parameter is not the same as the link utilization, nor is it the same as the memory bandwidth utilization, rather it is a combination of both of these and the read-to-write ratio. The % Remote and % Random parameters will limit the achievable bandwidth in a real system, but the impact of % Remote and % Random is not calculated in Enhanced mode. The impact of posted writes is not considered in this mode.

Figure 26: HMC Gen2 Power Calculator in Enhanced Mode



Hybrid Memory Cube Gen 2
Micron Confidential

	Case 1	Case 2	Case 3
	Input Menus: (Drop Down)		
Process Point	Typ	Typ	Max
Voltage	Typ	Typ	Typ
DRAM 0 Temp	40	105	105
Input Mode:		Enhanced	
Memory Size		2GB	
Link Setting		15_DC	Gbps
Link Width		Full	
Number of Links		3	
Packet Size		128	Bytes
% Busy		OK	100
% Random		OK	50
% Remote		OK	75
% Read		OK	50
Total BW GB/s		143	

3.4.4 HMC Gen2 Power Calculator Advanced Mode

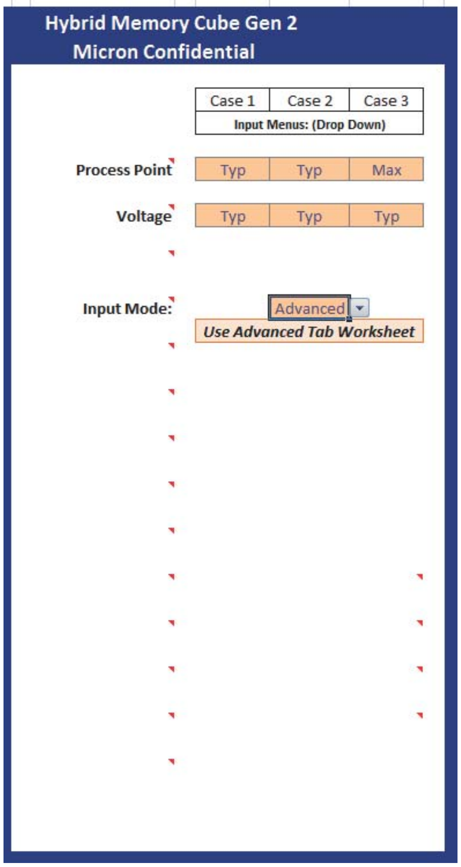
The Advanced calculator mode replaces many parameters with those entered on the separate Advanced worksheet. The Process Point and Voltage parameters remain on the DataSheet tab, but the other parameters are moved to the Advanced tab worksheet. This mode allows the combination of all the different transaction types and the comparison between different link configurations. The most thorough use of this mode requires an understanding of each transaction size and the bandwidth associated with each transaction size.

In Advanced mode, the bandwidth of each transaction type must be determined, and for each transaction type that has posted writes, random addressing, and remote access, the ratio of the bandwidth of that transaction type needs to be entered into the table. In this mode a more precise value of temperature may be entered for each use case.

The Advanced mode enables the user to enter bandwidth values larger than those that are actually achievable. This mode will scale those bandwidths down to values only slightly larger than what is achievable. During this process the green "OK" indicator is replaced with a yellow "Over" indicator. The Over indicator is not an error, it is meant to inform the user that the bandwidth has been reduced. When the Over flag displays, the reduced bandwidths are shown in blue. It is possible to use this as a percentage mode by ensuring that the sum of all write and atomic transactions add up to 100 and that the sum of all read transactions add up to 100. The resulting 200 GB/s cause the Advanced mode to enter the Over mode and scale the bandwidth according to the memory size, link speed, link width and number of links selected for that use case.

Due to rounding errors, the Over flag may appear when the difference in bandwidths is minimal. This is shown in Case 3. Again, this condition is not an error.

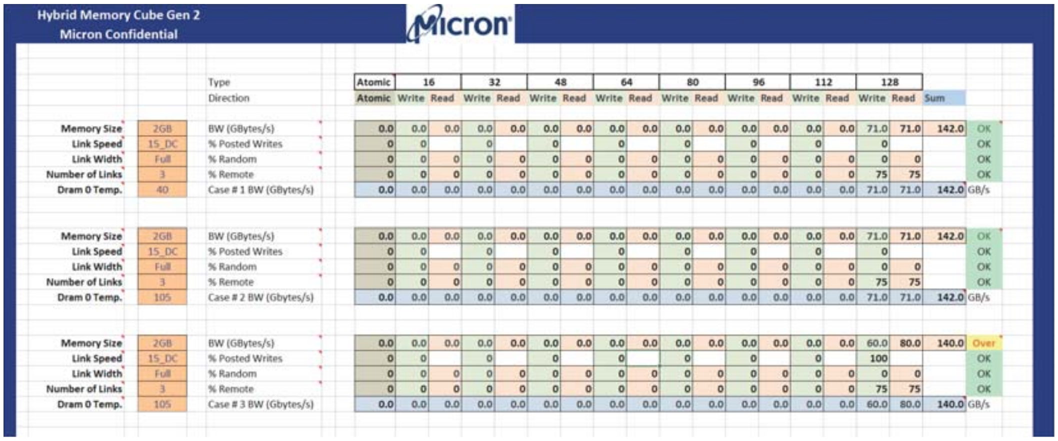
Figure 27: HMC Gen2 Power Calculator in Advanced Mode: DataSheet Tab



Hybrid Memory Cube Gen 2
Micron Confidential

	Case 1	Case 2	Case 3
	Input Menus: (Drop Down)		
Process Point	Typ	Typ	Max
Voltage	Typ	Typ	Typ
Input Mode:	Advanced		
	Use Advanced Tab Worksheet		

Figure 28: HMC Gen2 Power Calculator in Advanced Mode – Advanced Tab



Hybrid Memory Cube Gen 2
Micron Confidential

		Type	Atomic	16		32		48		64		80		96		112		128			
		Direction	Atomic	Write	Read	Write	Read	Write	Read	Write	Read	Write	Read	Write	Read	Write	Read	Write	Read	Sum	
Memory Size	2GB	BW (GBytes/s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	71.0	71.0	142.0	OK
Link Speed	15_DC	% Posted Writes	0	0		0		0		0		0		0		0		0			OK
Link Width	Full	% Random	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		OK
Number of Links	3	% Remote	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	75		OK
Dram 0 Temp.	40	Case # 1 BW (GBytes/s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	71.0	71.0	142.0	GB/s
Memory Size	2GB	BW (GBytes/s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	71.0	71.0	142.0	OK
Link Speed	15_DC	% Posted Writes	0	0		0		0		0		0		0		0		0			OK
Link Width	Full	% Random	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		OK
Number of Links	3	% Remote	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	75		OK
Dram 0 Temp.	105	Case # 2 BW (Gbytes/s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	71.0	71.0	142.0	GB/s
Memory Size	2GB	BW (GBytes/s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	60.0	80.0	140.0	Over
Link Speed	15_DC	% Posted Writes	0	0		0		0		0		0		0		0		100			OK
Link Width	Full	% Random	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		OK
Number of Links	3	% Remote	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	75		OK
Dram 0 Temp.	105	Case # 3 BW (Gbytes/s)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	60.0	80.0	140.0	GB/s

3.4.5 Chaining and Various Link Configuration Use Cases

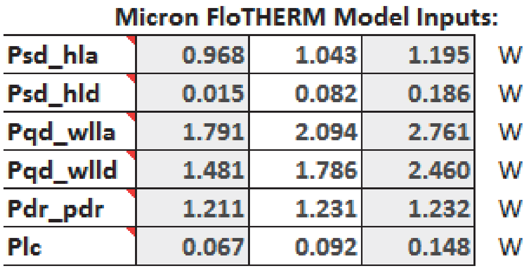
For each HMC that has downstream HMC devices, the power calculator must be used twice with different quantities entered for bandwidth. For the values of I_{DD} (current of V_{DD}) and Plc (power of the logic core), the bandwidth value for that HMC and the bandwidth values for all downstream HMC devices must be added together. For all other values, use the bandwidth values for the initial HMC only. In both cases the number of links includes all the active links, including all pass through links. In chaining configurations where two hosts are addressing the same HMC, ensure that the bandwidth used includes the bandwidth from both hosts.

For HMC configurations not directly supported by the power calculator, additional steps are necessary to determine the correct currents and power values. For HMC applications where multiple link speeds are used, use the power calculator for each link speed and then combine the values of I_{TR} (current of V_{TR}), I_{TT} (current of V_{TT}). The values used for Psd_hla and Psd_hli in the FloTHERM model must be associated with the link speed of each link. In any design where both full-width and half-width links are used, the power calculator must be used twice, once for full-width and once for half-width (a half-width link uses slightly more than half the power of a full-width link).

3.4.6 HMC Gen2 Power Calculator Outputs as FloTHERM Inputs

The power calculator provides the values for each parameter in the Micron HMC thermal model, as shown in the following figure.

Figure 29: HMC Gen2 Thermal Model Power Sources



Micron FloTHERM Model Inputs:

Psd_hla	0.968	1.043	1.195	W
Psd_hld	0.015	0.082	0.186	W
Pqd_wlla	1.791	2.094	2.761	W
Pqd_wlld	1.481	1.786	2.460	W
Pdr_pdr	1.211	1.231	1.232	W
Plc	0.067	0.092	0.148	W

List of Terms

- **Psd_hla**: Power of the SerDes for a half link configured in Active mode.
- **Psd_hld** : Power of the SerDes for a half link configured in Down mode.
  - In this section down mode represents both sleep mode and power-down mode. Each SerDes is constructed as two half links so that when a half link configuration is selected, the HMC is able to reduce the power associated with the unused half of the link. When the HMC is configured to use less than four full-width links, each half of the unused full-width link becomes a down mode half link for the purpose of thermal analysis. When the HMC is configured to use half-width links, then the half width that is used is the 0 to 7 half.
- **Pqd_wlla**: Power of the QuaDrant of the HMC that is associated with a local link configured in Active mode.
- **Pqd_wlld**: Power of the QuaDrant of the HMC that is associated with a local link configured in Down mode.

  - These two terms are associated with the four quadrants of the HMC logic die. Each quadrant has a connection to the Serdes link that is closest to it on the die and has a component of power that is associated with the use of that link.
- **Pdr_pdr**: Power per individual DRAM.
  - This is the power that each DRAM die in the HMC will be dissipating. The 2 GB HMC uses four DRAM die.
- **Plc**: Power of the logic core.
  - Plc represents the power dissipated on the logic die that is not associated with the other regions on the logic die.

Each of the above terms is calculated to match the individual regions as specified in the FloTHERM model. These power values may be multiplied by the appropriate number of DRAM die and the configured number of links to verify that the resulting sum matches the indicated total power for that use case.

3.5 Thermal Modeling

3.5.1 Introduction

Ensuring device specifications are met under all expected use conditions is crucial in modern electronics design. For systems using the HMC, junction temperature of the die inside the HMC as it is used in the system is a critical specification.

This section provides information needed to develop thermal system solutions that meet the junction temperature specifications for both the DRAM and the logic layers.

Junction temperatures are specified at the temperature sensor locations only. Other areas of the device may experience higher temperatures than the temperature sensor locations, but temperatures at locations other than the temperature sensors are accommodated in the design of the HMC.

- Logic layer: $0°C \leq T_J \leq 110°C$
- DRAM layers: $0°C \leq T_J \leq 105°C$

Micron provides a simulation model designed for use with the Mentor Graphics FloTHERM® thermal modeling software package.

- 31 mm x 31 mm package/2 GB HMC (4-high): HMC_Gen2_31x31_4H_v*.*.pdml

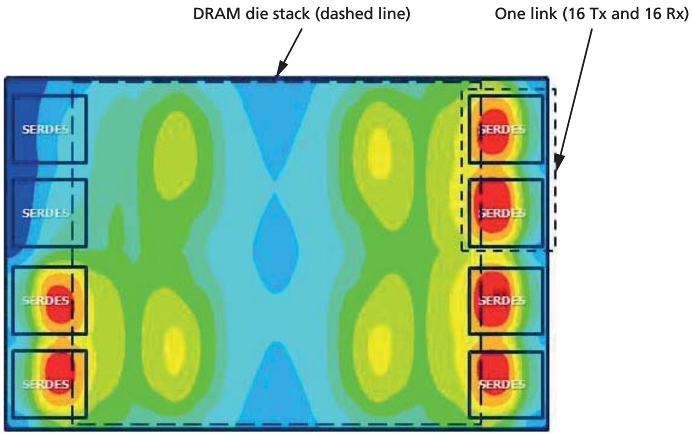
The most recent version of the package model is available on micron.com in a zip file with a name similar to: HMC_Gen2_FloTHERM_models_2016-5-24.zip.

3.5.2 HMC Gen2 Thermal Description

The HMC device is comprised of a high-speed logic die that sits below four DRAM die, which are connected using through-silicon vias (TSVs).

The floor plan of the HMC logic layer was designed such that the high power-consumption areas, specifically the high-speed SerDes blocks, were positioned at the periphery of the device. This provides a thermal path from the SerDes to the lid that does not pass through the DRAM stack above the logic die. Figure 1 shows the physical relationship between the DRAM and the high-speed SerDes blocks and the typical temperature profile. In this example, all links except the one located in the upper left corner are enabled.

Figure 30: Logic Die Temperature Distribution Example



3.5.3 Thermal Modeling

Using FloTHERM, thermal modeling computes the three-dimensional temperature profile taking into account the various thermal paths and thermal sources. The computed temperatures can be compared to the HMC-specified junction temperatures. Before using the HMC FloTHERM model it is necessary to determine the power dissipation of the various components of the HMC as it is configured and used in the application.

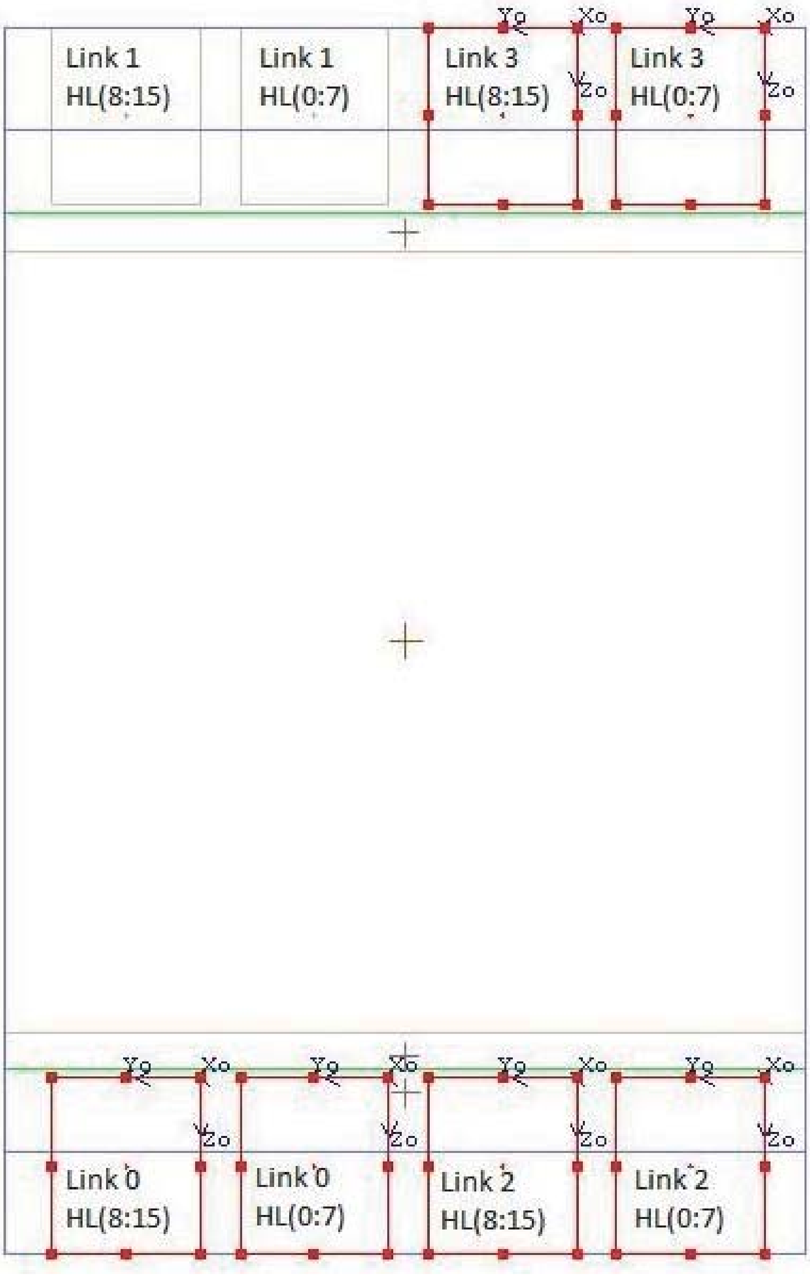
This section provides an example of where the various power dissipation parameter values are entered into the FloTHERM model. A three-link, full-width configuration is used to demonstrate how power is assigned to active and inactive half links. Each of the following FloTHERM menus show where the power value is entered.

3.5.3.1 Link and Half-Link Location Identification

The thermal model uses eight regions to assign the power for each half link. For designs using full-width links, power is assigned to both half links as active half links. For designs using active half link, power is assigned to half link (0–7). The **Psd_hla** parameter is used as the source for assigning power to active half links; the **Psd_hld** parameter is used as the source for assigning power to inactive half links.

The logic power subblock labeled SerDes00 represents Link 0 bits 0:7, and SerDes01 represents Link 0 bits 8:15.

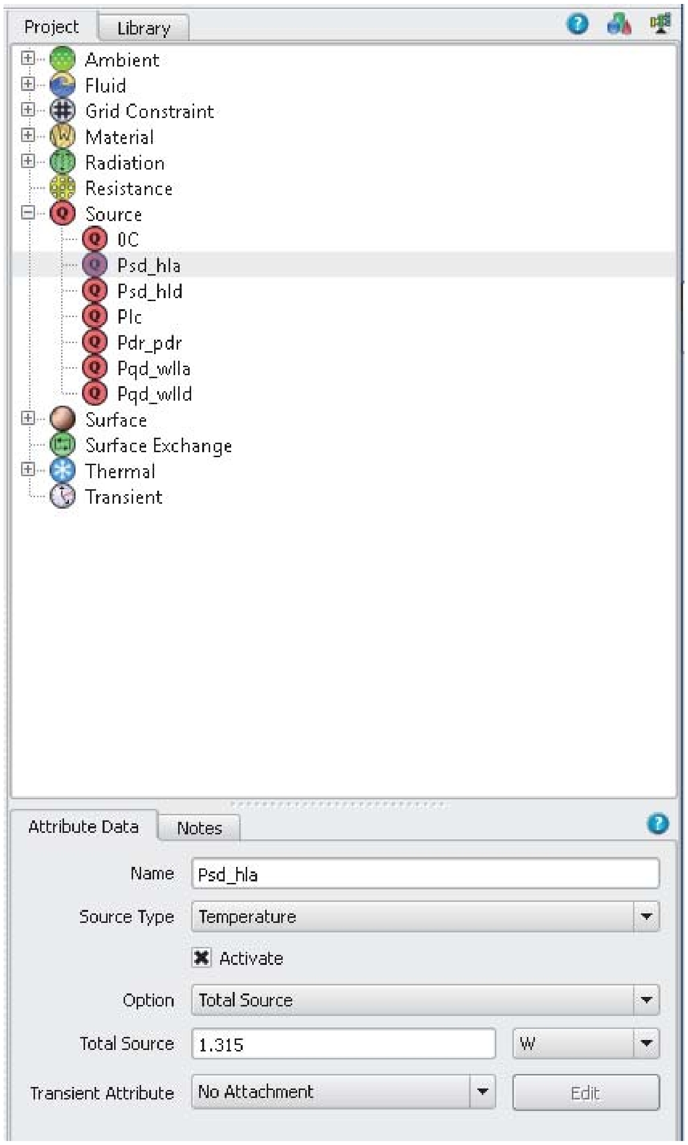
Figure 31: Link and Half-Link Identification



3.5.3.2 Assigning Values to Power Sources

Enter the value for each parameter provided by the power calculator into the appropriate power source as indicated below.

Figure 32: Power Source Parameters



Name: Psd_hla

Source Type: Temperature

☒ Activate

Option: Total Source

Total Source: 1.315 W

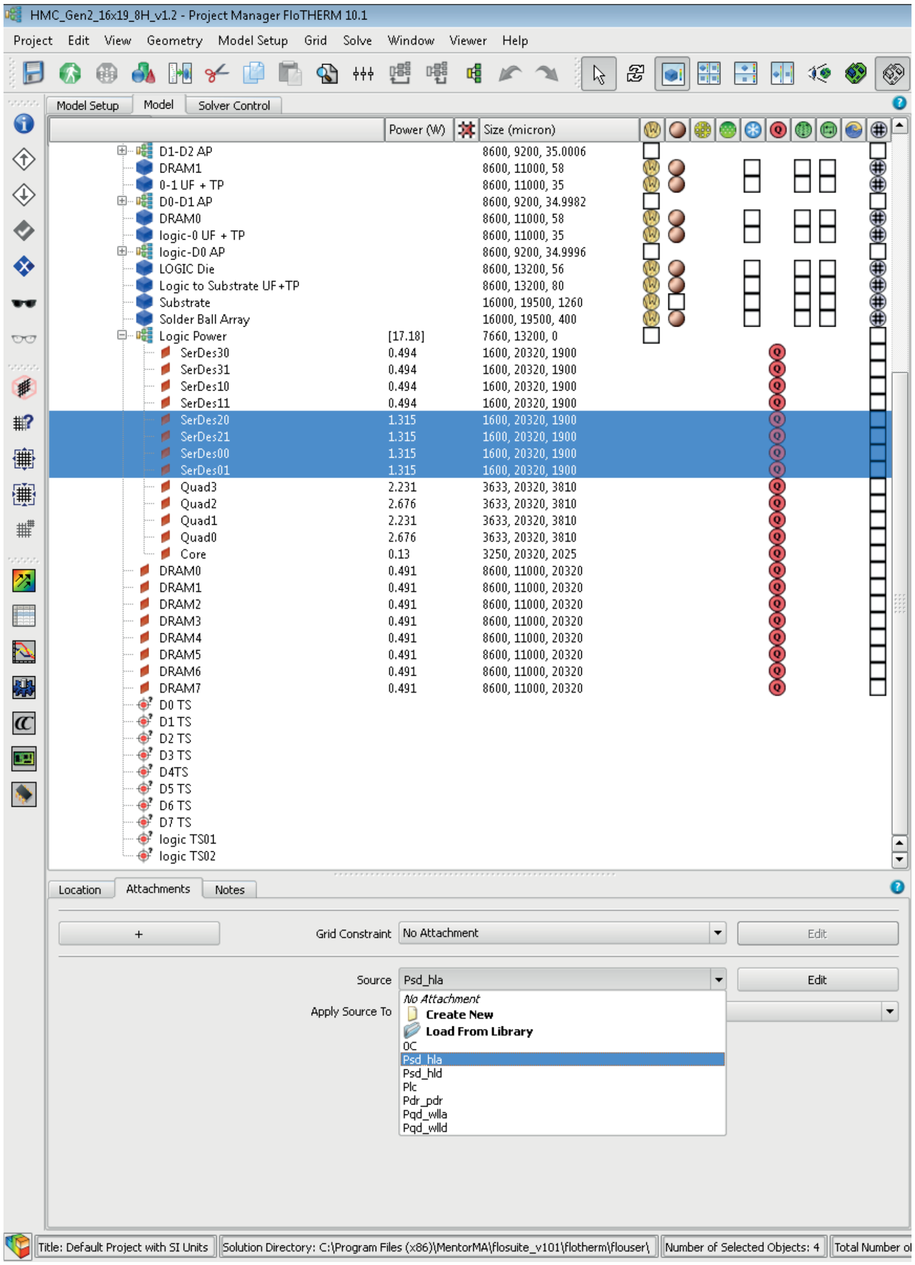
Transient Attribute: No Attachment

3.5.3.3 Assign Sources to Physical Regions

Select the appropriate source for each region in the HMC, taking into consideration link widths and which links are active. The links are labeled **SerDes(0..3)(x)** for the four full links and **SerDes(x)(0..1)** each half link. IE: The two half links which make up the full link for Link 3 are **SerDes30** and **SerDes31**. The quadrants adjacent to active half links use power source **Pqd_wlla**; the other quadrants use **Pqd_whld.** The power source **Plc** occurs once in the center of the logic die. Each DRAM die in the stack uses the power source **Pdr_prd**.

The following example shows the selection of the four half links of links 0 and 2 and their assignment to the power source **Psd_hla**, which has a power of 1.068 Watts per half link. The correct power source must be assigned to each of the thirteen regions on the logic die and the one region of each DRAM die.

Figure 33: Select Physical Regions



	Power (W)	Size (micron)
D1-D2 AP		8600, 9200, 35.0006
DRAM1		8600, 11000, 58
0-1 UF + TP		8600, 11000, 35
D0-D1 AP		8600, 9200, 34.9982
DRAM0		8600, 11000, 58
logic-0 UF + TP		8600, 11000, 35
logic-D0 AP		8600, 9200, 34.9996
LOGIC Die		8600, 13200, 56
Logic to Substrate UF+TP		8600, 13200, 80
Substrate		16000, 19500, 1260
Solder Ball Array		16000, 19500, 400
Logic Power	[17.18]	7660, 13200, 0
SerDes30	0.494	1600, 20320, 1900
SerDes31	0.494	1600, 20320, 1900
SerDes10	0.494	1600, 20320, 1900
SerDes11	0.494	1600, 20320, 1900
SerDes20	1.315	1600, 20320, 1900
SerDes21	1.315	1600, 20320, 1900
SerDes00	1.315	1600, 20320, 1900
SerDes01	1.315	1600, 20320, 1900
Quad3	2.231	3633, 20320, 3810
Quad2	2.676	3633, 20320, 3810
Quad1	2.231	3633, 20320, 3810
Quad0	2.676	3633, 20320, 3810
Core	0.13	3250, 20320, 2025
DRAM0	0.491	8600, 11000, 20320
DRAM1	0.491	8600, 11000, 20320
DRAM2	0.491	8600, 11000, 20320
DRAM3	0.491	8600, 11000, 20320
DRAM4	0.491	8600, 11000, 20320
DRAM5	0.491	8600, 11000, 20320
DRAM6	0.491	8600, 11000, 20320
DRAM7	0.491	8600, 11000, 20320
D0 TS		
D1 TS		
D2 TS		
D3 TS		
D4TS		
D5 TS		
D6 TS		
D7 TS		
logic TS01		
logic TS02		

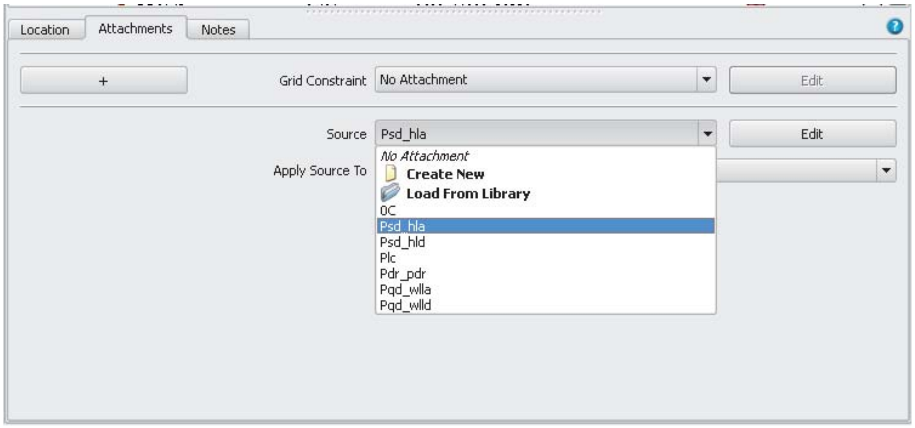
Grid Constraint: No Attachment

Source: Psd_hla

Apply Source To

Title: Default Project with SI Units | Solution Directory: C:\Program Files (x86)\MentorMA\flosuite_v101\flotherm\flouser\ | Number of Selected Objects: 4 | Total Number o

Figure 34: Select Appropriate Power Source for Physical Regions



Grid Constraint: No Attachment

Source: Psd_hla

Apply Source To

Figure 35: Verify Selected Physical Regions

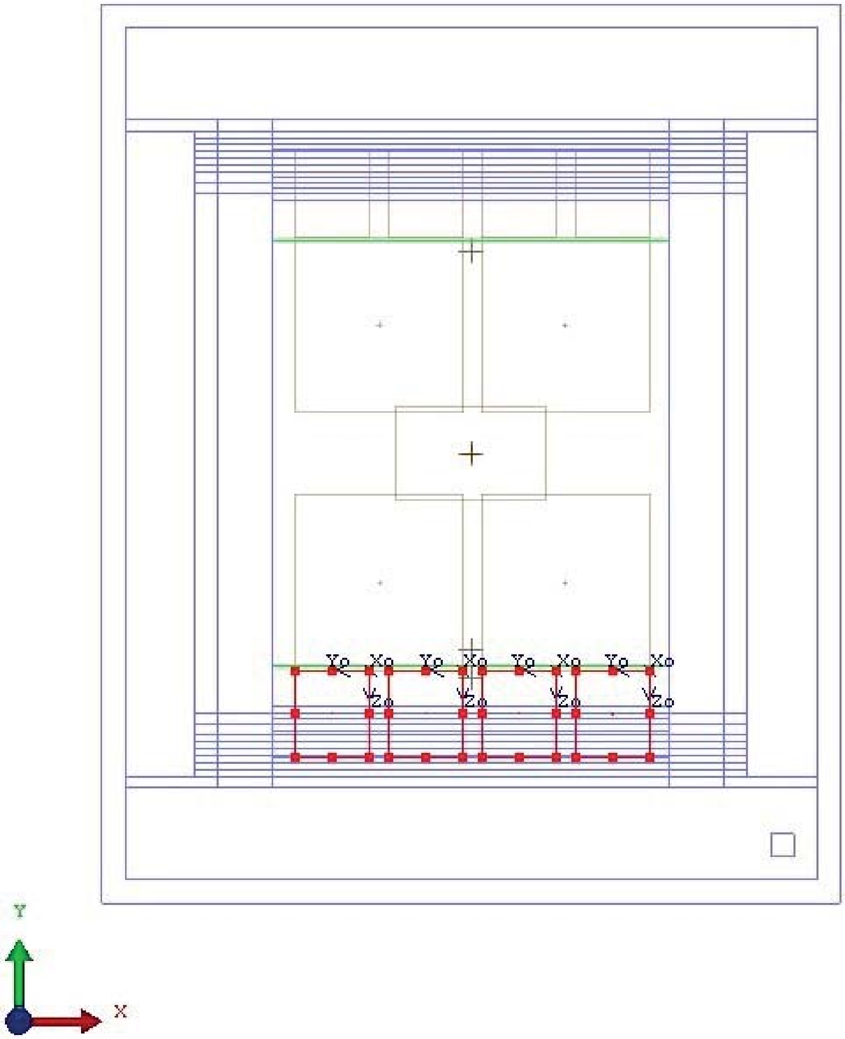
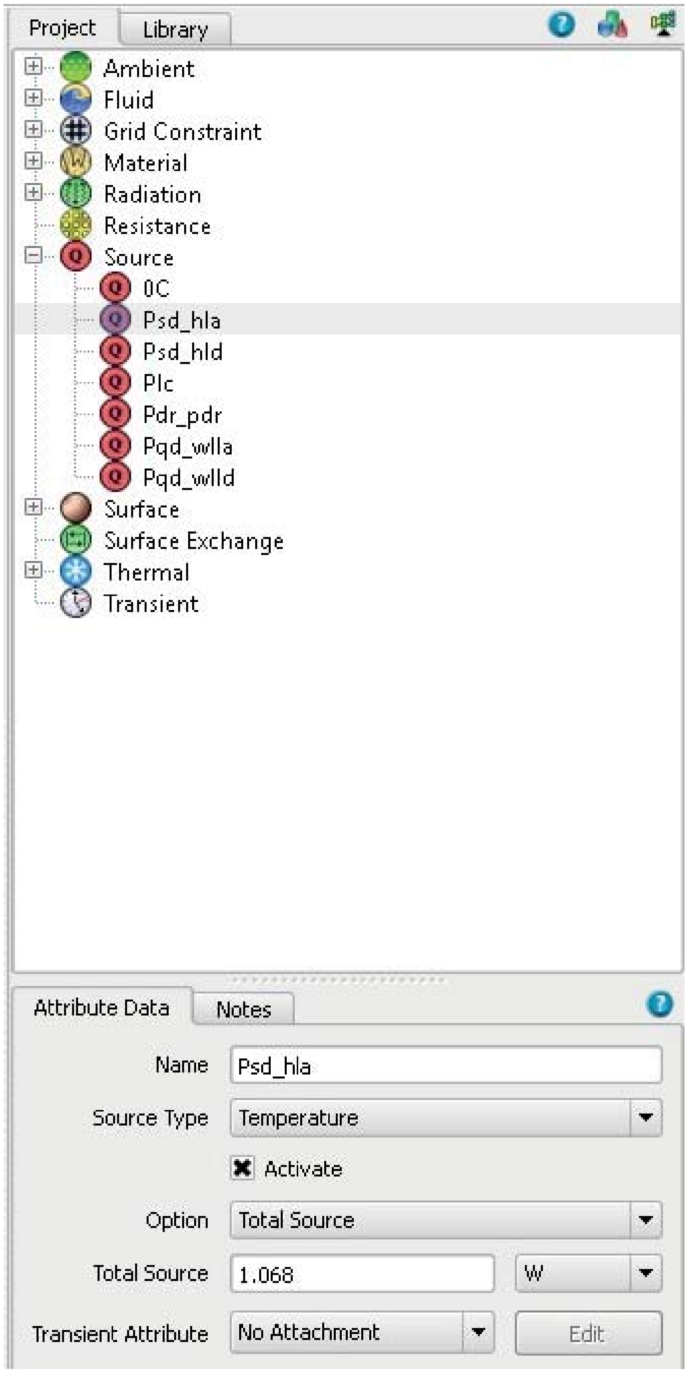


Figure 36: Verify Power Assignments to Model Power Sources



Name: Psd_hla

Source Type: Temperature

☒ Activate

Option: Total Source

Total Source: 1.068 W

Transient Attribute: No Attachment

After the power values have been entered into the sources and the sources applied to the correct regions, the thermal model is ready to simulate the HMC thermal conditions for the desired use case.

3.6 Board Routing Guidelines

The HMC is a revolutionary breakthrough in memory technology. It provides extreme memory bandwidth using an abstracted protocol that is independent of any memory timings and is accessed through high-speed SerDes interfaces. Not only is the performance unmatched by any other solution, but the design and routing requirements are much simpler than traditional DDR-type interfaces.

For example, a DDR3-based PC3-12800 RDIMM requires 146 active signals per channel for a peak bandwidth of 12.8 GB/s, yet a single HMC Gen2 device link with only 67 active signals can provide up to 60 GB/s of bandwidth.* If all four links are wired, the HMC (Gen2) device can provide up to 160 GB/s of total memory bandwidth, as shown in the figure on the next page.

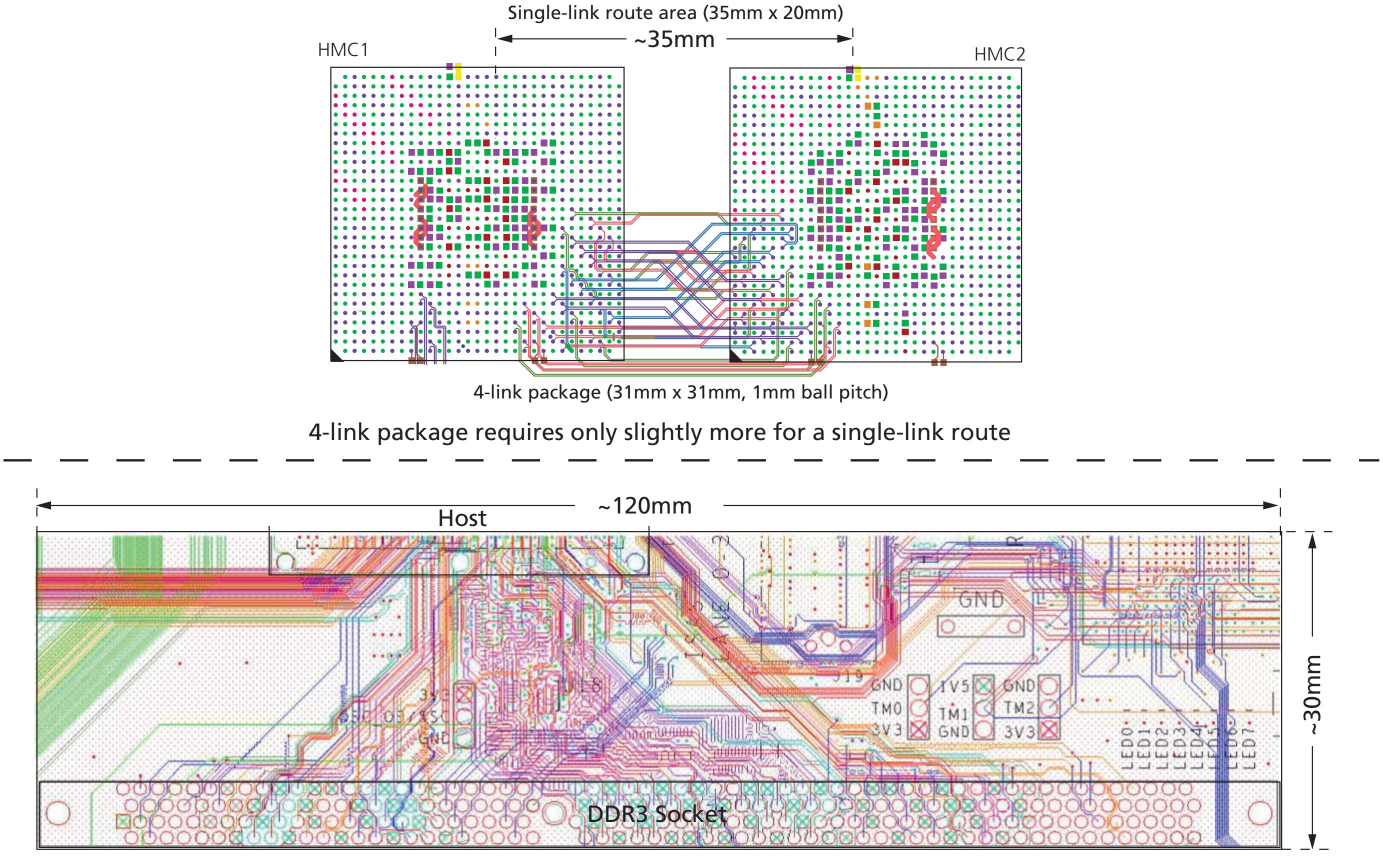
*In this document, 1 GB of capacity equals 1,073,741,824 bytes (2^{30}), and 1 GT/s equals 1,000,000,000 bytes transferred per second (10^{9}).

This section provides sound methods, proven solutions, and detailed PCB layout guidelines to enable successful designs using the HMC Gen2 device. It discusses how to effectively lay out and route signals between multiple HMC devices and what to expect when interfacing to a host controller with an ideal ballout.

Examples of PCB routes, power delivery requirements, route limitations, and signal evaluation for key corners are included. A primary focus is using cost-effective substrates with minimal layers to achieve robust signal integrity.

3.6.1 HMC vs. DDR3 Traditional Routing

Figure 37: HMC-to-HMC Single Channel vs. Traditional Channel Route



4-link package requires only slightly more for a single-link route

DDR3 Channel: A single-DIMM channel requires about 3600mm to route

3.6.2 Understanding HMC Function and Features

This section provides insight to basic HMC Gen2 device functionality; however, Micron advises becoming familiar with the device data sheet prior to starting the board-level design. This device reflects a revolutionary change in memory technology. HMC architecture makes designs simpler, more robust, and transparent to other memory types that could be within the cube, whether it be DRAM or flash, volatile or nonvolatile.

3.6.2.1 Interface and Signals

The HMC Gen2 device product uses high-speed SerDes interfaces to transfer commands and data across the link. Each link consists of up to 16 duplex lanes in both transmit and receive directions for a total of 32 differential pairs. The link can be configured for the standard full-width, half-width, or asymmetrical structure to optimize for a majority of either READ or WRITE operations. The lanes can be set to run at 10 Gb/s, 12.5 Gb/s, or 15 Gb/s, as required to meet system performance targets. The physical interface is based on robust transceivers that can tolerate up to 12dB of channel loss at the Nyquist frequency of 7.5 GHz. The physical interface can either be DC or AC coupled.

All four links in the HMC Gen2 device use the same differential reference clock, which runs at 125 MHz, 156.25 MHz, or 166.67 MHz. Slower speed signals include a fatal error output signal (FERR_N), a reset (P_RST_N), power-down input/return for each link (LxRXPS LxTXPS), bootstrap inputs for device configuration, a clock reference select (REFCLKSEL), JTAG TAP (TMS, TCK, TDI, TDO, TRST_N), I ^{2}C bus (SCL, SDA), and a precision resistor (0.1%) for impedance calibration (EXTRESxx). In addition, the device supports two slower sideband maintenance channels. Typically, users will choose either I^{2}C or JTAG.

3.6.2.2 Package Options

Micron currently offers a single standard package option for the HMC Gen2 device:

- Standard 896-ball (1mm pitch), 31mm x 31mm package with four independent links for a total link bandwidth of up to 240 GB/s

The standard 896-ball package is organized with a link in each quadrant, and the primary power is sourced through the center ball array. The standard package type has a copper plate incorporated for improved thermal conductivity and has a total package thickness of less than 4mm.

> **Note:** Depending on the operating conditions and environment, the layout should reserve additional keep-out area around the device in case mechanical mounting for a heat spreader needs to be accommodated.

Figure 38: Standard (4-Link) Package Ballout

Link 2

Link 0

Link 3

Link 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A		VSS	L2TXP[4]	L2TXN[4]	VSS	VSS	L2RXP[8]	L2RXN[8]	VSS	VSS	L2TXP[13]	L2TXN[13]	VSS	VSS	L0RXN[2]	L0RXP[2]	VSS	VSS	L0TXN[6]	L0TXP[6]	VSS	VSS	L0RXN[3]	L0RXP[3]	VSS	VSS	L0TXN[15]	L0TXP[15]	VSS	
B	VSS	VSS	VSS	L2TXP[14]	L2TXN[14]	VSS	VSS	L2RXP[12]	L2RXN[12]	VSS	VSS	L2TXP[12]	L2TXN[12]	VSS	VSS	VSS	VSS	L0TXN[7]	L0TXP[7]	VSS	VSS	L0RXN[7]	L0RXP[7]	VSS	VSS	L0TXN[5]	L0TXP[5]	VSS	VSS	VSS
C	L2RXP[1]	L2RXN[1]	VSS	VSS	L2TXP[0]	L2TXN[0]	VSS	VSS	L2RXP[13]	L2RXN[13]	VSS	DNU	L2TXP[8]	L2TXN[8]	VSS	VSS	L0TXN[3]	L0TXP[3]	VSS	VSS	L0RXN[6]	L0RXP[6]	VSS	DNU	L0TXN[11]	L0TXP[11]	VSS	VSS	L0RXN[10]	L0RXP[10]
D	VSS	L2RXP[0]	L2RXN[0]	VSS	VSS	L2TXP[15]	L2TXN[15]	VSS	VSS	L2RXP[14]	L2RXN[14]	VSS	VSS	L2TXP[9]	L2TXN[9]	L0TXN[2]	L0TXP[2]	VSS	VSS	L0RXN[5]	L0RXP[5]	VSS	VSS	L0TXN[4]	L0TXP[4]	VSS	VSS	L0RXN[11]	L0RXP[11]	VSS
E	DNU	VSS	L2RXP[4]	L2RXN[4]	VSS	VSS	L2TXP[11]	L2TXN[11]	VSS	VSS	L2RXP[15]	L2RXN[15]	VSS	VSS	VSS	VSS	VSS	VSS	L0RXN[4]	L0RXP[4]	VSS	VSS	L0TXN[0]	L0TXP[0]	VSS	VSS	L0RXN[15]	L0RXP[15]	VSS	VSS
F	VSS	VSS	VSS	L2RXP[5]	L2RXN[5]	VSS	VSS	L2TXP[10]	L2TXN[10]	VSS	VSS	L2RXP[10]	L2RXN[10]	VSS	L2RXP[9]	L2RXN[9]	VSS	L0RXN[1]	L0RXP[1]	VSS	VSS	L0TXN[1]	L0TXP[1]	VSS	VSS	L0RXN[14]	L0RXP[14]	VSS	VSS	VSS
G	L2TXP[5]	L2TXN[5]	VSS	VSS	L2RXP[6]	L2RXN[6]	VSS	VSS	L2TXP[1]	L2TXN[1]	VSS	VSS	L2RXP[11]	L2RXN[11]	VSS	VSS	L0RXN[0]	L0RXP[0]	VSS	VSS	L0TXN[10]	L0TXP[10]	VSS	VSS	L0RXN[13]	L0RXP[13]	VSS	VSS	L0TXN[14]	L0TXP[14]
H	VSS	L2TXP[6]	L2TXN[6]	VSS	VSS	L2RXP[7]	L2RXN[7]	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	L0RXN[12]	L0RXP[12]	VSS	VSS	L0TXN[13]	L0TXP[13]	VSS
J	EXTREST P	VSS	L2TXP[7]	L2TXN[7]	VSS	VSS	VSS	VTR	VSS	VTT	VTR	VSS	VTT	VSS	VTR	VSS	VTT	VSS	VSS	VTR	VSS	VTR	VSS	VSS	VSS	VSS	L0TXN[12]	L0TXP[12]	VSS	DNU
K	VSS	EXTREST N	VSS	L2TXP[3]	L2TXN[3]	VSS	VSS	VSS	VTT	VSS	VTT	VDDPLLB2	VDDPLLA2	VDDPLLB2	VDDPLLA2	VDDPLLB0	VDDPLLA0	VDDPLLB0	VDDPLLA0	VSS	VTT	VSS	VTT	VSS	VSS	L0TXN[8]	L0TXP[8]	VSS	VSS	VSS
L	L2RXP[3]	L2RXN[3]	VSS	VSS	L2TXP[2]	L2TXN[2]	VSS	VTR	VSS	VTT	VSS	VTR	VSS	VTT	VSS	VTR	VSS	VTT	VSS	VTR	VSS	VTR	VSS	VSS	L0TXN[9]	L0TXP[9]	VSS	VSS	L0RXN[8]	L0RXP[8]
M	VSS	VSS	L2RXP[2]	L2RXN[2]	VSS	VSS	DNU	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	TST_VSS[1]	VSS	VSS	L0RXN[9]	L0RXP[9]	VSS	VSS
N	REFCLK_ BOOT[0]	VSS	VSS	VSS	VSS	L2RXPS	DNU	DNU	VDD	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	L0TXPS	DNU	VSS	VSS	VSS	VSS	VSS
P	DNU	L3RXPS	L3TXPS	DNU	DNU	TST_VSS[1]	TDI	VSS	DNU	DNU	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	CUB[1]	DNU	DNU	DNU	DNU	L1RXPS	VDDPLL R
R	DNU	VSS	DNU	VDDK	TMS	VDDK	TRST_N	DNU	VSS	VDDK	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	TST_VSS[1]	VSS	TST_VSS[1]	VSS	TST_VSS[1]	VSS	TST_VSS[1]	VDDK	SCL	VSS	REFCLK N
T	TDO	VSS	TST_VSS[1]	VSS	TCK	VSS	DNU	TST_VSS[1]	VDD	VDD	VSS	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	SDA	VDDK	TST_VSS[1]	VSS	DNU	VSS	REFCLK P
U	DNU	L2TXPS	DNU	DNU	FERR_N	TST_VSS[1]	DNU	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VCCP	VSS	DNU	DNU	L0RXPS	REFCLK_ BOOT[1]	P_RST_N
V	TST_VSS[1]	VSS	VSS	VSS	VSS	REFCLK SEL	DNU	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS	DNU	L1TXPS	CUB[0]	VSS	VSS	VSS	VSS	TST_VSS[1]
W	VSS	VSS	L3RXP[2]	L3RXN[2]	VSS	VSS	CUB[2]	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VDDM	VSS	VSS	VSS	VSS	L1RXN[9]	L1RXP[9]	VSS	VSS
Y	L3RXP[3]	L3RXN[3]	VSS	VSS	L3TXP[2]	L3TXN[2]	VSS	VSS	VTR	VSS	VTR	VSS	VTT	VSS	VTR	VSS	VTT	VSS	VTR	VSS	VTT	VSS	VTR	VSS	L1TXN[9]	L1TXP[9]	VSS	VSS	L1RXN[8]	L1RXP[8]
AA	VSS	EXTRES BN	VSS	L3TXP[3]	L3TXN[3]	VSS	VSS	VTT	VSS	VTT	VSS	VDDPLLB3	VDDPLLA3	VDDPLLB3	VDDPLLA3	VDDPLLB1	VDDPLLA1	VDDPLLB1	VDDPLLA1	VTT	VSS	VTT	VSS	VSS	VSS	L1TXN[8]	L1TXP[8]	VSS	VSS	VSS
AB	EXTRESB P	VSS	L3TXP[7]	L3TXN[7]	VSS	VSS	VSS	VSS	VTR	VSS	VTR	VSS	VSS	VTT	VSS	VTR	VSS	VTT	VSS	VTR	VTT	VSS	VTR	VSS	VSS	VSS	L1TXN[12]	L1TXP[12]	VSS	DNU
AC	VSS	L3TXP[6]	L3TXN[6]	VSS	VSS	L3RXP[7]	L3RXN[7]	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	L1RXN[12]	L1RXP[12]	VSS	VSS	L1TXN[13]	L1TXP[13]	VSS
AD	L3TXP[5]	L3TXN[5]	VSS	VSS	L3RXP[6]	L3RXN[6]	VSS	VSS	L3TXP[1]	L3TXN[1]	VSS	VSS	L3RXP[11]	L3RXN[11]	VSS	VSS	L1RXN[0]	L1RXP[0]	VSS	VSS	L1TXN[10]	L1TXP[10]	VSS	VSS	L1RXN[13]	L1RXP[13]	VSS	VSS	L1TXN[14]	L1TXP[14]
AE	VSS	VSS	VSS	L3RXP[5]	L3RXN[5]	VSS	VSS	L3TXP[10]	L3TXN[10]	VSS	VSS	L3RXP[10]	L3RXN[10]	VSS	L3RXP[9]	L3RXN[9]	VSS	L1RXN[1]	L1RXP[1]	VSS	VSS	L1TXN[1]	L1TXP[1]	VSS	VSS	L1RXN[14]	L1RXP[14]	VSS	VSS	VSS
AF	DNU	VSS	L3RXP[4]	L3RXN[4]	VSS	VSS	L3TXP[11]	L3TXN[11]	VSS	VSS	L3RXP[15]	L3RXN[15]	VSS	VSS	VSS	VSS	VSS	VSS	L1RXN[4]	L1RXP[4]	VSS	VSS	L1TXN[0]	L1TXP[0]	VSS	VSS	L1RXN[15]	L1RXP[15]	VSS	VSS
AG	VSS	L3RXP[0]	L3RXN[0]	VSS	VSS	L3TXP[15]	L3TXN[15]	VSS	VSS	L3RXP[14]	L3RXN[14]	VSS	VSS	L3TXP[9]	L3TXN[9]	L1TXN[2]	L1TXP[2]	VSS	VSS	L1RXN[5]	L1RXP[5]	VSS	VSS	L1TXN[4]	L1TXP[4]	VSS	VSS	L1RXN[11]	L1RXP[11]	VSS
AH	L3RXP[1]	L3RXN[1]	VSS	VSS	L3TXP[0]	L3TXN[0]	VSS	VSS	L3RXP[13]	L3RXN[13]	VSS	DNU	L3TXP[8]	L3TXN[8]	VSS	VSS	L1TXN[3]	L1TXP[3]	VSS	VSS	L1RXN[6]	L1RXP[6]	VSS	DNU	L1TXN[11]	L1TXP[11]	VSS	VSS	L1RXN[10]	L1RXP[10]
AJ	VSS	VSS	VSS	L3TXP[14]	L3TXN[14]	VSS	VSS	L3RXP[12]	L3RXN[12]	VSS	VSS	L3TXP[12]	L3TXN[12]	VSS	VSS	VSS	VSS	L1TXN[7]	L1TXP[7]	VSS	VSS	L1RXN[7]	L1RXP[7]	VSS	VSS	L1TXN[5]	L1TXP[5]	VSS	VSS	VSS
AK		VSS	L3TXP[4]	L3TXN[4]	VSS	VSS	L3RXP[8]	L3RXN[8]	VSS	VSS	L3TXP[13]	L3TXN[13]	VSS	VSS	L1RXN[2]	L1RXP[2]	VSS	VSS	L1TXN[6]	L1TXP[6]	VSS	VSS	L1RXN[3]	L1RXP[3]	VSS	VSS	L1TXN[15]	L1TXP[15]	VSS	

Note: 1. The ballout represents an x-ray view, looking through the package down to the balls.

3.6.2.3 Supply Rails

HMCs are constructed with multiple layers of silicon and a substrate. Multiple power supplies are required to power the device.

3.6.2.4 Independent Supply Rails

Reflect nominal values that include both DC and AC components.

- 0.9V supply (1 each): V_{DD} (logic core supply)
- 1.2V supplies (6 each): 1) V_{DDM} (DRAM source); 2) $V_{DDPLLA(xx)}$ (link PLLA source); 3) $V_{DDPLLB(xx)}$ (link PLLB source); 4) V_{DDPLLR} (intermediate frequency PLL source); 5) V_{TR} (link RX termination source); 6) V_{TT} (link TX termination source);
- 2.5V supply (1 each): V_{CCP} (DRAM wordline boost source)
- 1.5V supply (1 each): V_{DDK} (JTAG, NVM, I^2C source)

To ensure the most optimal and cleanest voltage regulation, Micron recommends using separate power nodes for each power supply.

Current for each rail can vary greatly based on the number of active links, lane speed, configuration, and usage conditions. Micron suggests following the PDN simulation kit guidance given in this section to optimize the PDN design.

3.6.3 Layout

This section uses a sample board designed for an HMC Gen2 4-link device to describe Micron's recommendations for the HMC system board layout.

3.6.3.1 Getting Started

The HMC short reach physical interface is based on robust, tolerant transceivers that provide a wide range of flexibility. Optimally, nominal trace differential impedance should be 100Ω. In the sample boards shown in this document, in order to make more efficient use of the routing space, we reduced the nominal impedance to 90Ω with a tolerance of ±10%.

For the stackup, we used 1oz copper for all layers; however, due to via plating, the top and bottom layers ended up slightly heavier. All dielectric material was standard FR4 with varying thickness as shown in the 12-layer Board Stackup Using Standard FR4-Type Materials table.

All signals were striplines that had their own ground reference. To achieve the 90Ω impedance with good crosstalk immunity, the trace widths were 70µm with a differential spacing of 120µm (approximately 3 mils and 5 mils respectively).

Table 15: 12-Layer Board Stackup Using Standard FR4-Type Materials

Subclass Name	Type	Thickness (mm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (mm)	Impedance (ohms)	Coupling Type	Spacing (mm)	DifZ0 (ohms)
–	Surface	–	1	0	–	–	–	–	–	–	–
–	Dialectric	0.02	2.6	0.035	–	–	–	–	–	–	–
Top	Conductor	0.045	2.6	0.035	–	–	0.080	57.723	EDGE	0.110	92.536
–	Dialectric	0.08	4.3	0.035	–	–	–	–	–	–	–
GND2	Plane	0.015	4.3	0.035	–	X	–	–	–	–	–
–	Dialectric	0.1	4.3	0.035	–	–	–	–	–	–	–
S3	Conductor	0.015	4.3	0.035	–	–	0.075	50.392	EDGE	0.100	89.833
–	Dialectric	0.1	4.3	0.035	–	–	–	–	–	–	–
GND4	Plane	0.015	4.5	0.035	–	X	–	–	–	–	–
–	Dialectric	0.1	4.5	0.035	–	–	–	–	–	–	–
S5	Conductor	0.015	4.5	0.035	–	–	0.080	54.552	EDGE	0.100	88.871
–	Dialectric	0.3	4.5	0.035	–	–	–	–	–	–	–
VIP2	Plane	0.015	4.3	0.035	–	X	–	–	–	–	–
–	Dialectric	0.1	4.3	0.035	–	–	–	–	–	–	–
VM	Plane	0.015	4.3	0.035	–	X	–	–	–	–	–
–	Dialectric	0.3	4.5	0.035	–	–	–	–	–	–	–
S8	Conductor	0.015	4.5	0.035	–	–	0.090	54.552	EDGE	0.100	88.871
–	Dialectric	0.1	4.5	0.035	–	–	–	–	–	–	–
GND9	Plane	0.015	4.5	0.035	–	X	–	–	–	–	–
–	Dialectric	0.1	4.3	0.035	–	–	–	–	–	–	–
S10	Conductor	0.015	4.3	0.035	–	–	0.075	50.392	EDGE	0.100	89.833
–	Dialectric	0.1	4.3	0.035	–	–	–	–	–	–	–
GND11	Plane	0.015	4.3	0.035	–	X	–	–	–	–	–
–	Dialectric	0.08	4.3	0.035	–	–	–	–	–	–	–
Bottom	Conductor	0.045	2.6	0.035	–	–	0.100	53.026	NONE	–	–

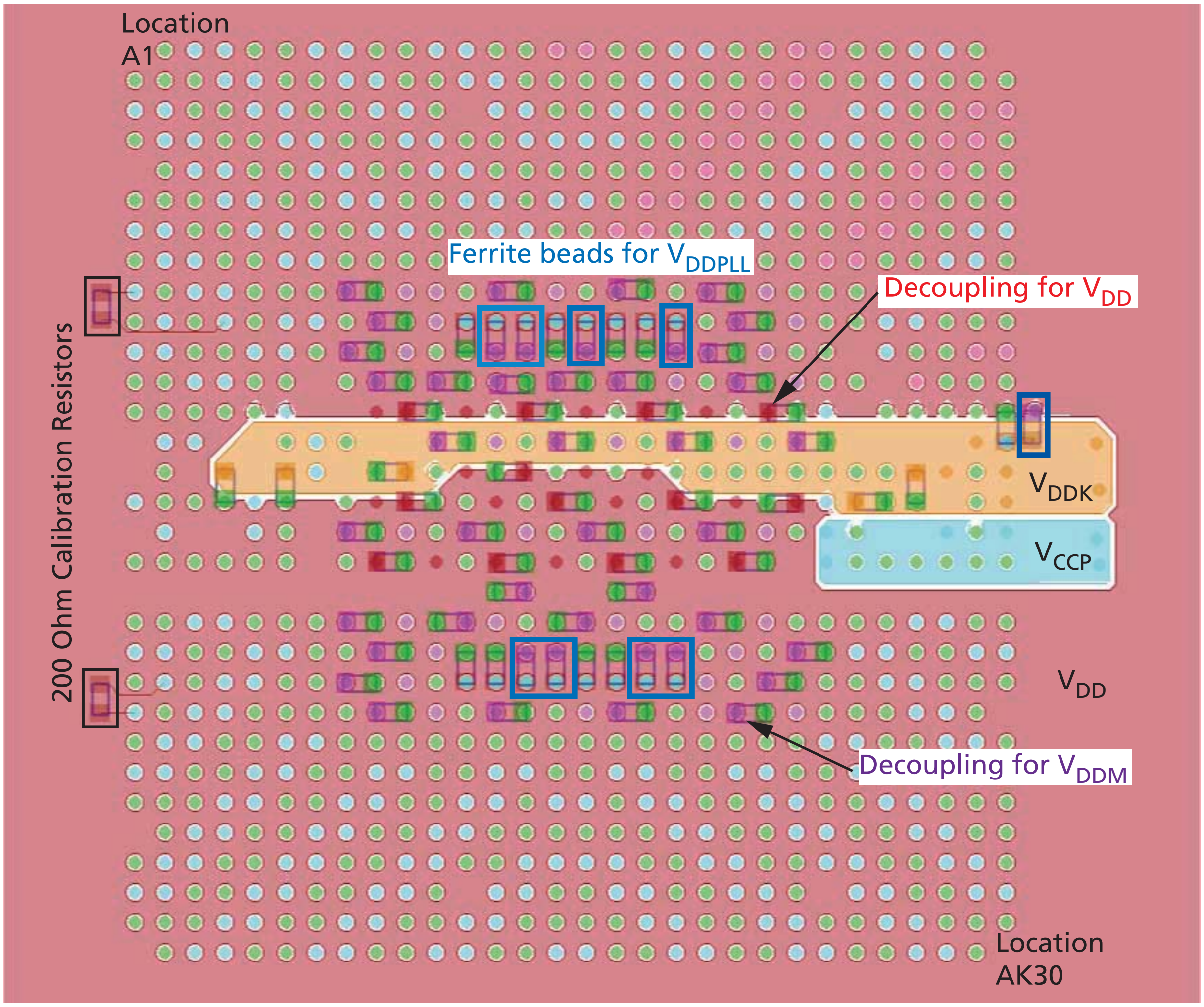
In summary, for the 31mm x 31mm package, a 12-layer PCB was used for routing HMC to HMC. Routing from the host controller to the cube can be achieved with four signal layers.

3.6.3.2 Power Planes

HMC-to-HMC routing was evaluated to determine how closely multiple devices can be placed together while using few routing layers. The 31mm x 31mm package has a 1mm ball pitch and is for designs that require the maximum memory bandwidth, which are typically the applications that also use higher layer count PCBs. The worst-case routing is between like-HMC devices because the route between an HMC and a customized host is assumed to support a mirrored ballout that result in more direct traces, as shown in Figure 51 (page 84).

For the 31mm x 31mm, 1mm pitch package, the two center layers for power planes were used. The layer labeled VM was split into three power rails. Figure 39 shows an isolated island for V_{DDK} (1.5V) and V_{CCP} (2.5V). The rest of the plane is for the logic V_{DD} (0.9V). In addition, the discrete components for decoupling and ferrite beads for V_{DDPLL} isolation were placed on the back side of the board and directly on the associated via/pad. In this design the 0201 and 0402 packages were used, as applicable.

Figure 39: Power Plane 2 – 31mm x 31mm Package



It is possible to connect all of the 1.2V rails together on the same plane. However, for most designs (particularly under full bandwidth conditions and with several devices in use), there will be independent islands for groups of 1.2V supply rails. This will help with power regulation.

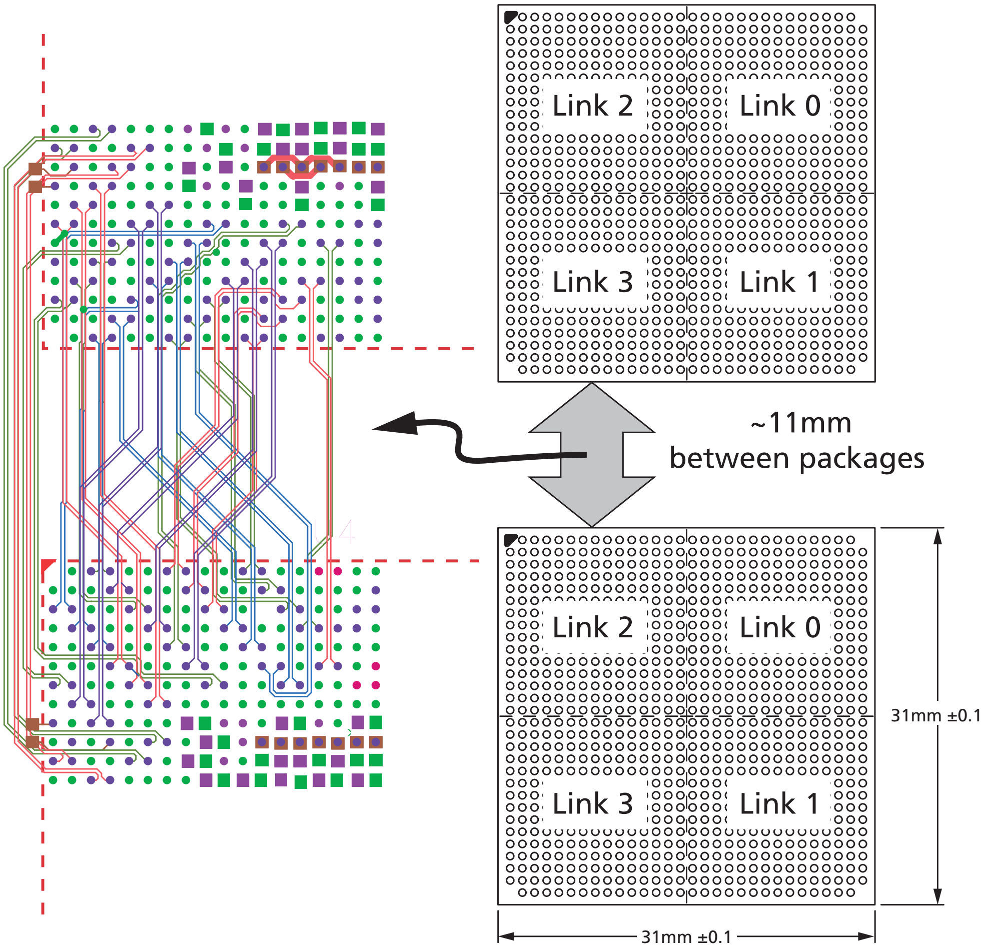
When the rails are separated, we suggest using three islands: One for V_{DDM} for the memory supply, one for V_{TT} and V_{TR} termination for the high-speed channels, and one for the V_{DDPLLx} (which is for the internal PLLs).

3.6.3.3 High Speed Routes

For every HMC link interface, there are up to a total of 32 high-speed differential pairs (16 pairs in, 16 pairs out) to be routed. This reflects a full-width link in both directions. Other options include half-width links (8 pairs in and 8 pairs out) or asymmetrical links where there could be a full-width in one direction and a half-width in the other.

The figure below shows how to chain two 31mm x 31mm HMC devices together through a single link. The 31mm x 31mm package requires four internal signal layers to achieve a full link route of 32 differential pairs. This makes the total number of layers 12 (4 signal layers, 2 supply planes, 4 ground planes and the top/bottom conductors).

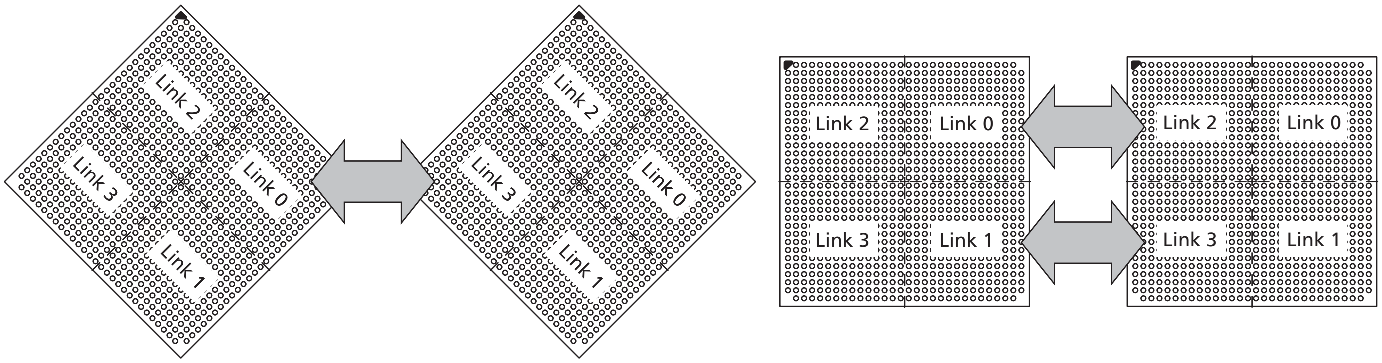
Figure 40: Breakout and Routing of the 31mm x 31mm Package



HMC-to-HMC Connections

As shown in the HMC-to-HMC Connections figure below, other placements or configurations are possible, but the 12-layer PCB count is expected to remain the same.

Figure 41: HMC-to-HMC Connections



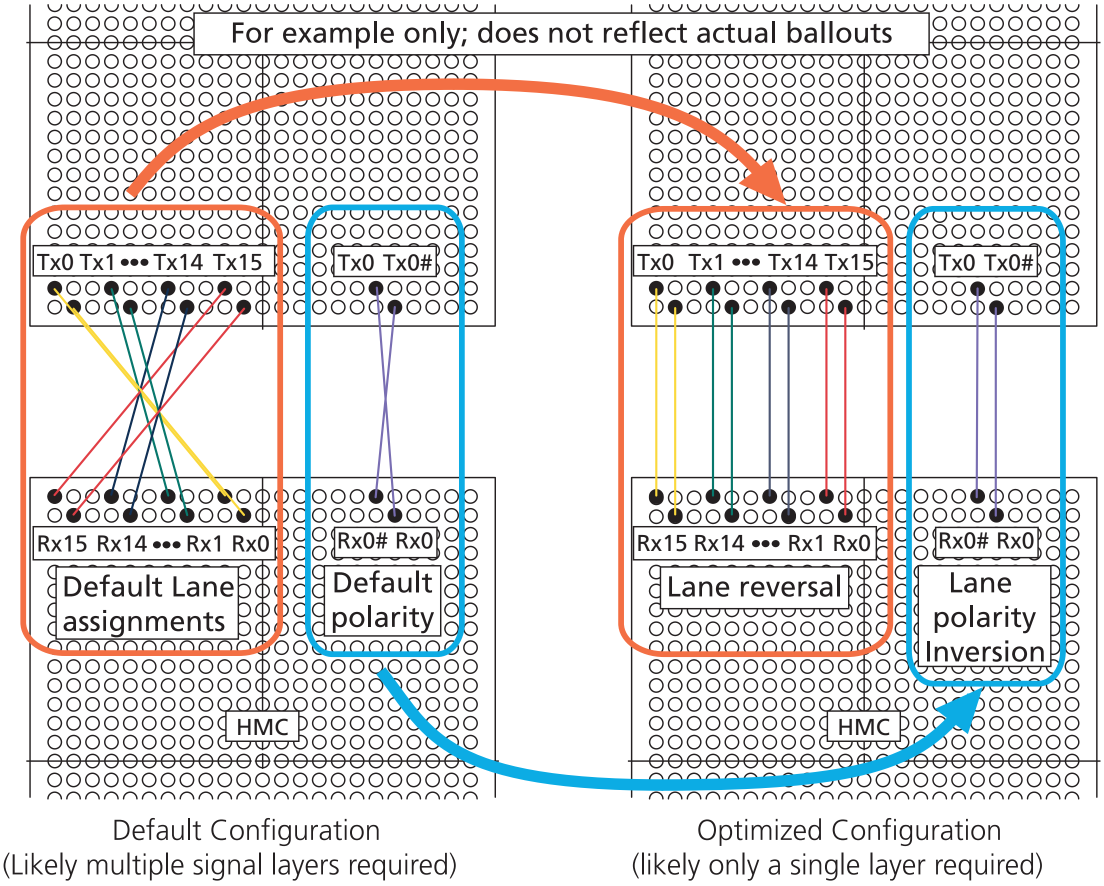
The difficulty with the 31 x 31mm package is due to the ball pitch because there are only 15 balls within 16mm of space that must share all signals and associated grounds. This means the placement of the TX or RX pairs fall deeper into the center array of the package, which requires many of the signal breakouts either to be stacked above each other on different planes or to exit from the side of the package.

Lane Reversal and Lane Polarity

HMC also supports both lane reversal and lane polarity, which may help simplify routing in circumstances where signal mismatches occur. For example, if the HMC is connected to a device where the full set of lanes (LSB, MSB, and sequentially all bits in between) are exactly reversed on the connecting device.

As shown in Figure 42 (page 77) below, rather than connecting TX0 to RX0, TX1 to RX1 and so on, it is acceptable to connect TX0 to RX15, TX1 to RX14, and so on. When the HMC powers up and completes the training sequence, it will detect that all lanes within the link are exactly reversed and it will perform the lane reversal internally. The device also has the capability to detect the polarity within each lane and internally swap the polarity as needed.

Figure 42: Using Lane Reversal and Lane Polarity to Simplify Routing



3.6.4 Channel Evaluation – Introduction

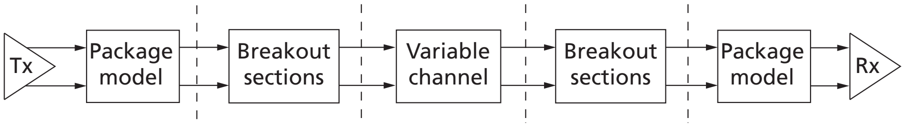
A major part of the layout is validation of functionality over all corners of operation, which often requires several iterations of the board design prior to final release. The following sections discuss our simulation results and provide a baseline to help streamline new designs.

3.6.4.1 Channel Evaluation – Signal Integrity Setup

For the package layout we broke the models into five blocks, as shown in Figure 43. IBIS-AMI models for HMC TX and RX are used to reflect the true transmitter and receiver characteristics. Additionally, the true typical TX and RX packages are used at each end of the channel. The 4-link package is simulated and monitor the system margin. The PCB includes the unique breakout for the package and the run length for the span of the package.

For each individual block we chose the worst-case node or trace. The trace-length limits ranged from very close proximity micro-strip to a maximum distance of 8 inch of strip line. The maximum insertion loss the HMC Gen2 device supports is 12dB from BGA to BGA, which is about the same as the insertion loss of the 8-inch strip line used in the simulation.

Figure 43: Block for Full Channel Simulations



The package models reflect the behavior as presented by a combination of the substrate’s multiport, high-frequency, s-parameter model. The breakout is without any external AC coupling and is routed to minimize skew within pairs. When the variable trace block is extended the maximum distance, standard routing practices are assumed with no nonuniform tight bends, right angles, or inconsistent skews between pairs. Figure 44 and Figure 45 show the insertion loss and return loss of the different traces used in the simulation.

Figure 44: Insertion Loss of Channels

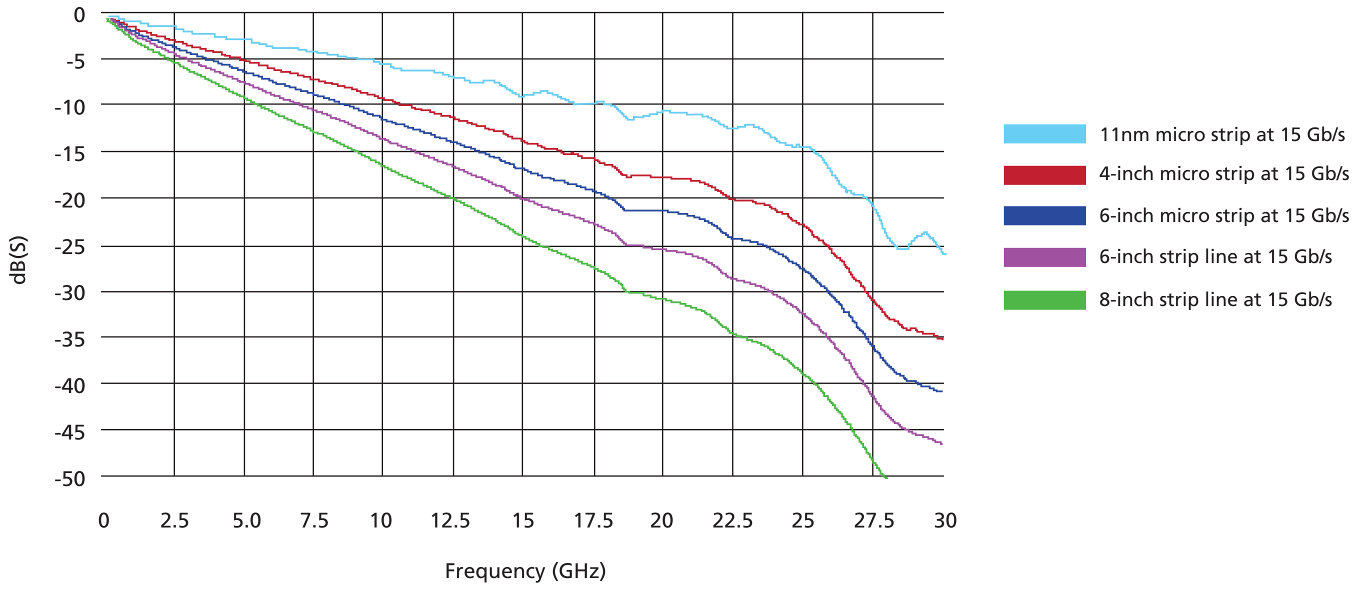
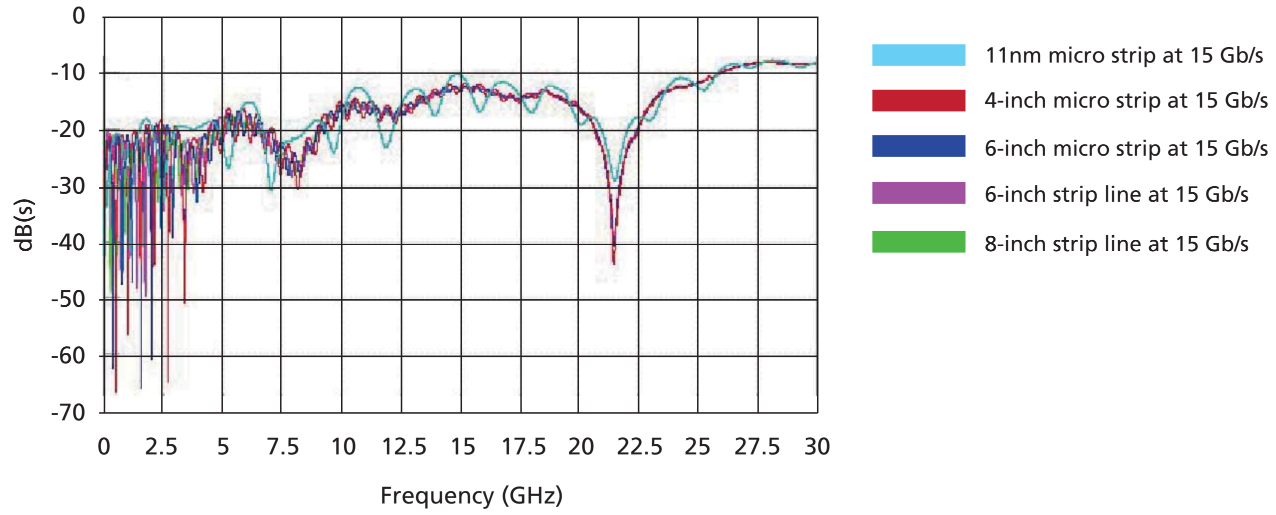


Figure 45: Return Loss of Channels



3.6.4.2 Signal Integrity Results

The simulation for the 4-link package is presented. All simulations reflect the HMC Gen2 IBIS-AMI models for both ends of the channel. Simulations with different TX or RX models will result in a different margin.

The simulations were running at 15 Gb/s, which reflects the fastest lane speed supported by the HMC Gen2 interface. We evaluated margins at bit error rates of 1E^{-12}, the target BER for HMC Gen2. Simulation was running at the slow corner of the PVT to capture the worst-case scenario.

Figure 46: Slow Corner for 11nm Micro Strip at 15 Gb/s

BER	HeightAtBERList	WidthAtBERList
1.000E-12	0.501	3.167E-11
1.000E-9	0.520	3.300E-11
1.000E-6	0.553	3.567E-11

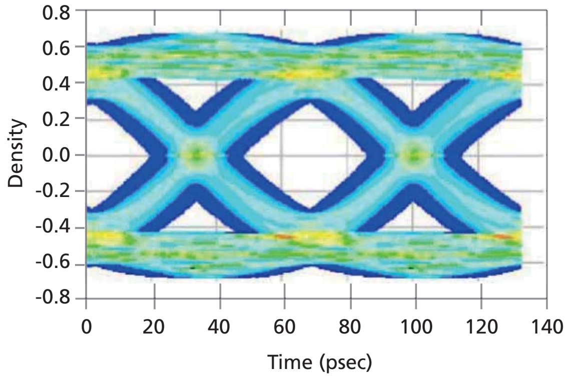


Figure 47: Slow Corner for 4-inch Micro Strip at 15 Gb/s

BER	HeightAtBERList	WidthAtBERList
1.000E-12	0.359	3.100E-11
1.000E-9	0.368	3.233E-11
1.000E-6	0.385	3.500E-11

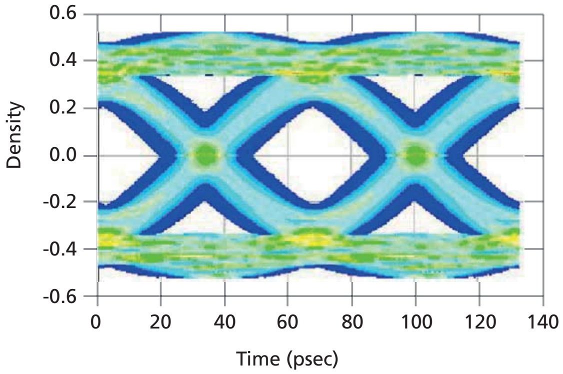


Figure 48: Slow Corner for 6-inch Micro Strip at 15 Gb/s

BER	HeightAtBERList	WidthAtBERList
1.000E-12	0.286	3.000E-11
1.000E-9	0.292	3.133E-11
1.000E-6	0.307	3.433E-11

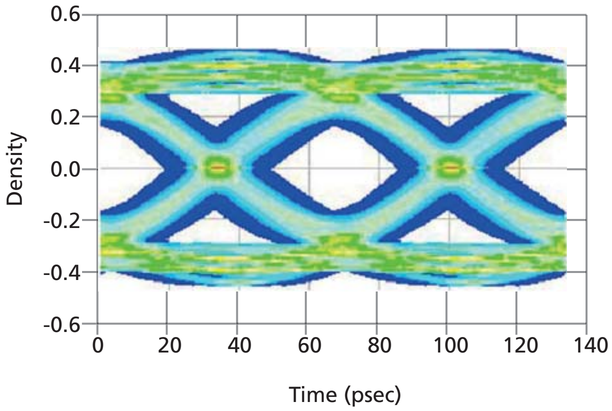


Figure 49: Slow Corner for 6-inch Strip Line at 15 Gb/s

BER	HeightAtBERList	WidthAtBERList
1.000E-12	0.212	2.700E-11
1.000E-9	0.219	2.833E-11
1.000E-6	0.232	3.200E-11

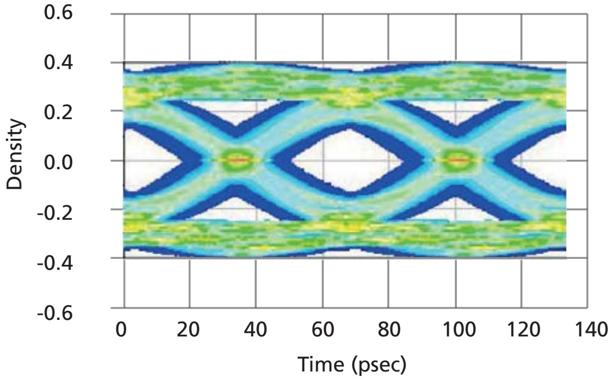
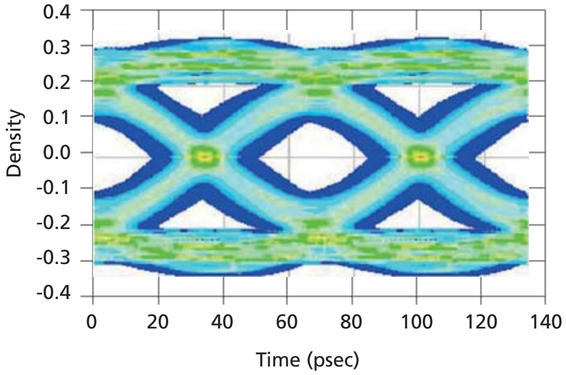


Figure 50: Slow Corner for 8-Inch Strip Line 15 Gb/s

BER	HeightAtBERList	WidthAtBERList
1.000E-12	0.157	2.600E-11
1.000E-9	0.162	2.733E-11
1.000E-6	0.169	3.033E-11



Evaluation of the slow corner at 15 Gb/s – the 31mm x 31mm package shows there is large margin for the layout when running a trace length of 11nm micro strip to 8 inches of strip Line.

At BER $1E^{-12}$, the eye width (~0.39UI) is more than twice the minimum required eye width (0.15UI), and the eye height (157mv/163mv) is about 8 times the minimum required eye height (20mv).

The simulation does not include the crosstalk impact of the channels and the package. When crosstalk is included in the simulation, the margin will decrease; however, the margin from the simulation provides reasonable room from for board/package designers to handle the typical crosstalk introduced through board and package.

Table 16: Complete Simulation Results for the 31 x 31 mm Package with Different Traces at BER 1E^{-12}

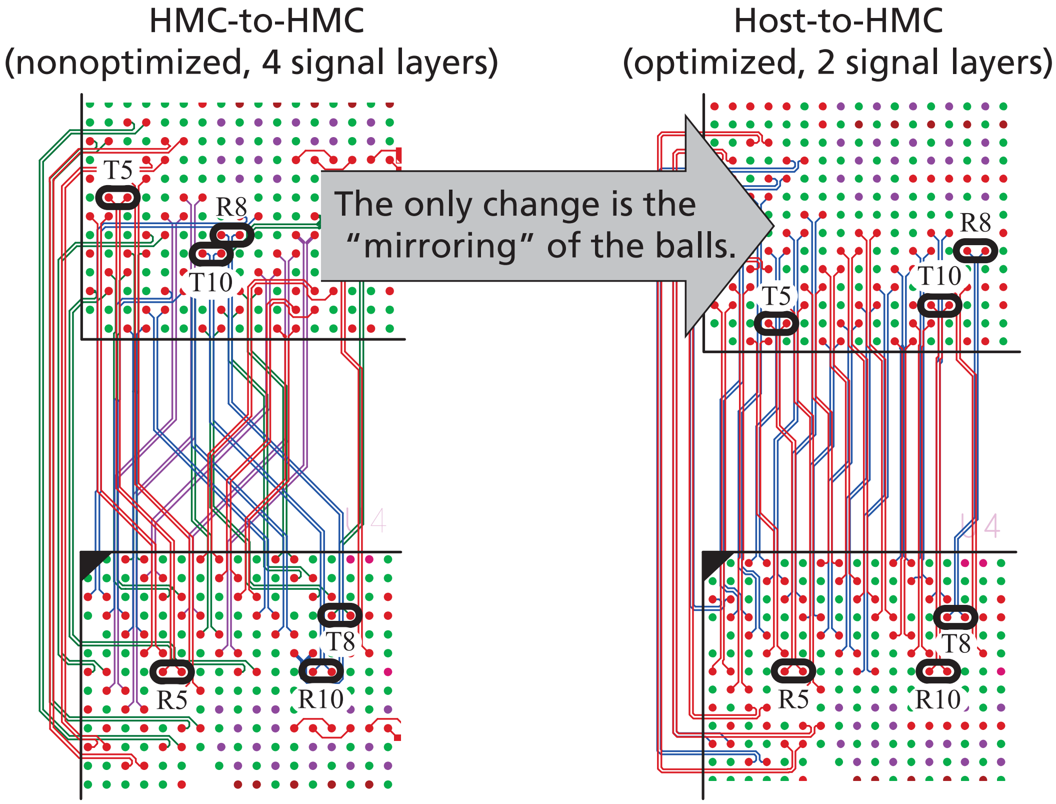
	31 x 31 mm Package		
	Eye Width (ps)	Eye Width (UI)	Eye Height (mv)
11mm micro strip	31.67	0.48	501.00
4-inch micro strip	31.00	0.47	359.00
6-inch micro strip	30.00	0.45	286.00
6-inch strip Line	27.00	0.41	212.00
8-inch strip Line	26.00	0.39	157.00

Note: 1. Requirement for HMC PHY is a 15% UI with a minimum of 20mV peak-to-peak amplitude.

3.6.5 Channel Route (Host-to-HMC) – Improved Route

The route between two HMC devices is limited by the HMC ballouts. However, on a new design, assuming the host has the ability to place transmit and receive balls in optimal locations, the improvements in board layout options can be remarkable. For example, the route between two 31mm x 31mm packages requires four signal layers because there are several signals that must transverse the plane against the grid. With the freedom to place each of the transmit and receive balls in a mirror image of the adjacent HMC, the required layers for signals will drop from four to only two. Figure 51 below shows a side-by-side comparison.

Figure 51: Example: Host-to-HMC Route – Two Layers


HMC-to-HMC
(nonoptimized, 4 signal layers)
Host-to-HMC
(optimized, 2 signal layers)
The only change is the "mirroring" of the balls.
T5
R8
T10
T8
R5
R10
R8
T10
T5
T8
R5
R10

Note: 1. Ball placement on the HMC does not change. All TX and RX pairs are routed, but the names of only a few are shown for simplicity.

3.6.6 Board Routing Conclusion

The signal integrity simulation shows that for an HMC-to-HMC connection, ranging from 11mm micro strip to 8-inch strip line, the HMC TX and RX can provide sufficient margin for a robust connection.

As with any design, it is important to use simulations as a way of validating an HMC Gen2 design. This section has demonstrated that the 31 x 31mm package supports simple layouts using standard FR4-type board materials with good signal integrity margin. Key takeaways include:

- Each HMC (full) link requires 32 differential pairs and a few power state control signals
- HMC supports full lane reversal and lane polarity options to aid layout
- Ideal impedance is 100Ω, but 90Ω (nominal for traces up to 8 inches) may be used
- In the operating range of 10–15 Gb/s, traces with up to 12dB insertion loss at the Nyquist frequency and less than -15db return loss up to the Nyquist frequency, are demonstrated to be successfully equalized by HMC equalization schemes
- Depending on power, the 1.2V planes may or may not be from a single source
- Sharing V_{DDPLL} supplies with other power rails is not recommended. Isolating each V_{DDPLL} power rail with a ferrite bead is recommended
- HMC can operate with a post equalization eye (at target BER, for example, $1E^{-12}$) of approximately 15% UI and 20mV peak-to-peak amplitude
- To achieve two signal layers when designing a host controller, it is important to mirror HMC ballout

4 Chapter 4 – Board Bring-Up, Software, Register Definitions

This section is intended to aid in the process of configuring, bringing up and debugging systems using HMC Gen2 devices. It contains a best-practices checklist, which is strongly recommended reading, and contains pointers to other parts of the HMC Gen2 User Guide that should receive added attention.

The Register Definitions section contained in this chapter was previously a standalone document, titled Register Addendum, but has been incorporated into this guide entirely. It is meant to assist the designer who is responsible for writing software to interface with the HMC's direct and indirect registers via sideband communications (I^2C or JTAG). It also describes how to use some of the features that are discussed in earlier chapters, such as the built-in self test or DRAM repairs.

4.1 Board Bring-Up Checklist

1. Verify that the voltage rails ramp up in the sequence specified in the data sheet
2. Verify that all the voltage rails meet the AC and DC data sheet specifications
3. Verify reference clock frequency is correct and that the signals meet the data sheet specifications
4. Verify I^{2}C or JTAG bus communication
    a. Verify analog signal quality using an oscilloscope
    b. Read device information status register (0x2c0003) to verify communication protocol
    c. Write the direct configuration registers required for initialization
        1. Disable NVM write (Direct register 0x280002) to avoid writing permanent records to the NVM during initial board bring up
    d. Program (write) indirect (ERI) configuration registers required for initialization
        1. Verify successful completion of ERI (bit [31] = 0 and external request status bits [30:26] = 0x0)
5. Device initialization and SerDes bring-up
    a. Execute Init Continue ERI
    b. Verify successful completion of ERI (bit [31] = 0 and external request status bits [30:26] = 0x0)
        1. If external request status of 0x12 (link initialization error) is returned
            a. Verify if extending the timeout delay (INIT Continue, ERIDATA0 bits [18:0]) allows the link initialization to complete
            b. Verify that the host operates correctly in internal loopback (host TX to host RX) mode
6. Run built-in self test (BIST) to verify proper DRAM operation
    a. Execute BIST without repair ERI
    b. Verify successful completion of ERI (bit [31] = 0 and external request status bits [30:26] = 0x0)
        1. Read ERIDATA1 and ERIDATA2 registers (Steps 5 and 6 of Repair ERI) to determine the number of errors reported
            a. Typical HMC devices should not have any errors
            b. Total errors (for all16 vaults) in excess of 1 or 2 errors normally indicates some sort of system issue (that is, improper power supply level, incorrect reference clock frequency, incorrect configuration, and so on)
    c. In the rare event that one or two errors are found BIST with repair can be executed to repair these errors
        1. Prior to running BIST with Repair, NVM write must be enabled or repairs will not be made
7. Verify that PRBS pattern(s) execute without error
    a. Read the data eye metrics (height and width) for each channel to verify adequate data eye size
8. Verify that simulated traffic executes without error

9. If any voltage or temperature corner testing is going to be done as part of your qualification process, make sure to disable NVM write to avoid filling up the non-volatile memory space

4.2 Software Design/Verification Checklist

1. Verify the packet protocol by using the 2-in-1 model
2. Verify proper HMC sideband communication via I^{2}C/JTAG bus
   a. Direct register access and Indirect register access (ERI) configuration/commands
   b. Initialization sequence (what happens before and after INIT Continue), refer to Register Definitions section for more information
   c. Firmware patching support
3. Interrupt handling (FERR_N) and Errstat logging
4. Boundary scan verification (if applicable)
   a. BSDL file available on micron.com for connectivity verification

4.3 JTAG Interface

HMC incorporates a standard test access port (TAP) controller that operates in accordance with IEEE Standard 1149.1. The input and output signals of the test access port use V_{DDK} as a supply.

The JTAG test access port uses the TAP controller on the HMC, from which the Test Data Registers (TDR), Instruction, Bypass, ID, Boundary-Scan, and CADATA can be selected. Each of these functions of the TAP controller is described in detail below.

The HMC includes boundary scan support that complies with the IEEE 1149.1 and 1149.6 standards. Boundary scan chain order is provided within BSDL files for each HMC configuration.

Warning: Use extreme caution when executing JTAG instructions during normal operation.

Executing certain JTAG instructions can disrupt normal operational functionality of the HMC. The HMC does not block JTAG instructions when in normal operational mode, that is, boundary scan instructions such as, EXTEST/CLAMP can disrupt the REFCLK input, and EXTEST_PULSE/TRAIN can cause link errors.

4.3.1 Disabling the JTAG Interface

It is possible to operate HMC without using the JTAG interface. To disable the TAP controller, tie the test reset signal, TRST_N, LOW. TCK, TDI, and TMS may all be considered “Don’t Care” when TRST_N is LOW. TDO should be left unconnected. At power-up, the TAP controller will come up in a reset state; this will not interfere with the operation of the HMC device.

4.3.2 Test Access Port (TAP)

Test Clock (TCK)

The test clock is used only with the test logic. All inputs are captured on the rising edge of TCK. All outputs are driven on the falling edge of TCK.

Test Mode Select (TMS)

The TMS input is used to issue commands to the TAP controller and is sampled on the rising edge of TCK.

All of the states in the TAP Controller State Diagram Figure 52 (page 91) are entered through the serial input of the TMS pin. A 0 in the diagram represents a LOW on the TMS pin during the rising edge of TCK; a 1 represents a HIGH on TMS.

Test Reset (TRST_N)

The TRST_N input provides asynchronous initialization of the TAP controller. During power-up, TRST_N must be asserted LOW and not be de-asserted HIGH until after P_RST_N is de-asserted HIGH. To ensure deterministic operation of the test logic, TMS should be held HIGH while the signal applied at TRST_N changes from LOW to HIGH.

If P_RST_N is asserted LOW after the HMC has been initialized, the JTAG port will perform normally with the exception of the CFG_RDA and CFG_WRA commands. CFG_RDA and CFG_WRA JTAG commands are specific to Micron's HMC device and provide READ/WRITE access to internal configuration and status registers. Because those registers are held in RESET while P_RST_N is asserted, CFG_RDA and CFG_WRA should not be used until after P_RST_N is de-asserted.

Test Data-In (TDI)

The TDI pin is used to input test instructions and data into the registers serially and can be connected to the input of any Test Data Register. The register between TDI and TDO is selected by the instruction that is loaded into the instruction register. For information on loading the instruction register, see Figure 56 (page 97) JTAG Operation – Loading Instruction Code and Shifting Out Data figure. TDI is connected to the most significant bit (MSB) of any register (see Figure 53 (page 91) Test Logic Block Diagram).

Test Data-Out (TDO)

The TDO output pin is used to clock test instructions and data serially out of the registers. The TDO output driver is only active during the Shift-IR and Shift-DR TAP controller states. In all other states, the TDO pin is in a High-Z state. The output changes on the falling edge of TCK. TDO is connected to the least significant bit (LSB) of any register (see Figure 53 (page 91) Test Logic Block Diagram).

4.3.3 TAP Controller

The TAP controller is a finite state machine that uses the state of the TMS pin at the rising edge of TCK to navigate through its various operating modes. The TAP Controller State Diagram is provided in Figure 52 (page 91). Each state is described in detail below.

Test-Logic-Reset

The Test-Logic-Reset controller state is entered when TMS is held HIGH for at least five consecutive rising edges of TCK. As long as TMS remains HIGH, the TAP controller remains in the Test-Logic-Reset state. The test logic is inactive during this state.

Run-Test/Idle

The Run-Test/Idle is a controller state in between scan operations. This state can be maintained by holding TMS LOW. From here, either the data register scan or the instruction register scan can be selected.

Select-DR-Scan

Select-DR-Scan is a temporary controller state. All test data registers retain their previous state while here.

Capture-DR

The Capture-DR state is where the data is parallel-loaded into the selected test data register.

Shift-DR

Data is shifted serially through the selected data register while in this state. As new data is input through the TDI pin, data is shifted out of the TDO pin.

Exit1-DR, Pause-DR, and Exit2-DR

The purpose of Exit1-DR is to provide a path for return to the Run-Test/Idle state (through the Update-DR state). The Pause-DR state is entered when there is a need to suspend data shifting through the selected test registers. When data shifting is to reconvene, the controller enters the Exit2-DR state and can then re-enter the Shift-DR state.

Update-DR

Data is parallel loaded from the shift register to the parallel output register on the falling edge of TCK in the Update-DR controller state.

Instruction Register States

The instruction scan states of the TAP controller are similar to the data scan states. The desired instruction is shifted serially into the instruction register during the Shift-IR state and is loaded during the Update-IR state.

Figure 52: TAP Controller State Diagram

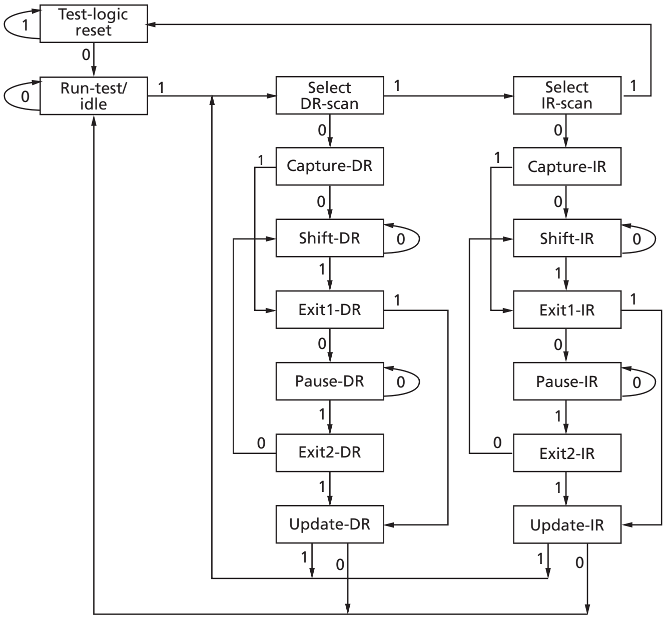
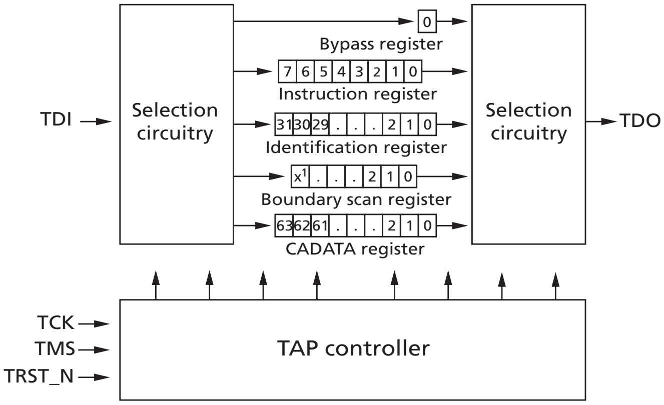


Figure 53: Test Logic Block Diagram



Note: 1. The length/size or number of bits used in the boundary scan register is provided in the device specific BSDL file.

4.3.4 Performing a TAP RESET

The TAP RESET is performed by forcing TMS HIGH (V_{DDK}) for five rising edges of TCK. This RESET does not affect the operation of the HMC and can be performed during HMC operation.

The TAP controller is also reset when TRST_N is driven LOW, which ensures that TDO comes up in a High-Z state.

4.3.5 TAP Registers

TAP Registers are connected between the TDI and TDO pins and enable data scanning into and out of the HMC test logic. Only one register can be selected at a time through the instruction register. Data is serially loaded into the TDI pin on the rising edge of TCK. Data is output on the TDO pin on the falling edge of TCK.

Instruction Register

Eight-bit instructions are loaded serially into the instruction register. This register is loaded during the Update-IR state of the TAP controller. Upon power up, the instruction register is loaded with the IDCODE instruction. It is also loaded with the IDCODE instruction if the controller is placed in a reset state as described in the previous section.

When the TAP controller is in the Capture-IR state, the two LSBs are loaded with a binary 01 pattern to accommodate fault isolation of the board-level serial test data path.

Bypass Register

To save time when shifting data serially through registers, it is sometimes advantageous to skip certain chips. The bypass register is a single-bit register that can be placed between the TDI and TDO pins. This enables data shifting through the HMC with minimal delay. The bypass register is set LOW (V_{SS}) during the Capture-DR state.

Identification (ID) Register

The ID register is loaded with a vendor-specific, 32-bit code during the Capture-DR state when the IDCODE command is loaded in the instruction register. The IDCODE is hardwired into the HMC and can be shifted out when the TAP controller is in the Shift-DR state. The ID register has a vendor code and other device-specific information described in the Table 18 (page 98).

Boundary-Scan Register

The Boundary-Scan register is loaded with the contents of the I/O ring when the TAP controller is in the Capture-DR state and is then placed between the TDI and TDO pins when the controller is moved to the Shift-DR state.

The behavior of the Boundary-Scan register is completely specified by the IEEE 1149.1 standard.

CADATA Register

CADATA is a 64-bit register used for configuration and status register access. This register is not defined by the IEEE 1149.1 standard.

Table 17: CADATA Register

Register Name	Sub-block	Description
CADATA	ADRS[63:32]	Stores the address for configuration and status register access.
	DATA[31:0]	1) Stores data to be written during configuration write operations. 2) Stores results of configuration read operations.

4.3.6 TAP Instruction Set

4.3.6.1 Overview

Many different instructions are possible with the 8-bit instruction register. All combinations used are listed in the Instruction Codes table and described in detail below. The remaining instructions are reserved and must not be used.

The TAP controller used in HMC Gen2 is fully compliant with the 1149.1 specification.

Instructions are shifted into the TAP controller during the Shift-IR state when the selected instruction register is placed between TDI and TDO. To execute the instruction after it is shifted into the controller, the TAP controller must be moved into the Update-IR state.

EXTEST

The EXTEST instruction enables circuitry external to the component package to be tested. Boundary-Scan register cells at output pins are used to apply a test vector, while those at input pins capture test results.

EXTEST_PULSE and EXTEST_TRAIN

The EXTEST_PULSE and EXTREST_TRAIN instructions provide an edge-detecting test mechanism for AC-coupled signals. The EXTEST_PULSE instruction provides a pulse of data onto an AC signal whereby the pulse width is controlled by the time spent in the Run-Test/Idle TAP Controller state. This produces a single “wide” pulse with a variable and controllable period. The EXTEST_TRAIN instruction provides a pulse train, the edges of which are generated by each falling edge of TCK while in the Run-Test/Idle TAP Controller state. DC pins behave according to the IEEE Std 1149.1 EXTEST instruction when either the EXTEST_PULSE or EXTEST_TRAIN instructions are selected.

HIGH-Z

The High-Z instruction places all HMC outputs into a High-Z state. Additionally it causes the bypass register to be connected between the TDI and TDO pins.

CLAMP

When the CLAMP instruction is loaded into the instruction register, the data driven by the output pins is determined from the values held in the boundary-scan register.

SAMPLE/PRELOAD

When the SAMPLE/PRELOAD instruction is loaded into the instruction register and the TAP controller is in the Capture-DR state, a snapshot can be taken of the states of the component's input and output signals without interfering with the normal operation of the assembled board. The snapshot is taken on the rising edge of TCK and is captured in the Boundary-Scan register. The data can then be viewed by shifting through the components TDO output.

IDCODE

The IDCODE instruction causes loading of a vendor-specific, 32-bit code to be loaded into the identification register. It also places the identification register between the TDI and TDO pins and enables shifting the IDCODE out of the device when the TAP controller enters the Shift-DR state. The IDCODE instruction is loaded into the instruction register upon power-up or whenever the TAP controller is placed in the Test-Logic-Reset state.

BYPASS

When the BYPASS instruction is loaded in the instruction register and the TAP is placed in a Shift-DR state, the bypass register is placed between TDI and TDO.

CFG_RDA

Selects the CADATA register so configuration and status register address and read-data can be serially shifted between TDI and TDO. This instruction is not defined by the IEEE 1149.1 standard.

CFG_WRA

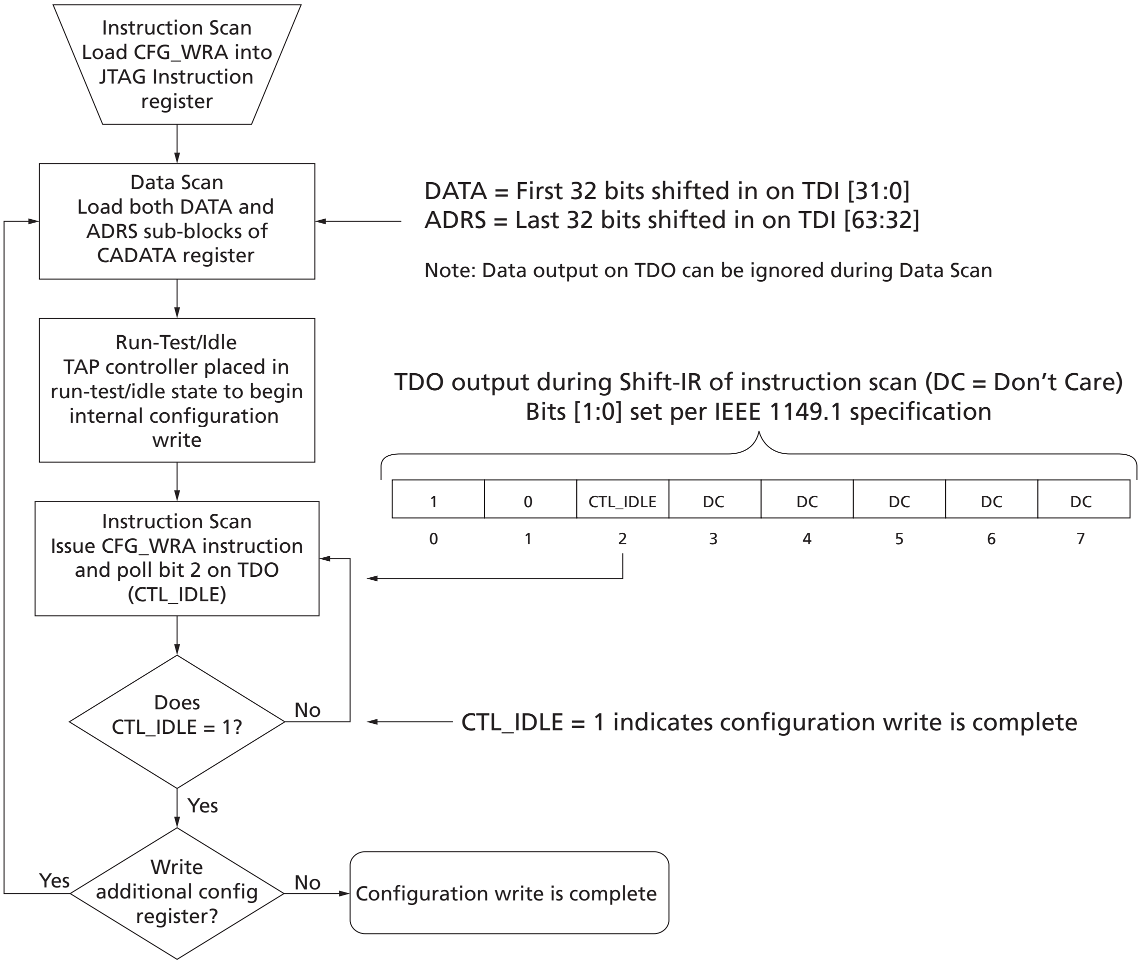
Selects the CADATA register so configuration and status register address and write-data can be serially shifted between TDI and TDO. This instruction is not defined by the IEEE 1149.1 standard.

Reserved for Future Use

The remaining instructions are not implemented but are reserved for future use. Do not use these instructions.

4.3.7 JTAG Configuration Register Write and Status Register Read

Figure 54: JTAG Configuration Register Write Flow Chart


Instruction Scan
Load CFG_WRA into JTAG Instruction register
Data Scan
Load both DATA and ADRS sub-blocks of CADATA register
Run-Test/Idle
TAP controller placed in run-test/idle state to begin internal configuration write
Instruction Scan
Issue CFG_WRA instruction and poll bit 2 on TDO (CTL_IDLE)
Does CTL_IDLE = 1?
No
Yes
Write additional config register?
Yes
No
Configuration write is complete

DATA = First 32 bits shifted in on TDI [31:0]
ADRS = Last 32 bits shifted in on TDI [63:32]

Note: Data output on TDO can be ignored during Data Scan

TDO output during Shift-IR of instruction scan (DC = Don’t Care)
Bits [1:0] set per IEEE 1149.1 specification

1	0	CTL_IDLE	DC	DC	DC	DC	DC
0	1	2	3	4	5	6	7

CTL_IDLE = 1 indicates configuration write is complete

Figure 55: JTAG Status Register Read Flow Chart

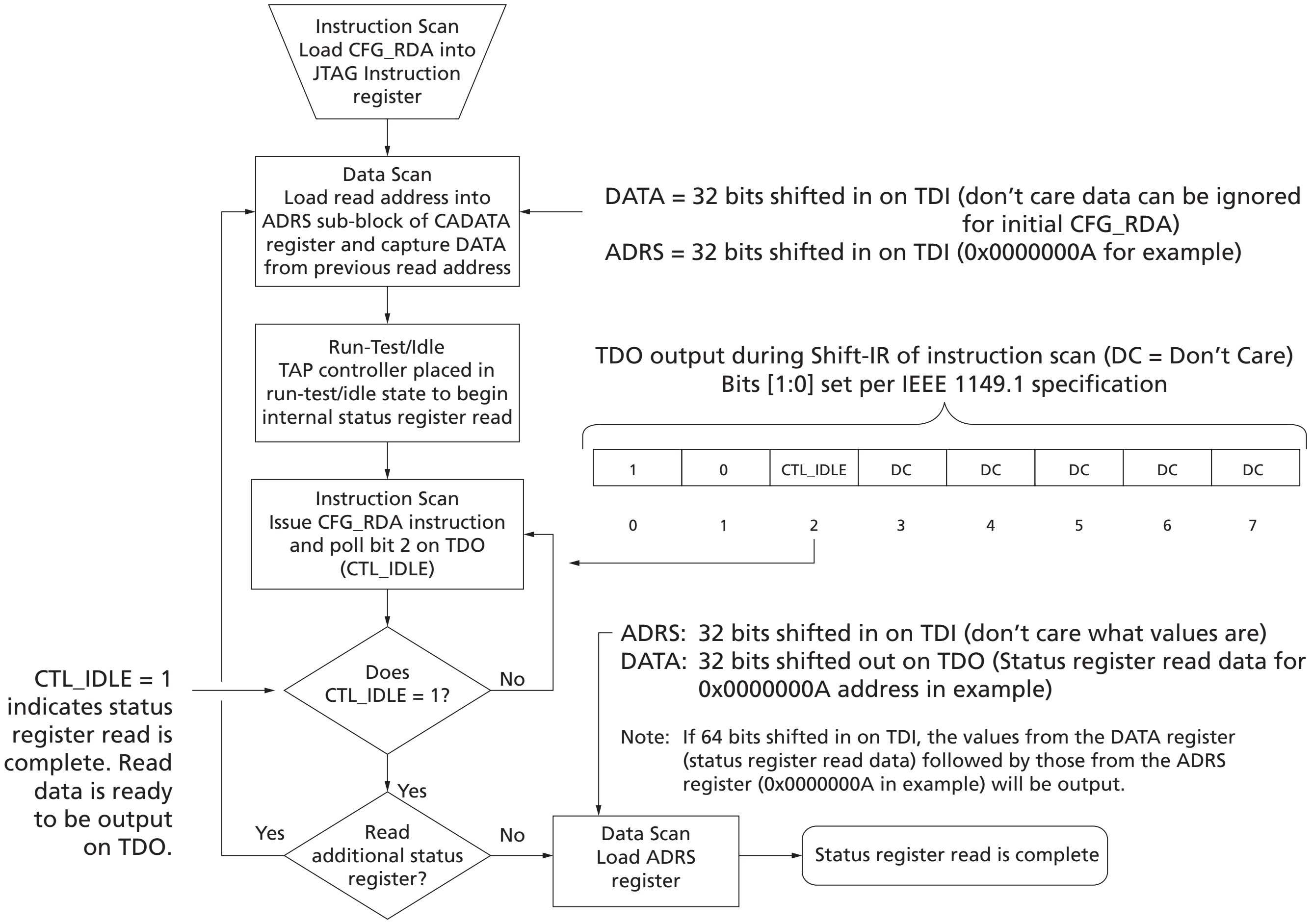
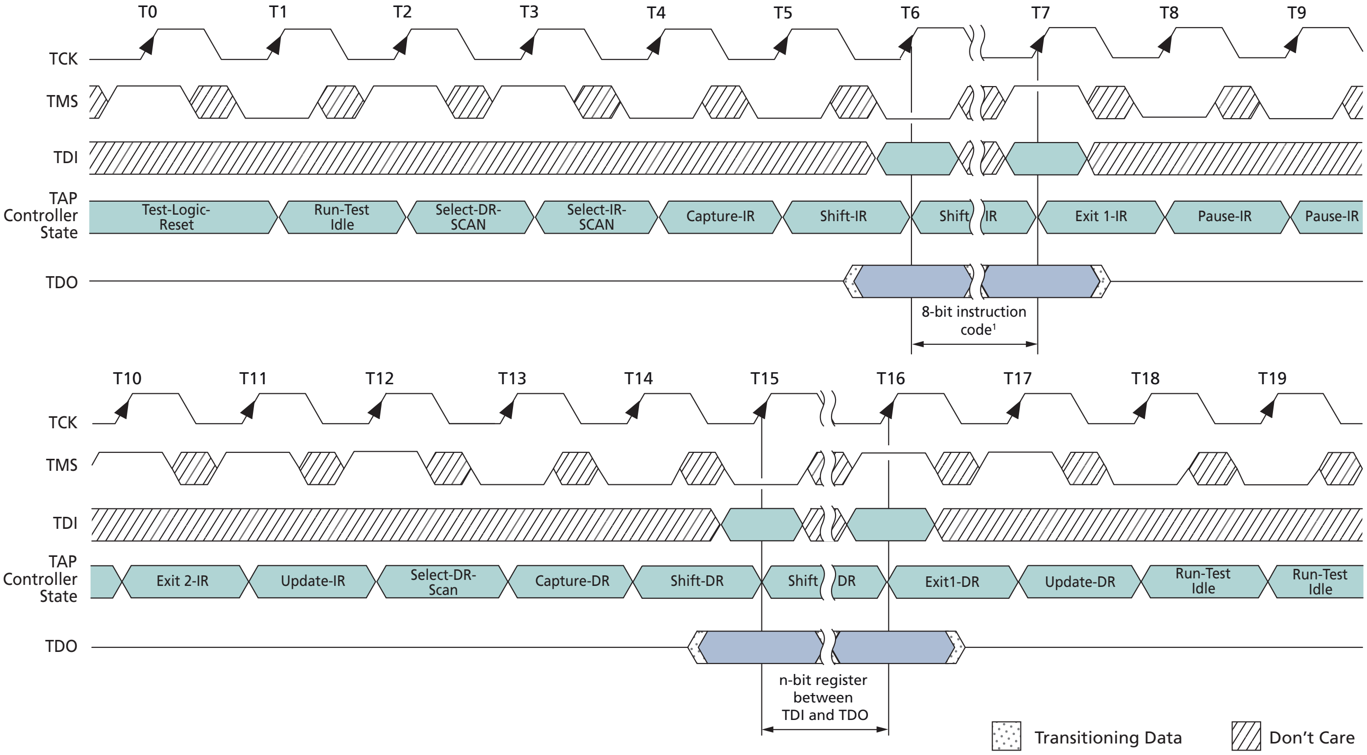
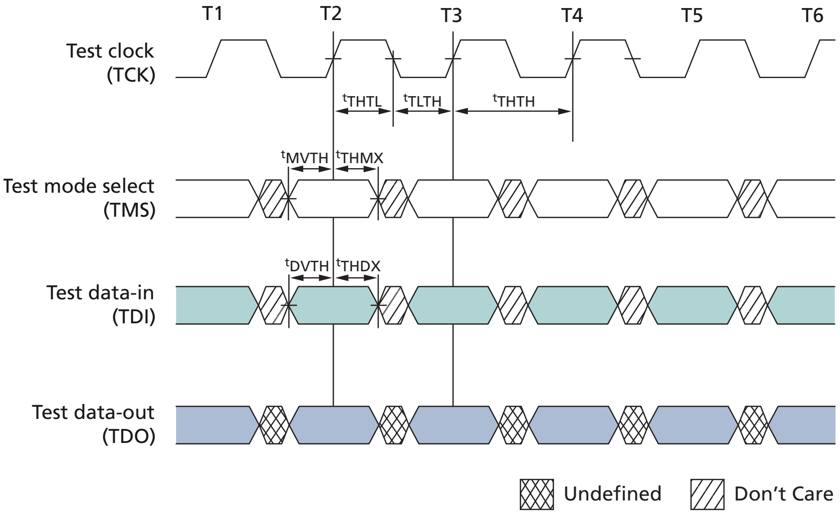

Instruction Scan
Load CFG_RDA into JTAG Instruction register
Data Scan
Load read address into ADRS sub-block of CADATA register and capture DATA from previous read address
DATA = 32 bits shifted in on TDI (don't care data can be ignored for initial CFG_RDA)
ADRS = 32 bits shifted in on TDI (0x0000000A for example)
Run-Test/Idle
TAP controller placed in run-test/idle state to begin internal status register read
TDO output during Shift-IR of instruction scan (DC = Don't Care)
Bits [1:0] set per IEEE 1149.1 specification
1
0
CTL_IDLE
DC
DC
DC
DC
DC
0
1
2
3
4
5
6
7
Instruction Scan
Issue CFG_RDA instruction and poll bit 2 on TDO (CTL_IDLE)
CTL_IDLE = 1 indicates status register read is complete. Read data is ready to be output on TDO.
Does CTL_IDLE = 1?
No
Yes
ADRS: 32 bits shifted in on TDI (don't care what values are)
DATA: 32 bits shifted out on TDO (Status register read data for 0x0000000A address in example)
Note: If 64 bits shifted in on TDI, the values from the DATA register (status register read data) followed by those from the ADRS register (0x0000000A in example) will be output.
Yes
Read additional status register?
No
Data Scan
Load ADRS register
Status register read is complete

Figure 56: JTAG Operation – Loading Instruction Code and Shifting Out Data



Note: 1. The LSB bits[1:0] will be 10 as shown in Figure 54 (page 95) and Figure 55 (page 96).

Figure 57: TAP Timing[1]



Note: 1. The timing specifications shown in this figure are listed in the HMC Gen2 data sheet.

Table 18: Identification Register Definitions

Instruction Field	All Devices	Description
Revision number (31:28)	0x0 or 0x2	0x0 indicates Revision 1 HMC device, 0x2 indicates revision 2 HMC device
Device ID (27:12)	0x0008	Identification of HMC
Micron JEDEC ID code (11:1)	00000101100	Enables unique identification of HMC vendors
ID register presence indicator (0)	1	Indicates the presence of an ID register

Table 19: Scan Register Sizes

Register Name	Bit Size
Instruction	8
Bypass	1
Device ID	32
Boundary scan	Dependent on number of links
CADATA	64

Table 20: Instruction Codes

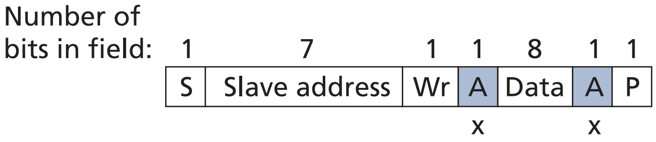
Instruction	Code	Active Register	Description
SAMPLE/PRELOAD	0x01	Boundary scan	1149.1 SAMPLE/PRELOAD instruction
IDCODE	0x02	Device ID	1149.1 IDCODE instruction
CLAMP	0x04	BYPASS	1149.1 CLAMP instruction
HIGHZ	0x08	BYPASS	1149.1 HIGHZ instruction
EXTEST	0x09	Boundary scan	1149.1 EXTEST instruction
EXTEST_PULSE	0x0A	Boundary scan	1149.6 EXTEST_PULSE instruction
EXTEST_TRAIN	0x0B	Boundary scan	1149.6 EXTEST_TRAIN instruction
CFG_RDA	0x2C	CADATA	Allows configuration and status register to be read
CFG_WRA	0x2D	CADATA	Allows configuration and status register to be written
BYPASS	0xFF	BYPASS	1149.1 BYPASS instruction

4.4 I^{2}C Interface

The I^{2}C bus is provided as a sideband interface to program, monitor, and access various modes within the HMC. The I^{2}C bus complies with the UM-10204 I^{2}C bus specification and includes the following attributes:

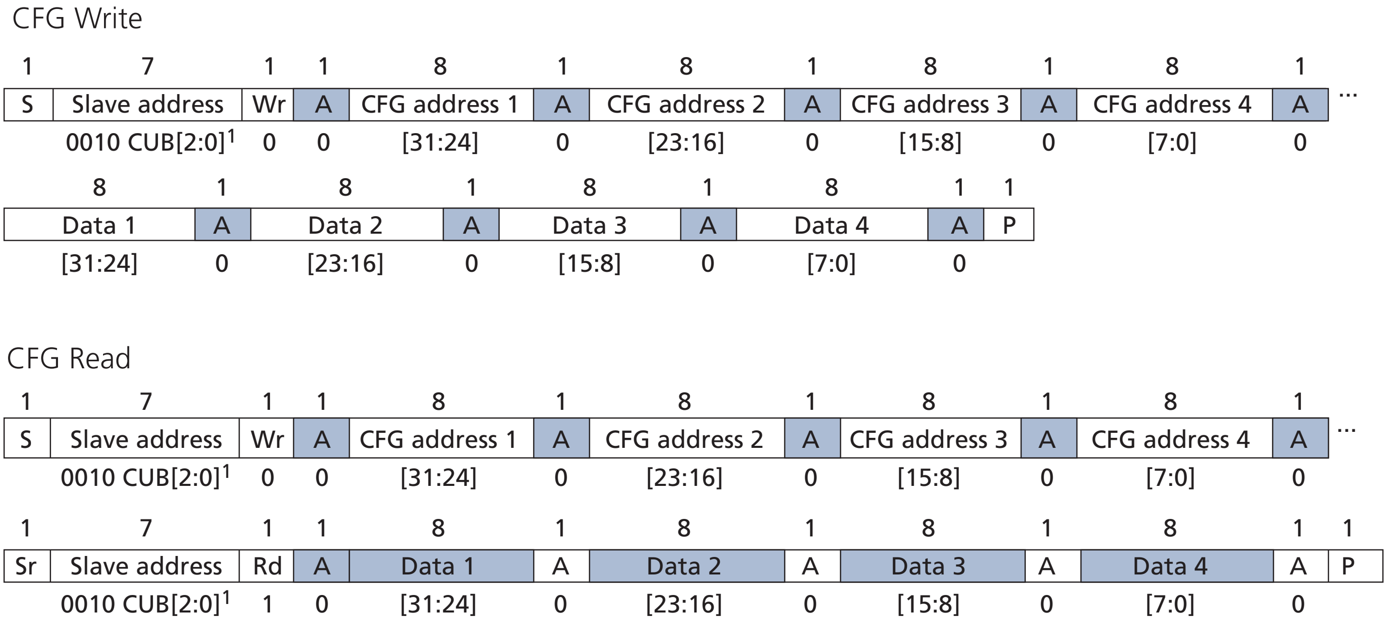
- Operates with V_{DDK} supply (1.5V nominal) with either an open-drain or open-collector output stage. See I^{2}C Parameters table for full specifications.
- Uses 7-bit slave addressing.
- Offers FTDI I^{2}C compatibility.
- Complies with the standard-mode (100 Kb/s) and fast-mode (400 Kb/s) electrical and timing parameters found in the UM-10204 I^{2}C bus specification.

Figure 58: I^{2}C Packet Protocol Key



Symbol	Description
S	Start condition
Sr	Repeated start condition
Rd	Read (bit value of 1)
Wr	Write (bit value of 0)
x	Value under field indicating required value
A	Acknowledge (Ack is 0, Nack is 1)
P	Stop condition
□	Master-to-slave
▨	Slave-to-master
...	Continuation of protocol

Figure 59: Configuration Register WRITE and READ Commands



CFG Write

1	7	1	1	8	1	8	1	8	1	8	1	
S	Slave address	Wr	A	CFG address 1	A	CFG address 2	A	CFG address 3	A	CFG address 4	A	...
	0010 CUB[2:0][1]	0	0	[31:24]	0	[23:16]	0	[15:8]	0	[7:0]	0	

8	1	8	1	8	1	8	1	1
Data 1	A	Data 2	A	Data 3	A	Data 4	A	P
[31:24]	0	[23:16]	0	[15:8]	0	[7:0]	0	

CFG Read

1	7	1	1	8	1	8	1	8	1	8	1	
S	Slave address	Wr	A	CFG address 1	A	CFG address 2	A	CFG address 3	A	CFG address 4	A	...
	0010 CUB[2:0][1]	0	0	[31:24]	0	[23:16]	0	[15:8]	0	[7:0]	0	

1	7	1	1	8	1	8	1	8	1	8	1	1
Sr	Slave address	Rd	A	Data 1	A	Data 2	A	Data 3	A	Data 4	A	P
	0010 CUB[2:0][1]	1	0	[31:24]	0	[23:16]	0	[15:8]	0	[7:0]	0	

4.5 Register Definitions

This section provides an overview of the configuration and status registers supported by the HMC Gen2 device. Refer to the full HMC Gen2 data sheet for the complete specification.

Note: Contact Micron for more details about the functionality supported with each hardware and firmware revision.

4.5.1 Register Types

All registers are accessible through in-band mode write and mode read requests or via sideband (JTAG/I^{2}C). Two of the distinct register types described throughout this document are direct registers, accessed directly using unique register addresses; and indirect registers, which are accessed through the External Request Interface and are referred to as ERI commands. There are two time periods for accessing these registers: configuration and runtime. Configuration registers and ERIs correlate to Step 5 of the initialization routine as defined in the HMC data sheet, configuring the device prior to and during the INIT Continue ERI command. Runtime registers and ERIs can be used following the clearing of the start bit after INIT Continue has completed execution. Register types include the four combinations of these two categories (direct/indirect and configuration/runtime), as well as two other types, device information status and debug. These combinations are detailed in Table 21 (page 101).

Table 21: Register Types and Access Methods

Register Type	Accessible Time	Description
Device information status	Any (before or after initialization)	Used to determine device information and status.
Direct configuration	Initialization	Configuration registers used to set up the device prior to INIT Continue ERI command initiation.
Indirect configuration	Initialization	ERI commands used to set up the device prior to INIT Continue ERI command initiation.
Direct runtime	Runtime	Status and control registers that are valid following INIT Continue ERI command completion.
Indirect runtime	Runtime	Status and control ERI commands that are valid following INIT Continue ERI command completion.
Debug	Runtime	Registers or register fields that are only to be used for debug purposes and may have an impact on normal operation. These are valid following INIT Continue ERI command completion.

4.5.2 Sideband Direct Register Addressing Methods

HMC registers can be accessed in-band using the link interface with MODE WRITE and MODE READ commands or using sideband (I^{2}C or JTAG) accesses, which are explained in more detail here. The details of in-band addressing methods are contained in the Hybrid Memory Cube data sheet. With respect to sideband accesses, there are two methods that can be used to address each direct register in the HMC. The simplest method for addressing direct registers is to read or write the entire 32-bit register data field using the register addresses shown in the tables in this document. This method can potentially require a read-modify-write in order to properly set some data fields or it may require masking by the host or sideband processor when reading a register in order to isolate the desired register data fields. The start/size addressing method is a more advanced and precise method for accessing HMC status and control register fields. Using this method, a single register field can be read or written and all other fields can be masked.

4.5.2.1 Start/Size Addressing Method

This addressing method allows specific fields within a register to be accessed directly without having to read or write other fields in that register. Such a method is useful when modifying or reading a single field per access is desired.

The following is the format for the address to be used with the start/size addressing method through sideband accesses to configuration and status registers in the HMC.

Table 22: Start/Size Addressing Bit Fields

Field	Bits	Number of Bits	Description
Start	31:27	5	**Start field:** Encoded to provide a value from 0 to 31 that points to the starting bit position within the 32 bits of data, 0 being the LSB position and 31 being the most significant bit (MSB) position.
Size	26:22	5	**Size field:** Indicates the number of contiguous bits to read or write (from 1 to 32), and is encoded as follows: 0x0 = 32 bits, 0x1–0x1F = 1–31 bits.
Register address	21:0	22	**Register address field:** References a specific register as defined in this document.

Figure 60 illustrates an example of using the full register write method to set the link response open loop mode for link 0 to a value of 1. Figure 61 illustrates the same command using the start/size addressing method. The full register write method requires that the register be read in full, modified (masking values not being changed) and written back to the link configuration register. The start/size addressing method only requires a single write to perform the same operation by automatically masking out values which are not being changed.

Figure 60: Full Register Write Method Example

```
openloop = 1 # Assign variable openloop value '1'
temp = (i2crd(0x20,0x00240000)) # Read 32 bit Link 0 Configuration Register
# Mask temp to assign openloop ('1') to only bit 2 of temp, and store in variable temp2
temp2 = ((temp&&0xFFFFFFFB||(openloop<<2))
# Write value stored in temp2 back to Link 0 Link Configuration Register
i2cwr(0x20,0x00240000,temp2))
```

Figure 61: Start/Size Register Write Method Example

```
openloop = 1 # Assign variable openloop value '1'
# Assign openloop ('1')as follows: start bit 2, size 1 address 0x240000
i2cwr(0x20,0x10640000,openloop)
```

4.5.3 Device Information Status Registers

Status registers discussed in this section may be accessed with the I^{2}C or JTAG buses at any point in time after ^{t}INIT. Access with in-band mode read requests must occur after the transaction layer has been initialized.

Table 23: Features

Register address: 0x2C0003

Name	Start Bit	Size	Description
Cube size	0	4	0x0: 2GB 0x1-0xF: Reserved
Number of vaults	4	4	0x0: 16 vaults 0x1-0xF: Reserved
Number of banks per vault	8	4	0x0: 8 banks 0x1-0xF: Reserved
PHY	12	4	0x0: HMC-15G-SR 0x1-0xF: Reserved
Firmware feature set	16	16	Indicates revision of firmware programmed into the device. [23:16]: Indicates minor revision [27:24]: Indicates major revision [31:28]: Indicates patch revision

Table 24: Revisions and Vendor ID

Register address: 0x2C0004

Name	Start Bit	Size	Description
Vendor ID	0	8	Vendor ID Micron = 0x2C
Product revision	8	8	Product revision
Protocol revision	16	8	Protocol revision 0x01 = HMCC Procotol revision 1.0, 0x11 = HMCC Protocol revision 1.1
PHY revision	24	8	PHY revision 0x01

Table 25: Cube Serial Number 1

Register address: 0x2C0001

Name	Start Bit	Size	Description
Serial number 1	0	32	Unique cube serial number (1 of 2)

Table 26: Cube Serial Number 2

Register address: 0x2C0002

Name	Start Bit	Size	Description
Serial number 2	0	32	Unique cube serial number (2 of 2)

4.5.4 Configuration Registers and ERIs

All of the register tables in this document include a column that specifies "type":
RW = Register can be read from and written to.
RO = Register can only be read from.
RWS = Register self clears after being written to.

4.5.4.1 Direct Configuration Registers

There are four types of direct configuration register fields:

1. **INIT:** Must only be set during the configuration register load period of the initialization routine. Recommended values are included where appropriate.
2. **Debug:** Must match reset value for normal operation. Contact your Micron representative for information on how to use these registers for debug purposes.
3. **Runtime:** These register fields can be changed during the configuration register load period of the initialization routine or during normal operation (after transaction layer initialization).
4. **Reserved:** Must be set to 0.

Table 27: Global Configuration Registers

Register address: 0x280000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Settings	Notes
Reserved	0	4	–	–	Reserved	Reserved	0x0	
Stop on fatal error	4	1	RW	0	When set to 1, a fatal error will cause the device to stop returning response packets back to the host (except for error response packets, when possible) and will stop transaction execution at the vault controller	Debug/INIT	0	
Reserved	5	1	–	0	Reserved	Reserved	0	
Warm reset	6	1	RWS	0	Writing a 1 initiates a warm reset sequence; this bit auto clears	Runtime	0 unless warm reset required	1
Reserved	7	25	–	–	Reserved	Reserved	0x000000	

Note: 1. If a warm reset is required during normal operation, only the warm reset bit should be modified. Therefore, Micron recommends using the start/size addressing method by writing 0x1 to address 0x30680000.

Table 28: Request Identification Register

Register address: Link 0 = 0x000000; Link 1 = 0x010000; Link 2 = 0x020000; Link 3 = 0x030000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Settings
Source link ID (SLID)	0	3	RW	Actual cube link number	Sets the HMC link ID, which is used for internal routing of packets and will also be used in the SLID field in the response packet from the HMC, as shown in the specification. Legal values are 0, 1, 2, or 3 in current devices.	Debug	This field will be set automatically by the HMC upon boot and must not be modified.
Cube ID (CUB)	3	3	RW	Value of CUB pins	Cube number, which is used to match a cube with the CUB field in a request packet header that is directed to the appropriate cube.	INIT	By default, this register field is set by the HMC with the value read from the CUB pins. It can be changed by the host in order to align with the desired cube identifier being sent by the host in the CUB field of the request packet header.
Reserved	6	26	–	–	Reserved	Reserved	0x0000000

Table 29: Link Configuration Register

Register address: Link 0 = 0x240000; Link 1 = 0x250000; Link 2 = 0x260000; Link 3 = 0x270000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Settings	Notes
Link mode	0	2	RW	0x1	0x0: Link is not used 0x1: Link is a host link and a source link (typical use case) 0x2: RESERVED 0x3: Link is a pass-through link (used for chained configurations)	INIT	System configuration dependent	
Link response open loop mode	2	1	RW	0	0x0: Response open loop mode is off 0x1: Response open loop mode is on	INIT	Dependent upon how host consumes tokens	
Link packet sequence detection	3	1	RW	1	0x0: Packet sequence detection is off 0x1: Packet sequence detection is on	Debug	1	
Link CRC detection	4	1	RW	1	0x0: Link CRC error detection is off 0x1: Link CRC error detection is on	Debug	1	

Table 29: Link Configuration Register (Continued)

Register address: Link 0 = 0x240000; Link 1 = 0x250000; Link 2 = 0x260000; Link 3 = 0x270000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Settings	Notes
Link duplicate length detection	5	1	RW	1	0x0: DLN detection is off 0x1: DLN detection is on	Debug	1	
Packet input enable	6	1	RW	1	0x0: Decode and parsing of incoming packets is disabled 0x1: Decode and parsing of incoming packets is enabled	Debug	1	
Packet output enable	7	1	RW	1	0x0: Transmission of outgoing packets is disabled 0x1: Transmission of outgoing packets is enabled	Debug	1	1
Inhibit link down mode	8	1	RW	0	When set to 1, the HSS PLLs will remain on regardless of the state of LxRXPS signals (applies only to links not reset in power-down mode)	INIT	Host dependent, assess trade-off in low-power mode exit timing vs. power savings	
Link descramble enable	9	1	RW	1	0x0: Receiver descramblers are disabled 0x1: Receiver descramblers are enabled	Debug	1	
Link scramble enable	10	1	RW	1	0x0: Transmit scramblers are disabled 0x1: Transmit scramblers are enabled	Debug	1	
Error response packet	11	1	RW	1	When set to 1, the HMC will send error response packets on this link	INIT	Host dependent; error responses may be sent on any combination of links	
Reserved	12	20	–	–	Reserved	Reserved	0x00000	

Note: 1. Unless set by the host, this field will reset to 0x0 upon cold reset and be set to 0x1 during execution of the INIT Continue ERI.

Table 30: Link Run Length Limit Register

Register address: Link 0 = 0x240003; Link 1 = 0x250003; Link 2 = 0x260003; Link 3 = 0x270003

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting
Reserved	0	16	–	–	Reserved	Reserved	0x0000

Table 30: Link Run Length Limit Register (Continued)

Register address: Link 0 = 0x240003; Link 1 = 0x250003; Link 2 = 0x260003; Link 3 = 0x270003

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting
Transmit run length limit	16	8	RW	0x00	Sets the run length limit allowed on each lane (that is, the maximum number of zeros or ones that can be sent consecutively before a limiting circuit inserts a transition in the data stream to assure proper clock/data recovery at the far end receiver). A value of 50 (0x32) or more is required in this register to ensure proper run limiting operation. The default value is 0x00 (no limiting).	INIT	Dependent on host RX requirements; if host RX is DC-coupled, set to 0x00
Reserved	24	8	–	–	Reserved	Reserved	0x00

Table 31: Link Retry Register

Register address: Link 0 = 0x0C0000; Link 1 = 0x0D0000; Link 2 = 0x0E0000; Link 3 = 0x0F0000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting	Notes
Retry enable	0	1	RW	1	0x0: Retry is disabled 0x1: Retry is enabled	Debug	1	
Retry limit	1	3	RW	0x7	Controls the number of consecutive retry attempts for which there is no resulting progress (that is, local error abort never clears or received RRP does not advance despite unretired packets in the local retry buffer). When this limit has been met, an error response packet is sent with link error in the ERRSTAT field to notify the host that the retry attempt limit has been met and FERR_N pin will assert. If retry limit = 0, there is no limit on the number of retries that can occur with no resulting progress.	INIT	0x7	

Table 31: Link Retry Register (Continued)

Register address: Link 0 = 0x0C0000; Link 1 = 0x0D0000; Link 2 = 0x0E0000; Link 3 = 0x0F0000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting	Notes
Retry timeout period	4	3	RW	0x5	0x0: 154ns 0x1: 205ns 0x2: 307ns 0x3: 410ns 0x4: 614ns 0x5: 820ns 0x6: 1229ns 0x7: 1637ns	INIT	Dependent on host retry pointer return time	
Error abort mode	7	1	RO	0	Indicates that link has been in error abort mode. This bit is auto-cleared when the specified number of consecutive IRTRY packets have been received in order to clear error abort mode.	Debug	N/A	
INIT retry packet transmit number	8	6	RW	0x06	0x2–0x3F: The number of IRTRY packets transmitted from link master during LinkRetry_Init is approximately four times the number specified in this field	Debug	0x06 (This will send 22–28 IRTRYs during LinkRetry_Init)	1
Reserved	14	2	–	–	Reserved	Reserved	0x0	
INIT retry packet receive number	16	6	RW	0x10	The number of consecutive IRTRY packets that the link slave will detect before it clears the error abort mode.	INIT	0x10 (16 IRTRY packets expected)	1
Reserved	22	2	–	–	Reserved	Reserved	0x0	
Link retry state	24	4	RO	0x1	Indicates status of the link retry state machine as follows: 0x1: Idle 0x2: Waiting for packet in progress to finish 0x4: Sending IRTRY 0x8: Sending retry buffer	Debug	N/A	
Reserved	28	4	–	–	Reserved	Reserved	0x0	

Note: 1. The number of IRTRYs transmitted and received by the host should be equivalent to those loaded into the INIT Retry Packet Transmit Number and INIT Retry Packet Receive Number fields, respectively. The host should transmit at least 22 IRTRYs during LinkRetry_INIT and should expect 16 IRTRY packets before clearing error abort mode.

Table 32: Input Buffer Token Count Register

Register address: Link 0 = 0x040000; Link 1 = 0x050000; Link 2 = 0x060000; Link 3 = 0x070000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting	Notes
Link input buffer max token count	0	8	RW	Recommended maximum token count for normal operation	Value represents the buffer space available, as measured in flits, in the link input buffer when it is empty	INIT	System-dependent based on bandwidth and latency requirements	1
Reserved	8	24	–	–	Reserved	Reserved	0x000000	

Note: 1. The maximum token count allowed for normal operation is 219. The default value for this register will be the maximum value available for normal operation and can be checked by reading this value prior to writing it. The HMC data sheet Link Flow Control During Retry section contains discussion as to how the total sum of tokens are used for normal use as dictated by the protocol.

Table 33: Address Configuration Register

Register address: 0x2C0000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting
Address mapping mode	0	4	RW	0x2	0x0: 32-byte block size (MAX). Low-interleave address mapping with the byte address set as the lowest five bits of the ADRS field in the request header, along with the default vault and bank mapping. 0x1: 64-byte block size (MAX). Low-interleave address mapping with the byte address set as the lowest six bits of the ADRS field in the request header, along with the default vault and bank mapping. 0x2: 128-byte block size (MAX). Low-interleave address mapping with the byte address set as the lowest seven bits of the ADRS field in the request header, along with the default vault and bank mapping. 0x3-0x7: Reserved. 0x8: User-defined 32-byte MAX block size. Byte address is defined as the lowest five bits of the ADRS field in the request header, along with user-defined vault and bank mapping. The user-defined vault and bank address start bits and size fields must be nonzero. 0x9: User-defined 64-byte MAX block size. Byte address is defined as the lowest six bits of the ADRS field in the request header, along with user-defined vault and bank mapping. The user-defined vault and bank address start bits and size fields must be nonzero. 0xA: User-defined 128-byte MAX block size. Byte address is defined as the lowest seven bits of the ADRS field in the request header, along with user-defined vault and bank mapping. The user-defined vault and bank address start bits and size fields must be nonzero. 0xB-0xF: Reserved.	INIT	Host-dependent. Requests contained in completely random addresses do not benefit from any specific address mapping mode.

Table 33: Address Configuration Register (Continued)

Register address: 0x2C0000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting
User-defined vault address	4	5	RW	0x00	Encoded field that specifies corresponding bit of ADRS to be LSB of vault address.	INIT	Set to 0x00 if using low-interleave address mapping (0x0-0x2 in address mapping mode field). Must be nonzero if user-defined addressing mode is used.
User-defined bank address	9	5	RW	0x00	Encoded field that specifies corresponding bit of ADRS to be LSB of bank address.	INIT	Set to 0x00 if user-defined addressing mode is NOT used. Must be nonzero if user-defined addressing is mode used.
Reserved	14	18	–	–	Reserved	Reserved	0

Table 34: Vault Control Register

Register address: 0x108000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting	Notes
DRAM initialization mode	0	2	RW	0x0	0x0: DRAM is initialized to all 0s with correct ECC 0x1: Reserved 0x2: Initialize DRAM to user pattern 0x3: Reserved	INIT	0x0, unless there is a need for a unique pattern.	1
Reserved	2	1	RO	0	Reserved	Reserved	0	2
Demand scrubbing	3	1	RW	1	Set to 0 to disable a scrub write to DRAM of corrected data when an SBE is detected on a transaction read request.	INIT	1	4

Table 34: Vault Control Register (Continued)

Register address: 0x108000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting	Notes
Patrol scrubbing	4	1	RW	1	Set to 0 to disable a scrub write to DRAM of corrected data when an SBE is detected on a patrol read request.	INIT	1: Writing the corrected data back into the memory array reduces the risk of the SBE turning into an uncorrectable MUE on subsequent accesses.	4
Packet CRC detection	5	1	RW	1	Set to 0 to disable the CRC check performed on incoming packets to the vault controllers. This allows the vault controller to execute request packets that have CRC errors.	Debug	1	
Command/ Address retry count	6	3	RW	0x1	0x0: DRAM Command/Address retry disabled; 0x1: DRAM Command/ Address retry enabled	Debug	0x1	3
Data ECC correction disable	9	1	RW	0x0	Set to 1 to disable correction of an SBE if detected by the vault controller during read requests. Data from the DRAM will be returned unmodified in the response packet.	Debug	0	
MUE repair enable	10	1	RW	0x1	Enable to invoke a repair operation when an MUE is detected during either demand requests or patrol requests.	INIT	1	4
SBE repair enable	11	1	RW	0x1	Enable to invoke a repair operation when hard or frequent SBEs are detected during either demand requests or patrol requests. SBE repair has two sub types of repair, dynamic and permanent. Dynamic occurs on the fly while the system is still running. Permanent requires the device to do a cold reset to complete. Dynamic and/or permanent repair of hard SBEs reduces the risk of the SBE turning into an MUE on subsequent accesses.	INIT	1	4

Table 34: Vault Control Register (Continued)

Register address: 0x108000

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting	Notes
Enable SBE report	12	1	RW	0x0	Set to 0x1 (enable) to report any SBEs that occurred on a read request. These will be reported within the ERRSTAT field of the read response.	INIT	0	
Reserved	13	19	–	–	Reserved	Reserved	0x00000	2

Notes:
1. The DRAM Initialization Mode register field is set at initialization but will be used for DRAM initialization upon power up and cold reset as well as after DRAM BIST where applicable. See the DRAM BIST section for further details as to which modes will use this register field.
2. Reserved bit values may be nonzero during runtime.
3. Only the values of 0x0 and 0x1 are supported.
4. Refer to Table 12 (page 31) to determine the impact of these bits.

Table 35: Disable NVM Write and Bootstrap Status

Register address: 0x280002

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting
Reserved	0	1	–	–	Reserved	Reserved	0
Cube ID	1	3	RO	N/A	These bits reflect the values of the CUB[2:0] pins captured when P_RST_N is de-asserted.	RO	N/A
Reference clock frequency	4	2	RO	N/A	These bits reflect the values of the REFCLK_BOOT[1:0] pins captured when P_RST_N is de-asserted. 00 = 125 MHz 01 = 156.25 MHz 10 = 166.67 MHz	RO	N/A
Reserved	6	2	–	–	Reserved	Reserved	0
Disable NVM write	8	1	RW	0x0	When bit is set to 1, all writing to NVM is disabled (includes logging and field repair).	INIT	0x0 Host should set to 0x1 during device margin testing to avoid potentially using field repair resources due to voltages below specification.

Table 35: Disable NVM Write and Bootstrap Status (Continued)

Register address: 0x280002

Name	Start Bit	Size	Type	Reset Value	Description	Register Type	Recommended/ Required Setting
Boot flag	9	23	RO	N/A	Contains boot information. HMC will populate with nonzero value.	Debug	N/A

4.5.4.2 Indirect Configuration Registers

This type of access uses the external request data registers and the external request command register to access more complex commands and registers via the external request interface. The external data and request registers are accessible through mode write and mode read requests or sideband (JTAG/I^{2}C). The following section outlines how to run these commands by using the ERI register set.

Figure 62 illustrates the steps required to read or write ERI configuration registers.

Figure 62: ERI Command Access Flow

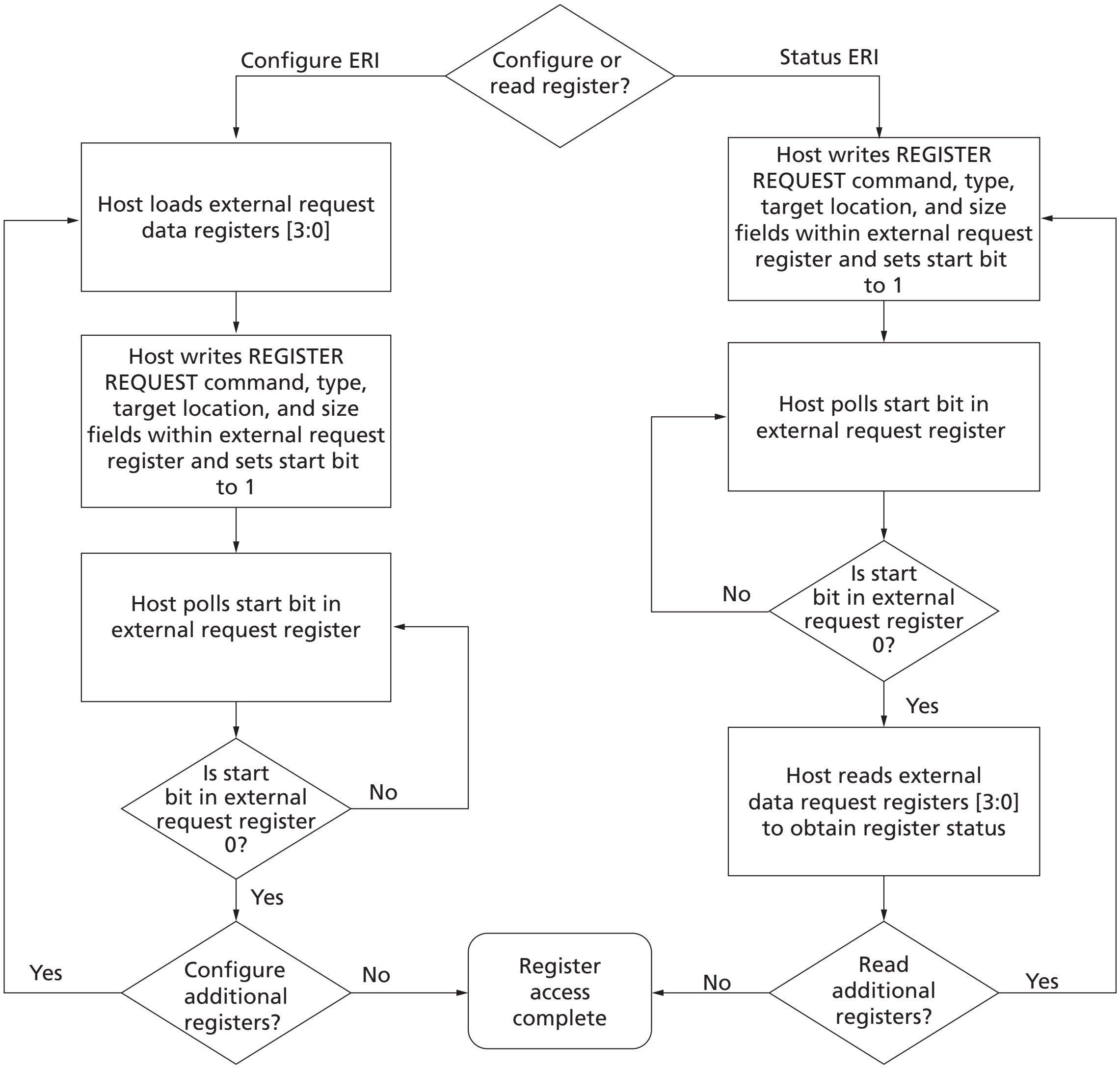

Configure or read register?
Configure ERI
Status ERI
Host loads external request data registers [3:0]
Host writes REGISTER REQUEST command, type, target location, and size fields within external request register and sets start bit to 1
Host polls start bit in external request register
Is start bit in external request register 0?
No
Yes
Configure additional registers?
Yes
No
Register access complete
Host writes REGISTER REQUEST command, type, target location, and size fields within external request register and sets start bit to 1
Host polls start bit in external request register
Is start bit in external request register 0?
No
Yes
Host reads external data request registers [3:0] to obtain register status
Read additional registers?
No
Yes

Table 36: External Request Data Register 0 (ERIDATA0)

Register address: 0x2B0000

Name	Start Bit	Size	Type	Reset Value	Description	Recommended Value
External data register 0	0	32	RW	N/A	Multipurpose register used to access configuration and status registers	ERI command dependent

Table 37: External Request Data Register 1 (ERIDATA1)

Register address: 0x2B0001

Name	Start Bit	Size	Type	Reset Value	Description	Recommended Value
External data register 1	0	32	RW	N/A	Multipurpose register used to access configuration and status registers	ERI command dependent

Table 38: External Request Data Register 2 (ERIDATA2)

Register address: 0x2B0002

Name	Start Bit	Size	Type	Reset Value	Description	Recommended Value
External data register 2	0	32	RW	N/A	Multipurpose register used to access configuration and status registers	ERI command dependent

Table 39: External Request Data Register 3 (ERIDATA3)

Register address: 0x2B0003

Name	Start Bit	Size	Type	Reset Value	Description	Recommended Value
External data register 3	0	32	RW	N/A	Multipurpose register used to access configuration and status registers	ERI command dependent

Table 40: External Request Register (ERIREQ)

Register address: 0x2B0004

Name	Start Bit	Size	Type	Reset Value	Description
Register request commands	0	8	RW	0x00	0x00: NO OPERATION 0x05: LINK CONFIGURATION 0x06: PHY CONFIGURATION 0x07: DRAM BIST WITH REPAIR 0x08: DRAM BIST WITHOUT REPAIR 0x09: MANUAL DRAM REPAIR 0x0A: TEMPERATURE MONITOR 0x0B: TEMPERATURE HISTORY 0x0C: REFRESH RATE ADJUSTMENT 0x0E: SIDEBAND DRAM ACCESS 0x0F: USER DRAM PATTERN DEFINITION 0x10: Chaining Topology Configuration 0x20: DRAM REPAIR HEALTH STATUS 0x21: NVM LOG READ START 0x22: NVM LOG READ NEXT 0x25: PRBS Verification 0x26: PATCH LOAD 0x27: PATCH EXECUTE 0xFF: INIT CONTINUE; command is executed by host after it has completed its register configuration, allowing internal HMC configuration to continue. All others: RESERVED
Type	8	8	RW	0x00	Reserved
Target location	16	6	RW	0x00	0x00–0x3E: Target location of command (including but not limited to links and vaults) 0x3F: Used to load global registers; also used to broadcast to all links
Size	22	4	RW	0x0	0x0: No data registers associated with request 0x1–0x4: Number of data registers associated with request 0x5–0x1F: Reserved
External request status	26	5	RO	0x00	0x00: Request was successful 0x01: Request caused an internal error 0x02: Request was invalid 0x03: Multi-cycle request complete The device sets this field when the last group of data registers for a status request is sent to the host; the field is also set after the last group of data registers expected from a configuration request 0x04–0x1F: ERI Specific

Table 40: External Request Register (ERIREQ) (Continued)

Register address: 0x2B0004

Name	Start Bit	Size	Type	Reset Value	Description
Start	31	1	RW	0	Host sets to 1 to inform the device there is a valid external request; HMC device sets to 0 after request is complete. Host polls bit and when it is 0, it can issue subsequent EXTERNAL REQUEST commands and read the external request status bits to determine the success of the pending EXTERNAL REQUEST command.

4.5.4.3 Early Configuration Time ERIs

Configuration ERI Sequences

The ERI registers in this section must only be issued during the configuration register load period of the initialization routine. Runtime reconfiguration is not supported. A cold reset must occur if configuration registers need to change after initialization. The only exception is the refresh rate adjustment ERI, which can be changed during initialization or runtime.

Table 41: Link Configuration ERI

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[1:0]: Link 0 speed	0x0: 10 Gb/s 0x1: 12.5 Gb/s 0x2: 15 Gb/s 0x3: Reserved	1, 2, 3, 4, 5
				[15:2]: Reserved	Must be 0x00	
				[17:16]: Link 0 link width	0x0: Full width TX and RX 0x1: Half-width TX and RX 0x2: Full-width TX/half-width RX 0x3: Half-width TX/full-width RX	
				[25:18]: Reserved	Must be 0x00	
				[26]: Link 0 Serial Loopback (TX-to-RX)	0: Disabled (required in normal operation) 1: Enabled	
				[29:27]: Link 0 PRBS patterns	0x0: PRBS7+ 0x1: PRBS7- 0x2: PRBS15+ 0x3: PRBS15- 0x4: PRBS23+ 0x5: PRBS23- 0x6: PRBS31+ 0x7: PRBS31-	
				[30]: Link 0 TX pattern generation	0: Disabled (required in normal operation or when enabling parallel RX-to-TX loopback) 1: Enabled	
				[31]: Link 0 parallel loopback (RX-to-TX)	0: Disabled (required in normal operation) 1: Enabled	
2–4	MODE WRITE REQUEST	ERIDATA[3:1]	0x2B000[3:1]	Same as Step 1 for remaining links: Link 1 uses ERIDATA1 Link 2 uses ERIDATA2 Link 3 uses ERIDATA3	Same as Step 1 for remaining links	
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x05: Link configuration	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	1: Inform HMC of valid request	

Table 41: Link Configuration ERI (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
6	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x05: Link configuration	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request	
				[31]: Start	Start bit polling sequence If [31] = 1, pause for 10µs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step	

Notes:
1. PRBS patterns are only valid when PRBS pattern generation is enabled.
2. Links are not trained when either TX pattern generation or parallel loopback (RX to TX) is enabled.
3. When using half-width link configurations, the lower half of the link is the half that is used.
4. Example PRBS algorithms are: PRBS7 = 1 + X^6 + X^7; PRBS15 = 1 + X^{14} + X^{15};PRBS23 = 1 + X^{18} + X^{23}; PRBS31 = 1 + X^{28} + X^{31}.
5. Link parallel loopback (RX to TX) functionality requires that Link TX pattern generation be disabled.

Table 42: PHY Configuration ERI

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[4:0]: Link 0 TX K2 coefficient	System dependent; Valid range: 0x0 to 0x15	1
				[15:5]: Reserved	Must be 0x000	
				[19:16]: Link 0 TX K0 coeficient	System dependent; Valid range: 0x0 to 0x0D	1
				[29:20]: Reserved	Must be 0x000	
				30: AC-coupling capacitors used on board external to HMC Receivers	0: DC-coupled RX; 1: AC-coupled RX	
				31: Reserved	Must be 0	
2–4	MODE WRITE REQUEST	ERIDATA[3:1]	0x2B000[3:1]	Same as Step 1 for remaining links: Link 1 uses ERIDATA1 Link 2 uses ERIDATA2 Link 3 uses ERIDATA3	Same as Step 1 for remaining links	

Table 42: PHY Configuration ERI (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x06: PHY configuration	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	1: Inform HMC of valid request	
6	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x06: PHY configuration	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request	
				[31]: Start	Start bit polling sequence If [31] = 1, pause for 10µs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step	

Note: 1. 0x0 is recommended unless simulation results indicate the need for a different value.

Table 43: Refresh Rate Adjustment ERI[1, 2]

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[0]: Refresh rate adjustment	0: Default refresh rates 1: Double the default refresh rates
2	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0C: Refresh rate adjustment
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	1: Inform HMC of valid request

Table 43: Refresh Rate Adjustment ERI[1, 2] (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
3	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x0C: Refresh rate adjustment
				[25:8]: Reserved	0x00000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request
				[31]: Start	Start bit polling sequence If [31] = 1, pause for 10μs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step

Notes: 1. HMC automatically adjusts refresh rates, and therefore, the refresh rate adjustment ERI is for debug use only.
2. The refresh rate adjustment ERI may be issued during the configuration register load time or runtime.

Configuring Chained HMC Devices

In a chained system, each cube must be configured to route requests and responses through the cube network and to release tokens upstream in an ordered manner. The routing for each link in each device is configured by using the chaining configuration ERI. This section describes the Chaining Configuration ERI commands and register fields that are required to set up the routing tables.

Prior to executing the chaining configuration ERI, the link configuration register for every link of each cube must have already been properly programmed.

Table 44: Chaining Configuration ERI

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	**Request Map Link 0** [15:0]: Link 0 accesses which cube(s)		1
				Request Map Link 1 [31:16]: Link 1 accesses which cube(s)		
2	MODE WRITE REQUEST	ERIDATA1	0x2B0001	**Request Map Link 2** [15:0]: Link 2 accesses which cube		1
				Request Map Link 3 [31:16]: Link 3 accesses which cube(s)		

Table 44: Chaining Configuration ERI (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
3	MODE WRITE REQUEST	ERIDATA2	0x2B0002	**Solicited Response Map** [15:0]: Which link will send solicited responses		1
				Unsolicited Response Map [31:16]: Which link will send unsolicited responses		1
4	MODE WRITE REQUEST	ERIDATA3	0x2B0003	**Source Link ID (SLID) Map** [15:0]: Which Link is attached to a SLID		1
				Token Dependency Map [31:16] Which link(s) are dependant upon another link(s) for token release		
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x10: Chaining topology config	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	1: Inform HMC of valid request	
6	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x10: Chaining topology config	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x01: Internal error 0x02: Invalid request	
				[31]: Start	Start bit polling sequence If [31] = 1, pause for 10µs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step	

Note: 1. Contact Micron for register configuration values.

Completing Initialization

When the INIT Continue ERI is executed, the device will perform internal initialization tasks and initiate the training sequence on each of the active links. The start bit in the ERIREQ register will be 1 until the internal initialization is complete, the device will then release tokens during the transaction layer initialization. For normal operation, the INIT Continue ERI must be executed after writing all other INIT time registers and executing all configuration ERIs.

Link training must be executed during the time period specified as ^{t}INITCONTINUE, as outlined in the HMC Gen2 data sheet. The default timeout delay may be extended by setting the Timeout Delay field when executing the INIT Continue ERI as shown in the following table. Each bit set in the timeout delay field adds approximately 104µs for a total possible delay of approximately 55 seconds added to the ^{t}INITCONTINUE timeout period. When the internal INIT Continue process completes (link training is achieved or the maximum allowed timeout delay is reached), the start bit will be set to 0 to signify the completion of INIT Continue. If link training is not achieved within the specified time (including any added timeout delay), the INIT Continue read status will indicate a link initialization error.

Table 45: INIT Continue ERI

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[18:0]: Timeout delay	Timeout delay [18:0] multiplied by 104µs = actual timeout delay
				[31:19]: Reserved	0x0000: Reserved
2	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0xFF: INIT Continue
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	1: Inform HMC of valid request
3	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST	0xFF
				[15:8]: Type	0x00
				[21:16]: Target location	0x00
				[25:22]: Size	0x00
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x00: Request was successful 0x01: Request caused an internal error 0x02: Invalid request 0x11: Vault initialization error 0x12: Link initialization error 0x13: Vault initialization and link initialization error 0x14: Critical log warning 0x15: Critical log and vault initialization error 0x16: Critical log and link initialization error 0x17: Critical log, vault, and link initialization error
				[31]: Start	Start bit polling sequence: If [31] = 1, pause 10µs and issue subsequent mode read request to ERIREQ If [31] = 0, continue to next step

Table 46: INIT Continue Error Status Handling

INIT Continue Status	Suggested Action
Request successful (status = 0x00)	Expected status, no errors and device is ready for normal use at this point.
Request caused an internal error (status = 0x01)	Indicates that an internal and possibly fatal error has occurred during INIT Continue ERI execution. Device may be damaged or running at voltage conditions outside of specified range.
Invalid Request (status = 0x02)	Indicates that this ERI hasn't been implemented in this firmware revision. Verify that 0xFF was cleanly transmitted.
Vault initialization error (status = 0x11)	A serious error has occurred while attempting to train the memory interface or initialize DRAM from the vault controller logic. NVM write disable will be automatically asserted by the firmware to prevent unintended SBE or MUE repairs from being executed permanently. Verify that the voltages, temperatures and clocks are within specification. If so, running the BIST function can repair certain issues which may lead to this error, such as a TSV failure. To accomplish this, disable the patrol and demand scrubbing features, enable NVM write, and run the DRAM BIST With Repair ERI to attempt to recover and permanently repair the issue. Note that the DRAM portion of BIST will not be executed and will indicate a failing status if INIT Continue has vault initialization error status. Make note of the TSV status output of BIST to determine if TSVs have been repaired through this process.
Link initialization error (status = 0x12)	The link training with the host did not complete during the time that INIT Continue was being executed. Verify that the host is enabled and sending NULL flits prior to initiating the INIT Continue ERI. Verify that the host is able to train the HMC link in the time required by INIT Continue to complete. If unable to meet this requirement, use the appropriate timeout delay as shown in the INIT Continue ERI detailed description.
(status = 0x13)	Combination of status values 0x11 and 0x12. See suggested actions for each of those two status return values.
Critical log warning (status = 0x14)	The NVM log space is full and an attempt to add to the NVM log occurred. BIST, MUE and SBE repairs can still occur if there is sufficient DRAM repair health, however the logging of the repairs will not be possible. Read the NVM log in order to determine what has filled it by using the NVM Log Read ERIs and address the underlying issue that led to this condition.
(status = 0x15)	Combination of status values 0x11 and 0x14. See suggested actions for each of those two status return values.
(status = 0x16)	Combination of status values 0x12 and 0x14. See suggested actions for each of those two status return values.
(status = 0x17)	Combination of status values 0x11, 0x12 and 0x14. See suggested actions for each of those three status return values.

4.5.5 I^{2}C Configuration Register Load Example

The following is an example of code used to initialize the HMC with the I^{2}C bus during the configuration register load period of the initialization routine. The I^{2}C functions used in the example are represented by "i2cwr" and "i2crd" and depicts properly formatted configuration register WRITE and READ commands as outlined in the HMC data sheet. In this example, all four links are full width and operating at 15 Gb/s with the options noted in the comments in the script. Not all configuration registers are required to be written for normal operation.

```
#===== Global Configuration Direct Register ========================================================
# No changes from default desired, register not written


#===== Request Identification Register =============================================================
# No changes from default desired, register not written.
# This will rely upon the CUB pins for the Cube ID for a positive match to the CUB field in
# packet headers from the host.


#===== Link Run Length Limit Register ==============================================================
# Setting run length limit to 200 decimal (0xC8) to meet requirements of host receiver if
# AC Coupled.  Can leave as default (0x00000000 for infinite run length) if DC Coupled
i2cwr(0x10,0x00240003,0x00C80000) # Link 0 run length 0xC8 (200 dec)
i2cwr(0x10,0x00250003,0x00C80000) # Link 1 run length 0xC8 (200 dec)
i2cwr(0x10,0x00260003,0x00C80000) # Link 2 run length 0xC8 (200 dec)
i2cwr(0x10,0x00270003,0x00C80000) # Link 3 run length 0xC8 (200 dec)


#===== Link Retry Register =========================================================================
# No changes from default desired.  Verify based upon host requirements.


#===== Link Configuration Direct Register ==========================================================
#Setting link configuration as follows:
# bits 1:0 = 0x1 for host/source link
# bit 2 = '0' for response open loop mode off
# bit 3 = '1' for normal packet sequence detection (on)
# bit 4 = '1' to enable link CRC
# bit 5 = '1' to enable duplicate length detection
# bit 6 = '1' to decode/parse incoming packets (normal operation)
# bit 7 = '1' to enable transmission of packets
# bit 8 = '0' to allow links to enter down mode if RXPS set accordingly
# bit 9 = '1' to enable RX descramblers
# bit 10 = '1' to enable TX scramblers
# bit 11 = '1' to enable error response packets
i2cwr(0x10,0x00240000,0x00000EF9) # Link 0 Configuration
i2cwr(0x10,0x00250000,0x00000EF9) # Link 1 Configuration
i2cwr(0x10,0x00260000,0x00000EF9) # Link 2 Configuration
i2cwr(0x10,0x00270000,0x00000EF9) # Link 3 Configuration


#===== Input Buffer Token Count Register ===========================================================
# Setting all links to have 219 (decimal) input buffer tokens
i2cwr(0x10,0x040000,0x000000DB) # Link0 Input Buffer Token Count = 219
i2cwr(0x10,0x050000,0x000000DB) # Link1 Input Buffer Token Count = 219
i2cwr(0x10,0x060000,0x000000DB) # Link2 Input Buffer Token Count = 219
i2cwr(0x10,0x070000,0x000000DB) # Link3 Input Buffer Token Count = 219


#===== Address Configuration Register ==============================================================
# No changes from default desired, so register not modified.


#===== Vault Control Register ======================================================================
# Sets all vaults as follows:
# bits 1:0 = 0x0 DRAM initalized to all 0's
# bit 2 = '1' to enable Hard SBE detect
# bit 3 = '1' to enable Demand scrubbing
# bit 4 = '1' to enable Patrol scrubbing
# bit 5 = '1' to enable CRC detection in packets
# bits 8:6 = 0x1 to enable CA Retry
# bit 9 = '0' to allow SBE correction
# bit 11 = '1' to enable Dynamic Repair
# bit 12 = '0' to disable SBE reporting
# All others Reserved (0x0)
i2cwr(0x10,0x00108000, 0x0000087C)
```

```
#===== Link Configuration ERI ===================================================================
#Set link confiuration ERI as follows (Same across all four links):
#Bits 1:0 = 0x2 for 15G
#Bits 17:16 = 0x0 for full width
#Bits 31:27 = 0x0 for normal operation (no loopback or PRBS generation/check)
# All others Reserved
i2cwr(0x10,0x002B0000, 0x00000002) #Link 0 ERIDATA0
i2cwr(0x10,0x002B0001, 0x00000002) #Link 1 ERIDATA1
i2cwr(0x10,0x002B0002, 0x00000002) #Link 2 ERIDATA2
i2cwr(0x10,0x002B0003, 0x00000002) #Link 3 ERIDATA3
#ERIREQ command for configuring all links based upon ERIDATA registers previously loaded
i2cwr(0x10,0x002B0004, 0x813F0005)
#Bits 7:0 = 0x05 ; Link Config Register Request Command
#Bits 15:8 = 0x00 ; Reserved
#Bits 21:16 = 0x3F ; Global Target
#Bits 25:22 = 0x4 ; Size = 4 for all ERIDATA registers
#Bits 30:26 = 0x00 ; Reserved
#Bit 31 = Start
# i2crd 0xF86B0004 simply reads bit 31 (start bit) of ERIREQ
# Alternatively, host can read 0x2B0004 and detect  bit 31 clear
while (i2crd(0x20,0xF86B0004)): wait(10usec)  # once start bit (bit 31) is cleared ('0') move to next commands/ERI
# Read ERI Request Status value (would be returned with 32b read of 0x2B0004
# 11010 (start bit = 26), 00101 (size = 5), address 1010110000000000000100 (2b0004)
i2crd(0x20,0xD16B0004)


#===== PHY Configuration ERI ===================================================================
#Set PHY confiuration ERI as follows (Same across all four links):
#Bits 4:0 = 0x0 K2 coefficient
#Bits 19:16 = 0x0 DC Coupled RX on board
# All others Reserved (0x0)
i2cwr(0x10,0x002B0000, 0x00000000) #Link 0 ERIDATA0
i2cwr(0x10,0x002B0001, 0x00000000) #Link 1 ERIDATA1
i2cwr(0x10,0x002B0002, 0x00000000) #Link 2 ERIDATA2
i2cwr(0x10,0x002B0003, 0x00000000) #Link 3 ERIDATA3
#ERIREQ command for configuring all links based upon ERIDATA registers previously loaded
i2cwr(0x10,0x002B0004, 0x80000006)
#Bits 7:0 = 0x06 ; PHY Config Register Request Command
#Bit 31 = Start
# i2crd 0xF86B0004 simply reads bit 31 (start bit) of ERIREQ
# Alternatively, host can read 0x2B0004 and detect  bit 31 clear
while (i2crd(0x20,0xF86B0004)): wait(10usec)  # once start bit (bit 31) is cleared ('0') move to next commands/ERI
# Read ERI Request Status value (would be returned with 32b read of 0x2B0004
# 11010 (start bit = 26), 00101 (size = 5), address 1010110000000000000100 (2b0004)
i2crd(0x20,0xD16B0004)


#===== INIT Continue ERI =================================================================
# INIT Continue must be executed in order for the HMC to execute initialization and begin link training.
# Please see "Power on and Initialization section of HMC Datasheet for details.
i2cwr(0x10,0x002B0004, 0x800000FF) #INIT Continue ERIREQ
# Read from 0xF86B0004 simply reads bit 31 (start bit) of ERIREQ
# Alternatively, host can read 0x2B0004 and mask for bit 31 clear
while (i2crd(0x20,0xF86B0004)): wait(10usec) # wait for clear of start bit (bit 31)
# Read ERI Request Status value (would be returned with 32b read of 0x2B0004
# 11010 (start bit = 26), 00101 (size = 5), address 1010110000000000000100 (2b0004)
i2crd(0x20,0xD16B0004)
```

4.5.6 Runtime Registers and ERIs

Registers in this section may be accessed with the I^{2}C or JTAG buses at any point during runtime. Runtime is defined as the period of time after initialization registers and ERIs have been executed, and the start bit in the INIT Continue ERI has been cleared (equals 0). Access with in-band mode register requests must occur after the transaction layer has been initialized.

4.5.6.1 Runtime Status Registers

Table 47: Global Status

Register address: 0x280001

Name	Start Bit	Size	Description
L0RXPS	0	1	Link 0 RX power state level 1: Active state 0: Sleep or down
L1RXPS	1	1	Link 1 RX power state level 1: Active state 0: Sleep or down
L2RXPS	2	1	Link 2 RX power state level 1: Active state 0: Sleep or down
L3RXPS	3	1	Link 3 RX power state level 1: Active state 0: Sleep or down
Reserved	4	2	Reserved
Self refresh	6	1	When 1, HMC is in self refresh state; occurs after all links exit active mode
Reserved	7	25	Reserved

Table 48: Link Training Synchronization Status

Register address: Link 0 = 0x240005; Link 1 = 0x250005; Link 2 = 0x260005; Link 3 = 0x270005

Name	Start Bit	Size	Description
Descrambler sync status	0	16	[0] = Lane 0,... [15] = Lane 15. Bit values of 1 indicate per lane recent descrambled RX bits were zeros, required for descrambler sync. During link training, if the link state (see Link Interface Status) does not progress past Descrambler Sync (0x04); bit values of 0 can identify lanes unable to achieve descrambler sync. After successful link training, all bits will have random values during request traffic, but will have a value of 1 in all active lane positions when a sustained stream of NULL FLITs are driven into the HMC.

Table 48: Link Training Synchronization Status (Continued)

Register address: Link 0 = 0x240005; Link 1 = 0x250005; Link 2 = 0x260005; Link 3 = 0x270005

Name	Start Bit	Size	Description
TS1 sync status	16	16	[0] = Lane 0,... [15] = Lane 15. Bit values of 1 indicate per lane if TS1 is received while in link state TS1 Sync (0x10) or Lane Deskew (0x20). These values are frozen when exiting Lane Deskew (0x20), as a result, these bytes should remain 0xFFFF when the link is trained.

Table 49: Lane Deskew Status

Register address: Link 0 = 0x240006; Link 1 = 0x250006; Link 2 = 0x260006; Link 3 = 0x270006

Name	Start Bit	Size	Description
Lane deskew status	0	32	Value of 0x0 indicates minimal deskew, value of 0x3 indicates maximal deskew. [0:1] = Lane 0,... [31:32] = Lane15.

Table 50: Link Token Status

Register address: Link 0 = 0x0C0001; Link 1 = 0x0D0001; Link 2 = 0x0E0001; Link 3 = 0x0F0001

Name	Start Bit	Size	Description
Response (transmit) token count	0	10	Indicates current number of tokens available for transmitting FLITs
RESERVED	10	6	Reserved
Last RRP	16	8	Indicates last RRP value received on the link
Next FRP	24	8	Indicates FRP value that will be used next

Table 51: Link Interface Status[1]

Register address: Link 0 = 0x240002; Link 1 = 0x250002; Link 2 = 0x260002; Link 3 = 0x270002

Name	Start Bit	Size	Description
Link state	0	8	Link training state 0x01: Reset state 0x02: RX CLK align 0x04: Descrambler sync 0x08: Nonzero wait 0x10: TS1 sync 0x20: Lane deskew 0x40: Zeroes wait 0x80: Link active

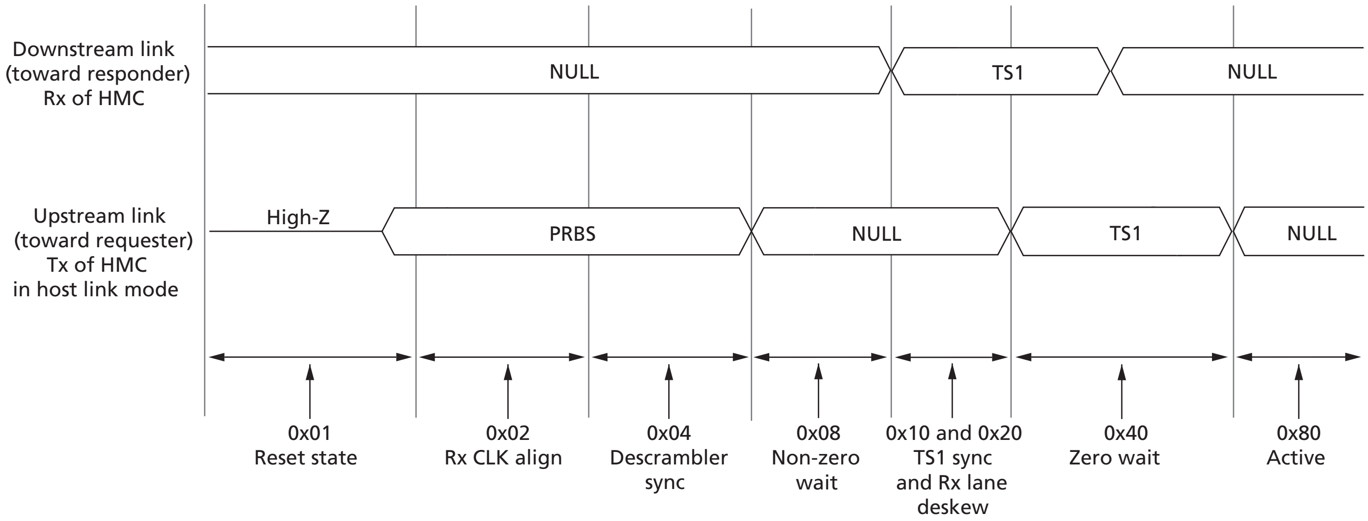
Table 51: Link Interface Status[1] (Continued)

Register address: Link 0 = 0x240002; Link 1 = 0x250002; Link 2 = 0x260002; Link 3 = 0x270002

Name	Start Bit	Size	Description
PLLB lock (required for 15G operation)	8	2	0x0: Neither of the PLLBs associated with the link is locked 0x1: PLLB associated with the half-link configuration is locked 0x3: PLLB PLLs associated with the full-link configuration are locked
PLLA lock (required for 10G and 12.5G operation)	10	2	0x0: Neither of the PLLAs associated with the link is locked 0x1: PLLA associated with half-link configuration is locked 0x3: PLLA PLLs associated with the full-link configuration are locked
Reserved	12	2	Reserved
Link down power state	14	1	0x1: Link is in down mode
Link sleep power state	15	1	0x1: Link is in sleep mode
Link active power state	16	1	0x1: Link is in active mode
Reserved	17	1	Reserved
Lane reversal	18	1	Status of auto-detected lane reversal 0x0: Lanes not reversed 0x1: Lanes reversed
Reserved	19	13	Reserved

Note: 1. Interface status registers are not valid until after INIT Continue.

Figure 63: Link Training State


Downstream link (toward responder) Rx of HMC
NULL
TS1
NULL
Upstream link (toward requester) Tx of HMC in host link mode
High-Z
PRBS
NULL
TS1
NULL
0x01 Reset state
0x02 Rx CLK align
0x04 Descrambler sync
0x08 Non-zero wait
0x10 and 0x20 TS1 sync and Rx lane deskew
0x40 Zero wait
0x80 Active

Note: 1. The link training state begins at Step 6 of the power-on and initialization sequence.

Table 52: Lane Inversion Status

Register address: Link 0 = 0x240004; Link 1 = 0x250004; Link 2 = 0x260004; Link 3 = 0x270004

Name	Start Bit	Size	Description
Lane inversion	0	16	Status of auto-detected lane polarity inversion and correction found during descrambler synchronization; lane numbers map to decimal value 0x0: Lane polarity not inverted 0x1: Lane polarity inverted
Reserved	16	16	Reserved

Table 53: Temperature Monitor ERI Sequence

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x0A: Temperature monitor	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	0x1: Inform HMC of valid request	

Table 53: Temperature Monitor ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
2	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x0A: Temperature monitor	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request	
				[31]: Start	Start bit polling sequence If [31] = 1, pause for 10µs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step	
3	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Logic die temperature recorded	[15:0]: Temperature in ˚C using two's compliment representation Examples: 0˚C: (0x0000) 95C: (0x005F) 125˚C (0x007D)	1, 2
				[31:16]: Reserved	0x000000: Reserved	
4	MODE READ REQUEST	ERIDATA1	0x2B0001	[15:0]: Top DRAM (closest to lid)	[15:0]: Temperature in ˚C using two's compliment representation Examples: 0˚C:0x0000) 95C: (0x005F) 125˚C (0x007D)	1, 2
				[31:16]: Reserved	0x000000: Reserved	
5	MODE READ REQUEST	ERIDATA2	0x2B0002	[15:0]: Bottom DRAM (closest to logic die)	[15:0]: Temperature in ˚C using two's compliment representation Examples: 0˚C: (0x0000) 95C: (0x005F) 125˚C: (0x007D)	1, 2
				[31:16]: Reserved	0x0000: Reserved	

Notes:
1. The reported temperature is accurate within ±5˚C.
2. HMC functionality is only guaranteed within the operational range that is specified in the data sheet.

The Temperature History ERI is used to check historical high temperatures in the HMC. If the maximum operational threshold for temperature has been met or exceeded, the temperature will be logged in the NVM space permanently and reported when this ERI is executed. If the maximum operational has never been exceeded, the values reported when this ERI is executed will correspond to the maximum temperature reached after the previous cold reset.

Table 54: Temperature History ERI Sequence

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x0B: Temperature history	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	0x1: Inform HMC of valid request	
2	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x0B: Temperature history	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request	
				[31]: Start	Start bit polling sequence If [31] = 1, pause for 10μs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step	
3	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Max logic die temperature recorded	[7:0]: Temperature in ˚C Positive temperature: Equal to decimal equivalent of hex value Example: (0x005F = 95˚C)	1
				[31:16]: Reserved	0x0000: Reserved	
4	MODE READ REQUEST	ERIDATA1	0x2B0001	[15:0]: Max DRAM die temperature recorded	[7:0]: Temperature in ˚C Positive temperature: Equal to decimal equivalent of hex value Example: (0x005F = 95˚C)	1
				[31:16]: Reserved	0x0000: Reserved	

Note: 1. The reported temperature is accurate within ±5˚C.

Table 55: DRAM Repair Health Status ERI Sequence

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x20: DRAM repair health status
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request

Table 55: DRAM Repair Health Status ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
2	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x20: DRAM repair health status
				[25:8]: Reserved	0x00000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request 0x14: Critical log warning
				[31]: Start	Start bit polling sequence: If [31] = 1, pause for 10 µsec and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step
3	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Command/address TSV repair health status	[0] = 0: Vault 0 repairs available [0] = 1: Vault 0 repair warning [1] = 0: Vault 1 repairs available [1] = 1: Vault 1 repair warning ⋮ [15] = 0: Vault 15 repairs available [15] = 1: Vault 15 repair warning
				[31:16]: Reserved	0x0000: Reserved
4	MODE READ REQUEST	ERIDATA1	0x2B0001	[15:0]: DQ TSV repair health status	[0] = 0: Vault 0 repairs available [0] = 1: Vault 0 repair warning [1] = 0: Vault 1 repairs available [1] = 1: Vault 1 repair warning ⋮ [15] = 0: Vault 15 repairs available [15] = 1: Vault 15 repair warning
				[31:16]: Reserved	0x0000: Reserved
5	MODE READ REQUEST	ERIDATA2	0x2B0002	[15:0]: DRAM array repair health status	[0] = 0: Vault 0 repairs available [0] = 1: Vault 0 repair warning [1] = 0: Vault 1 repairs available [1] = 1: Vault 1 repair warning ⋮ [15] = 0: Vault 15 repairs available [15] = 1: Vault 15 repair warning
				[31:16]: Reserved	0x0000: Reserved

Table 55: DRAM Repair Health Status ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
6	MODE READ REQUEST	ERIDATA3	0x2B0003	[15:0]: SBE permanent repair status	[0] = 0: Vault 0 ok [0] = 1: Vault 0 repair pending upon cold reset … [15] = 0: Vault 15 ok [15] = 1: Vault 15 repairs pending upon cold reset
				[31:16]: Reserved	0x0000: Reserved

4.5.6.2 DRAM Array Initialization

The DRAM array can be initialized with a custom 16B user data pattern that is repeated throughout the array and aligned on 0 and 16B boundaries. This is a two step process:

1. Early in the initialization, the host must communicate the desired pattern by executing the User DRAM Pattern Definition ERI.
2. The host must then set the DRAM Initialization Mode field of the vault control register with the user-defined pattern prior to an INIT CONTINUE command. After the INIT CONTINUE command, the DRAM array will be initialized by the vault controller with the custom pattern.

Table 56: User DRAM Pattern Definition ERI

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[31:0]: Pattern	User pattern [31:0] (least significant bytes)
2	MODE WRITE REQUEST	ERIDATA1	0x2B0001	[31:0]: Pattern	User pattern [63:32]
3	MODE WRITE REQUEST	ERIDATA2	0x2B0002	[31:0]: Pattern	User pattern [95:64]
4	MODE WRITE REQUEST	ERIDATA3	0x2B0003	[31:0]: Pattern	User pattern [127:96] (most significant bytes)
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0F: Set custom user pattern
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request

Table 56: User DRAM Pattern Definition ERI (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
6	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0F: Set custom user pattern
				[25:8]: Reserved	0x00000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request
				[31]: Start	Start bit polling sequence: If [31] = 1, pause 10µs and issue subsequent mode read request to ERIREQ If [31] = 0, continue to next step

4.5.6.3 Sideband DRAM Access

The HMC has debug and test capability that allow reading and writing through sideband (JTAG or I^{2}C) directly to the DRAM array using the Sideband DRAM Access ERI. If writing with the Sideband DRAM Access ERI, the data written will be the pattern that has been defined by executing the User DRAM Pattern Definition ERI prior to each execution of the Sideband DRAM Access ERI. If reading from the DRAM array with the Sideband DRAM Access ERI, then the data read will be stored in the ERIDATA[3:0] registers upon clearing of the start bit following the Sideband DRAM Access ERI command execution. All in-band traffic should be quiesced to avoid any conflicts in the array that may result in unintended repairs.

Table 57: Sideband DRAM Access ERI (Read)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[3:0]: Byte Address	Address: Byte[3:0] = 0x0	
				[23:4]: DRAM Address	Address: DRAM[19:0]	
				[27:24]: Bank Address	Address: Bank[3:0]	1
				[31:28]: Vault Address	Address: Vault[3:0]	
2	MODE WRITE REQUEST	ERIDATA1	0x2B0001	[0]: Read/Write	0 = Read	
				[31:1]: Reserved	Reserved	
3	MODE WRITE REQUEST	ERIDATA2	0x2B0002	[31:0]: Reserved	Reserved	
4	MODE WRITE REQUEST	ERIDATA3	0x2B0003	[31:0]: Reserved	Reserved	

Table 57: Sideband DRAM Access ERI (Read) (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0E: Sideband DRAM Access	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	0x1: Inform HMC of valid request	
6	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0E: Sideband DRAM Access	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request	
				[31]: Start	Start bit polling sequence: If [31] = 1, pause 10µs and issue subsequent mode read request to ERIREQ If [31] = 0, continue to next step	
7	MODE READ REQUEST	ERIDATA0	0x2B0000	[31:0] DRAM Data	DRAM Data [31:0]	
8	MODE READ REQUEST	ERIDATA1	0x2B0001	[31:0] DRAM Data	DRAM Data [63:32]	
9	MODE READ REQUEST	ERIDATA2	0x2B0002	[31:0] DRAM Data	DRAM Data [95:64]	
10	MODE READ REQUEST	ERIDATA3	0x2B0003	[31:0] DRAM Data	DRAM Data [127:96]	

Note: 1. For 2GB (4-high) HMC devices, address bit Bank[3] will always be 0.

Table 58: Sideband DRAM Access ERI (Write)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	Execute the user DRAM Pattern Definition ERI to define data pattern to be written					
2	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[3:0]: Byte Address	Address: Byte[3:0] = 0x0	
				[23:4]: DRAM Address	Address: DRAM[19:0]	
				[27:24]: Bank Address	Address: Bank[3:0]	1
				[31:28]: Vault Address	Address: Vault[3:0]	
3	MODE WRITE REQUEST	ERIDATA1	0x2B0001	[0]: Read/Write	1 = Write	
				[31:1]: Reserved	Reserved	

Table 58: Sideband DRAM Access ERI (Write) (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
4	MODE WRITE REQUEST	ERIDATA2	0x2B0002	[31:0]: Reserved	Reserved	
5	MODE WRITE REQUEST	ERIDATA3	0x2B0003	[31:0]: Reserved	Reserved	
6	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0E: Sideband DRAM Access	
				[30:8]: Reserved	0x000000: Reserved	
				[31]: Start	0x1: Inform HMC of valid request	
7	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x0E: Sideband DRAM Access	
				[25:8]: Reserved	0x00000: Reserved	
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: Request was successful 0x02: Invalid request	
				[31]: Start	Start bit polling sequence: If [31] = 1, pause 10µs and issue subsequent mode read request to ERIREQ If [31] = 0, continue to next step	

Note: 1. The HMC Gen2 device (2GB, 4-high) address bit Bank[3] will always be 0.

4.5.6.4 NVM Log Read ERI

The HMC logs select events in internal non-volatile memory (NVM), which can be accessed by the host or sideband processor. The NVM log can be read at any time after INIT Continue completion by using the NVM Log Read ERI. The events logged include SBE repair addresses, MUE repair addresses, and maximum temperatures for logic and DRAM. There are two ERI commands used to read out the entire log file: Initially the LOG READ START command is issued and returns log data, then LOG READ NEXT commands are issued until the log has been completely read. The log should be read completely and concatenated in order to ensure that all data associated with each entry has been read. Entries vary in length but are made up of a multiple of 32 bits, and the final entry will read ERIDATAx [31] (valid bit) set to 0. The following table details how to read and interpret the log file.

Table 59: NVM Log Read Start and NVM Log Read Next ERI Sequence

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x21: NVM log read start
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request
2	MODE READ REQUEST	ERIREQ	0x2B0004	[25:0]: Reserved	0x000000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x01: Request caused an internal error 0x02: Request was invalid 0x03: Multi-cycle request complete
				[31]: Start	Start bit polling sequence: If [31] = 1, pause for 10 µsec and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step
3	MODE READ REQUEST	ERIDATA0	0x2B0000	[31:0]: Log Data	First NVM log data entry
4	MODE READ REQUEST	ERIDATA1	0x2B0001	[31:0]: Log Data	Subsequent NVM log data entry
5	MODE READ REQUEST	ERIDATA2	0x2B0002	[31:0]: Log Data	Subsequent NVM log data entry
6	MODE READ REQUEST	ERIDATA3	0x2B0003	[31:0]: Log Data	Subsequent NVM log data entry
7	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x22: NVM log read next
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request
8	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x22: NVM Log read next
				[25:8]: Reserved	0x000000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x01: Request caused an internal error 0x02: Invalid request 0x03: Multi-cycle request complete
				[31]: Start	Start bit polling sequence: If [31] = 1, pause for 10 µsec and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step

Table 59: NVM Log Read Start and NVM Log Read Next ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
9	MODE READ REQUEST	ERIDATA0	0x2B0000	[31:0]: Log Data	Subsequent NVM log data entry
10	MODE READ REQUEST	ERIDATA1	0x2B0001	[31:0]: Log Data	Subsequent NVM log data entry
11	MODE READ REQUEST	ERIDATA2	0x2B0002	[31:0]: Log Data	Subsequent NVM log data entry
12	MODE READ REQUEST	ERIDATA3	0x2B0003	[31:0]: Log Data	Subsequent NVM log data entry
13	Repeat steps 7 through 12 until valid bit (ERIDATAx [31]) = 0, indicating the last valid entry in the NVM log				

Table 60: Maximum Temperature Log Entries

Logs maximum logic die or DRAM die temperature recorded to date

Log Fields	Log Field Description
[7:0]: Info	[7:0]: Maximum temperature in ˚C Positive temperature: Equal to decimal equivalent of hex value Example: (0x005F = 95˚C)
[15:8]: Reserved	0x00 Reserved
[18:16]: Entries Remaining	0x0: No additional 32b entries remain
[23:19]: Reserved	0x0000: Reserved
[29:24]: Type	0x01: Maximum DRAM temperature log 0x02: Maximum logic temperature log
[30]: Sequence	0: First entry since cold reset 1: Subsequent entry since cold reset
[31]: Valid	1: Entry is valid

Table 61: MUE Repair Log Entry

Data about MUE repair event, logged when it occurred

Entry in Sequence	Log Fields	Log Field Description
First	[15:0]: Reserved	0x0000: Reserved
	[17:16]: Entries Remaining	0x1: One additional 32b entry remains
	[23:18]: Reserved	0x00: Reserved
	[29:24]: Type	0x04: MUE repair
	[30]: Sequence	0: First entry since cold reset 1: Subsequent entry since cold reset
	[31]: Valid	1: Entry is valid

Table 61: MUE Repair Log Entry (Continued)

Data about MUE repair event, logged when it occurred

Entry in Sequence	Log Fields	Log Field Description
Second	[0]: Reserved	0: Reserved
	[19:1]: DRAM Address	DRAM address [19:1]
	[23:20]: Bank Address	Bank address [3:0]
	[27:24]: Vault Address	Vault address [3:0]
	[30:28]: Reserved	Reserved (may be non-zero)
	[31]: Valid	1: Entry is valid

Table 62: SBE Repair Log Entry

Data about SBE repair event, logged when it occurred

Entry in Sequence	Log Fields	Log Field Description
First	[15:0]: Reserved	0x0000: Reserved
	[17:16]: Entries Remaining	0x1: One additional 32b entry remains
	[23:18]: Reserved	0x00: Reserved
	[29:24]: Type	0x03: SBE error repair
	[30]: Sequence	0: First entry since cold reset 1: Subsequent entry since cold reset
	[31]: Valid	1: Entry is valid
Second	[0]: Reserved	0: Reserved
	[19:1]: DRAM Address	DRAM address [19:1]
	[23:20]: Bank Address	Bank address [3:0]
	[27:24]: Vault Address	Vault address [3:0]
	[30:28]: Reserved	Reserved (may be non-zero)
	[31]: Valid	1: Entry is valid

Table 63: BIST Repaired DRAM Row Log Entry

Data about repairs to the DRAM that were implemented by running the BIST ERI

Entry in Sequence	Log Fields	Log Field Description
First	[15:0]: Reserved	0x0000: Reserved
	[17:16]: Entries Remaining	0x1: One additional 32b entry remains
	[23:18]: Reserved	0x00: Reserved
	[29:24]: Type	0x10: Row repaired by BIST
	[30]: Sequence	0: First entry since cold reset 1: Subsequent entry since cold reset
	[31]: Valid	1: Entry is valid

Table 63: BIST Repaired DRAM Row Log Entry (Continued)

Data about repairs to the DRAM that were implemented by running the BIST ERI

Entry in Sequence	Log Fields	Log Field Description
Second	[6:0]: Reserved	0x00: Reserved
	[19:7]: DRAM Address	DRAM address [19:7]
	[23:20]: Bank Address	Bank address [3:0]
	[27:24]: Vault Address	Vault address [3:0]
	[30:28]: Reserved	Reserved (may be non-zero)
	[31]: Valid	1: Entry is valid

Table 64: BIST Repaired DRAM Column Log Entry

Data about repairs to the DRAM implemented by running the BIST ERI

Entry in Sequence	Log Fields	Log Field Description
First	[15:0]: Reserved	0x0000: Reserved
	[17:16]: Entries Remaining	0x1: One additional 32b entry remains
	[23:18]: Reserved	0x00: Reserved
	[29:24]: Type	0x11: Column repaired by BIST
	[30]: Sequence	0: First entry since cold reset 1: Subsequent entry since cold reset
	[31]: Valid	1: Entry is valid
Second	[19:0]: Reserved	Reserved (may be non-zero)
	[23:20]: Bank Address	Bank address [3:0]
	[27:24]: Vault Address	Vault address [3:0]
	[30:28]: Reserved	Reserved (may be non-zero)
	[31]: Valid	1: Entry is valid

Table 65: BIST Repaired TSV Entry

Data about repairs to the TSV (3D-interconnect) implemented by running the BIST ERI

Entry in Sequence	Log Fields	Log Field Description
First	[3:0]: Vault Address	Vault address
	[15:4]: Reserved	0x0000: Reserved
	[17:16]: Entries Remaining	0x0: No additional 32b entries remain
	[23:18]: Reserved	0x00: Reserved
	[29:24]: Type	0x13: BIST repaired TSV
	[30]: Sequence	0: First entry since cold reset 1: Subsequent entry since cold reset
	[31]: Valid	1: Entry is valid

4.5.6.5 Reset and Power-Down Considerations

The HMC must not be powered down or cold reset while in the process of repairing or writing to nonvolatile memory. In order to ensure that the HMC completes critical tasks prior to shutting down, the Shutdown ERI must be executed prior to powering down or cold resetting the device. After the command has successfully completed, no further operations will be supported, and power should be removed or the device reset.

The HMC has a built in mechanism that allows a "soft" link power state control such that the LxRXPS pins can be overridden during runtime. If the Soft Link Power State ERI is executed and changes any link's current power state, all link power state control must use this ERI on all links until hardware control using the LxRXPS pins is regained by a cold reset. All requirements as outlined in the Power-On and Initialization section of the HMC Gen2 data sheet still apply, including the appropriate setting of LxRXPS pins and Link Configuration ERI. The Soft Link Power-Down ERI requires that all links be configured (all ERIDATA registers used) each time the ERI is executed.

Table 66: Shutdown ERI

Used to allow a safe power-down or cold reset

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x28: Shutdown
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request
2	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x28: Shutdown
				[25:8]: Reserved	0x000000
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x00: Request was successful 0x01: Request caused an internal error 0x02: Invalid request
				[31]: Start	Start bit polling sequence: If [31] = 1, pause 10µs and issue subsequent mode read request to ERIREQ If [31] = 0, proceed with reset or power-down

Table 67: Soft Link Power State ERI

Used to allow the override of LxRXPS power state after entering runtime

Step	Command	Register Name	Register Address	Register Fields	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[0]: Power	0: Override L0RXPS pin and force link state inactive (sleep or down) 1: Override L0RXPS pin and force link state active
				[31:1]: Reserved	0x0000000
2	MODE WRITE REQUEST	ERIDATA1	0x2B0001	[0]: Power Down	0: Override L1RXPS pin and force link state inactive (sleep or down) 1: Override L1RXPS pin and force link state active
				[31:1]: Reserved	0x0000000
3	MODE WRITE REQUEST	ERIDATA2	0x2B0002	[0]: Power Down	0: Override L2RXPS pin and force link state inactive (sleep or down) 1: Override L2RXPS pin and force link state active
				[31:1]: Reserved	0x0000000
4	MODE WRITE REQUEST	ERIDATA3	0x2B0003	[0]: Power Down	0: Override L3RXPS pin and force link state inactive (sleep or down) 1: Override L3RXPS pin and force link state active
				[31:1]: Reserved	0x0000000
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x11: Power-down
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request
6	MODE READ REQUEST	ERIREQ	0x2B0004	[7:0]: REGISTER REQUEST command	0x11: Power-down
				[25:8]: Reserved	0x000000
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x00: Request was successful 0x01: Request caused an internal error 0x02: Invalid request
				[31]: Start	Start bit polling sequence: If [31] = 1, pause 10µs and issue subsequent mode read request to ERIREQ If [31] = 0, proceed with reset or power-down

4.5.6.6 Built-In Self Test and Repair

The built-in self test (BIST) consists of using built-in algorithms implemented in hardware and software within the device to test the integrity of the DRAM array. Testing is controlled by the HMC after the operation is initiated by the host. Micron determines the breadth of the test suite and the range of algorithms included within the BIST function.

Use of the DRAM BIST registers is optional but recommended, particularly following significant temperature fluctuation as experienced during manufacturing and attachment to a printed circuit board. The DRAM BIST registers are runtime registers and may only be loaded with the I^2C or JTAG sideband interface after the start bit of the INIT Continue ERI has been cleared (0). If a vault initialization error status is returned by the execution of the INIT Continue ERI, see the INIT Continue Error Status Handling table for further details prior to running BIST.

DRAM BIST can be run with or with out the repair option enabled. When running the DRAM BIST WITH REPAIR command sequence, the Disable NVM Write bit must not be set or repairs will not take place.

The host is responsible for quiescing traffic on all links prior to running DRAM BIST. **Any traffic to the array during DRAM BIST will cause BIST failures and can cause permanent damage to the device.**

The state of HMC after DRAM BIST or repair depends upon the type of command run and the whether or not repairs were made.

Table 68: DRAM BIST and Manual Repair Command Results

Command	Faults Found/Repairs to be Performed	HMC Logic and DRAM Status
DRAM BIST WITHOUT REPAIR	Faults may or may not be found/ Repairs will not be performed	Status bits will be returned at completion of ERI and start bit will be cleared when MODE READs are performed as shown in Table 70 (page 148). DRAM will be initialized based upon value set with User Pattern Definition ERI and DRAM Initialization Mode field in the vault control register.
DRAM BIST WITH REPAIR	No faults found/ No repairs required	Status bits will be returned at completion of ERI when MODE READs are performed as shown in Table 69 (page 146). DRAM will be initialized based upon value set with User Pattern Definition ERI and DRAM Initialization Mode field in the vault control register.
DRAM BIST WITH REPAIR or MANUAL REPAIR	Between 1 to 64 faults found/ Repairs will be performed	The required repairs will be performed, start bit will be cleared, and status will be returned when MODE READs are performed as shown in Table 69 (page 146). Up to 64 DRAM repairs and up to 2 TSV repairs will be performed. **A cold reset must be completed before any further operations are carried out.**

Table 68: DRAM BIST and Manual Repair Command Results (Continued)

Command	Faults Found/Repairs to be Performed	HMC Logic and DRAM Status
DRAM BIST WITH REPAIR	More than 64 faults found/ Repairs will not be performed	Status bits will be returned at completion of ERI when MODE READs are performed as shown in Table 69 (page 146). If more than 64 repairs are found to be required in one run of the DRAM BIST with repairs, the repairs will not be made. This condition should only be seen if the device has a serious issue that may or may not be related to the state of the DRAM array. In this case, repairs are not written in order to avoid using all repair space for an issue that is unrelated to a needed DRAM repair. A cold reset is not required because repairs were not performed.
DRAM BIST WITH REPAIR or DRAM BIST WITHOUT REPAIR or MANUAL REPAIR	Non-zero ERI status returned	**Invalid Request** (value = 0x02): Indicates that this ERI hasn't been implemented in the firmware revision of this device. **Critical Log Warning** (value = 0x14): Indicates that the log space in NVM is full and an attempt to write to NVM occurred, but the field repair table is not full. As the field repair tables are not yet full, repairs were made but not logged into the NVM log. **Critical Repair Warning** (value = 0x15): The field repair table in NVM is full and no further repairs can be completed. If received as a response to the BIST With Repair ERI, one or more repairs was not completed. If received as a response to Manual Repair ERI, the manual repair was not completed. **Repair Warning** (value = 0x16): Indicates only one NVM entry is left in the field repair table after writing the last repair. The current repair will be applied but future repairs may result in a Critical Repair Warning.

Table 69: DRAM BIST With Repair ERI Sequence

Step	Command	Register Name	Register Address	Register Field	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[15:0]: Vault mask	Bit 0 = 0: Test Vault 0 Bit 0 = 1: Mask Vault 0 Bit 1 = 0: Test Vault 1 Bit 1 = 1: Mask vault 1 ... Bit 15 = 0: Test Vault 15 Bit 15 = 1: Mask vault 15
				[17:16]: Vaults in parallel [1]	0x0: Perform BIST on all vaults in parallel 0x1: Perform BIST on one vault at a time 0x2: Perform BIST on four vaults in parallel 0x3: Perform BIST on eight vaults in parallel
				[31:18]: Reserved	0x00000: Reserved

Table 69: DRAM BIST With Repair ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Field	Register Field Value
2	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x07: DRAM BIST with repair
				[15:8]: Type	0x00: Reserved
				[21:16]: Target location	0x00: Ignored
				[25:22]: Size	0x00: Ignored
				[30:26]: External request status	0x00: Read only
				[31]: Start	0x1: Inform HMC of valid request
3	MODE READ REQUEST	ERIREQ	0x2B0004	[25:0]: Reserved	0x000000 Reserved
				[30:26]: ERI Status	0x00: ERI successful 0x01: ERI internal error (repair not successful) 0x02: Invalid command (repair not successful) 0x14: Critical log warning 0x15: Critical repair warning (repair not successful) 0x16: Repair warning
				[31]: Start	Start bit polling sequence: If [31] = 1: Pause for 10µs and issue subsequent mode read request to ERIREQ If [31] = 0: Continue to next step
4	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Vault repair required	Bit 0 = 0: Vault 0 has no element failures (no repairs needed) Bit 0 = 1: Vault 0 has one or more elements requiring repairs Bit 1 = 0: Vault 1 has no element failures (no repairs needed) Bit 1 = 1: Vault 1 has one or more elements requiring repairs ... Bit 15 = 0: Vault 15 has no element failures (no repairs needed) Bit 15 = 1: Vault 15 has one or more elements requiring repairs
				[31:16]: Vault repair status	Bit 16 = 0: Vault 0 is repairable Bit 16 = 1: At least one element in Vault 0 is not repairable Bit 17 = 0: Vault 1 is repairable Bit 17 = 1: At least one element in Vault 1 is not repairable … Bit 31 = 0: Vault 15 is repairable Bit 31 = 1: At least one element in Vault 15 is not repairable

Table 69: DRAM BIST With Repair ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Field	Register Field Value
5	MODE READ REQUEST	ERIDATA1	0x2B0001	[31:0]: Vault 0–7 DRAM fail counts	Bits 3:0 = Vault 0 DRAM error count Bits 7:4 = Vault 1 DRAM error count ... Bits 31:28 = Vault 7 DRAM error count
6	MODE READ REQUEST	ERIDATA2	0x2B0002	[31:0]: Vault 8-15 DRAM fail counts	Bits 3:0 = Vault 8 DRAM error count Bits 7:4 = Vault 9 DRAM error count ... Bits 31:28 = Vault 15 DRAM error count
7	MODE READ REQUEST	ERIDATA3	0x2B0003	[4:0]: Rerun request	Bit 0 = 1: Request to rerun DRAM BIST after cold reset Bit 1 = 1: Temperature error, rerun at lower temp Bit 2 = 1: DRAM training error, rerun at different temp or voltage Bit 3: Greater than 64 repairs needed Bit 4 = 1: DRAM BIST is locked out by a prior SBE permanent, MUE, Manual repair process, or a previous BIST failure. A cold reset or power cycle is required.
				[15:5]: Reserved	Reserved
				[18:16] Types of repairs performed	Bit 16 = 1: TSV(s) Repaired Bit 17 = 1: Row(s) Repaired Bit 18 = 1: Column(s) Repaired
				[31:19]: Reserved	0x0000: Reserved

Note: 1. Ensure that the power delivery network and supplies are able to provide the required current for the number of parallel vaults being run in the BIST command.

Table 70: DRAM BIST Without Repair ERI Sequence

Step	Command	Register Name	Register Address	Register Field	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[15:0]: Vault mask	Bit 0 = 0: Test Vault 0 Bit 0 = 1: Mask Vault 0 Bit 1 = 0: Test Vault 1 Bit 1 = 1: Mask vault 1 ... Bit 15 = 0: Test Vault 15 Bit 15 = 1: Mask vault 15
				[17:16]: Vaults in parallel [1]	0x0: Perform BIST on all vaults in parallel 0x1: Perform BIST on one vault at a time 0x2: Perform BIST on four vaults in parallel 0x3: Perform BIST on eight vaults in parallel
				[31:18]: Reserved	0x00000: Reserved

Table 70: DRAM BIST Without Repair ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Field	Register Field Value
2	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x08: DRAM BIST without repair
				[15:8]: Type	0x00: Reserved
				[21:16]: Target location	0x00: Ignored
				[25:22]: Size	0x00: Ignored
				[30:26]: External request status	0x00: Read only
				[31]: Start	0x1: Inform HMC of valid request
3	MODE READ REQUEST	ERIREQ	0x2B0004	[25:0]: Reserved	0x000000 Reserved
				[30:26]: ERI Status	0x00: ERI successful 0x01: ERI internal error 0x02: Invalid command 0x15: Critical repair warning
				[31]: Start	Start bit polling sequence: If [31] = 1: pause for 10μs and issue subsequent mode read request to ERIREQ If [31] = 0: continue to next step
4	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Vault repair required	Bit 0 = 0: Vault 0 has no element failures (no repairs needed) Bit 0 = 1: Vault 0 has one or more elements requiring repairs Bit 1 = 0: Vault 1 has no element failures (no repairs needed) Bit 1 = 1: Vault 1 has one or more elements requiring repairs ... Bit 15 = 0: Vault 15 has no element failures (no repairs needed): Bit 15 = 1: Vault 15 has one or more elements requiring repairs
				[31:16]: Vault repair status	Bit 16 = 0: Vault 0 is repairable Bit 16 = 1: At least one element in Vault 0 is not repairable Bit 17 = 0: Vault 1 is repairable Bit 17 = 1: At least one element in Vault 1 is not repairable … Bit 31 = 0: Vault 15 is repairable Bit 31 = 1: At least one element in Vault 15 is not repairable

Table 70: DRAM BIST Without Repair ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Field	Register Field Value
5	MODE READ REQUEST	ERIDATA1	0x2B0001	[31:0]: Vault 0–7 DRAM fail counts	Bits 3:0 = Vault 0 DRAM error count Bits 7:4 = Vault 1 DRAM error count ... Bits 31:28 = Vault 7 DRAM error count
6	MODE READ REQUEST	ERIDATA2	0x2B0002	[31:0]: Vault 8–15 DRAM fail counts	Bits 3:0 = Vault 8 DRAM error count Bits 7:4 = Vault 9 DRAM error count ... Bits 31:28 = Vault 15 DRAM error count
7	MODE READ REQUEST	ERIDATA3	0x2B0003	[4:0]: Rerun request	Bit 0 = 1: Request to rerun DRAM BIST to correct additional errors Bit 1 = 1: Temperature error, rerun at lower temp Bit 2 = 1: DRAM training error, rerun at different temp or voltage Bit 3: Greater than 64 repairs needed Bit 4: DRAM BIST locked out by prior SBE permanent, MUE, Manual repair process, or a previous BIST failure. A cold reset or power cycle is required.
				[15:5]: Reserved	Reserved
				[18:16]: Types of repairs needed	Bit 16 = 1: TSV repair(s) needed Bit 17 = 1: Row repair(s) needed Bit 18 = 1: Column repair(s) needed
				[31:19]: Reserved	0x0000: Reserved

Note: 1. Ensure that the power delivery network and supplies are able to provide the required current for the number of parallel vaults being run in the BIST command.

Manual DRAM Repair ERI Sequence

If a DRAM array defect is at a known address, it is possible to manually repair the DRAM array by using the Manual DRAM Repair ERI. A manual repair may be desirable if testing outside of BIST has indicated a failure that has not been repaired by SBE Repair, MUE Repair, or DRAM BIST. Micron recommends the NVM log be read using the NVM log READ commands to verify that a repair was not already completed for this address prior to running the Manual DRAM Repair ERI sequence. When running the Manual DRAM Repair ERI sequence, the Disable NVM Write bit must be set to 0 and the DRAM Repair Health Status ERI must indicate that repairs are available, or repairs will not take place. After completion of the Manual DRAM Repair ERI sequence, a cold reset or power cycle is required before any other operations can be carried out. A block of addresses is repaired when the Manual DRAM Repair ERI is executed; therefore, before making multiple repairs, the array should be checked for errors remaining following the cold reset after each Manual DRAM Repair ERI is executed.

Table 71: Manual DRAM Repair ERI Sequence

Step	Command	Register Name	Register Address	Register Field	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[31:28]: Vault Address	Vault Address [3:0] for targeted repair
				[27:24]: Bank Address	Bank Address [3:0] for targeted repair
				[23:4]: DRAM Address	DRAM Address [19:0] for targeted repair
				[3:0]: Reserved	0x0: Reserved
2	MODE WRITE REQUEST	ERIDATA1	0x2B0001	[31:28]: Inverted Vault Address	Bitwise inversion of targeted Vault Address [3:0]
				[27:24]: Inverted Bank Address	Bitwise inversion of targeted Bank Address [3:0]
				[23:4]: Inverted DRAM Address	Bitwise inversion of targeted DRAM Address [19:0]
				[3:0]: Reserved	0xF: Bitwise inversion of ERIDATA0[3:0] from previous step
3	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x09: MANUAL REPAIR
				[15:8]: Type	0x00: Reserved
				[21:16]: Target location	0x00: Ignored
				[25:22]: Size	0x00: Ignored
				[30:26]: External request status	0x00: Read only
				[31]: Start	0x1: Inform HMC of valid request
4	MODE READ REQUEST	ERIREQ	0x2B0004	[25:0]: Reserved	0x000000 Reserved
				[30:26]: ERI Status	0x00: ERI successful 0x01: ERI internal error 0x02: Invalid command 0x14: Critical log warning 0x15: Critical repair warning 0x16: Repair warning
				[31]: Start	Start bit polling sequence: If [31] = 1: pause for 10µs and issue subsequent mode read request to ERIREQ If [31] = 0: continue to next step
5	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Reserved	Reserved
				[31:16]: Vault repair status	Bit 16 = 0: Address in vault 0 is repairable Bit 16 = 1: Address in vault 0 is not repairable ... Bit 31 = 0: Address in vault 15 is repairable Bit 31 = 1: Address in vault 15 is not repairable
6	MODE READ REQUEST	ERIDATA1	0x2B0001	Reserved	Reserved
7	MODE READ REQUEST	ERIDATA2	0x2B0002	Reserved	Reserved

Table 71: Manual DRAM Repair ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Field	Register Field Value
8	MODE READ REQUEST	ERIDATA3	0x2B0003	[16:0]: Reserved	Reserved
				[17]: Repair successful	0x1: Address repaired
				[31:18]: Reserved	Reserved

4.5.6.7 Firmware Patching

Many HMC functions are controlled or impacted by a simple internal processor. The firmware this processor executes is stored internal to the HMC in one-time programmable nonvolatile memory (NVM), and therefore, has limitations in terms of upgradeability. There is, however, a procedure for writing patches to a reserved portion of NVM. Patches are version-specific and care must be taken to ensure that the firmware is not corrupted in any way. It is critical that voltages be within specification on all rails during this process. If voltages drop below specification during the Patch Apply ERI execution, permanent and irreversible damage to the device is likely.

The patch file, which must be obtained directly from Micron, will be in Intel HEX file format, and the file must be parsed before sending the contents using the Patch Load ERI. The Patch Load and Patch Apply ERIs must be executed prior to reaching runtime (before initiating the INIT Continue ERI) and a cold reset is required following successful application.

A single example line showing the expected format is shown in Figure 64 (page 152), and the corresponding ERIDATA byte locations are shown in Figure 65 (page 152). The data size, address, end-of-file, and CRC fields are all present but must not be sent to the HMC through the ERI functions. Because the Intel HEX format is little endian, care must be taken in the realignment of the byte locations. A full file example is shown below.

Figure 64 (page 152) shows one line of the example file format to illustrate the pertinent data to be loaded into the HMC via the Patch Load ERI.

Figure 64: Intel HEX Format Example

Digital size	Address	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	CRC
:10	0000	01	23	45	67	89	AB	CD	EF	DE	AD	BE	EF	A1	A2	A3	A4	A5

Figure 65: Patch ERIDATA Register Ordering Example

ERIDATA0 = 0x67452301
ERIDATA1 = 0xEFCDAB89
ERIDATA2 = 0xEFBEADDE
ERIDATA3 = 0xA4A3A2A1

Table 72: Patch Load and Patch Apply ERI Process

Step	Command	Register Name	Register Address	Register Field	Register Field Value
1	MODE WRITE REQUEST	ERIDATA0	0x2B0000	[7:0]	Hex Data D0
				[15:8]	Hex Data D1
				[23:16]	Hex Data D2
				[31:24]	Hex Data D3
2	MODE WRITE REQUEST	ERIDATA1	0x2B0001	[7:0]	Hex Data D4
				[15:8]	Hex Data D5
				[23:16]	Hex Data D6
				[31:24]	Hex Data D7
3	MODE WRITE REQUEST	ERIDATA2	0x2B0002	[7:0]	Hex Data D8
				[15:8]	Hex Data D9
				[23:16]	Hex Data D10
				[31:24]	Hex Data D11
4	MODE WRITE REQUEST	ERIDATA3	0x2B0003	[7:0]	Hex Data D12
				[15:8]	Hex Data D13
				[23:16]	Hex Data D14
				[31:24]	Hex Data D15
5	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x26: Patch Load
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request
6	MODE READ REQUEST	ERIREQ	0x2B0004	[25:0]: Reserved	0x000000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: ERI was successful 0x01: Request caused an internal error 0x02: Request was invalid
				[31]: Start	Start bit polling sequence: If [31] = 1, pause for 10 µsec and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step
7	**Repeat Steps 1 though 6 until entire patch file is loaded. ":00000001FF" will indicate the end of the patch file.**				
8	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x27: Patch apply
				[30:8]: Reserved	0x000000: Reserved
				[31]: Start	0x1: Inform HMC of valid request

Table 72: Patch Load and Patch Apply ERI Process (Continued)

Step	Command	Register Name	Register Address	Register Field	Register Field Value
9	MODE READ REQUEST	ERIREQ	0x2B0004	[25:0]: Reserved	0x000000: Reserved
				[30:26]: External request status	Status will be valid after start bit is cleared: 0x0: ERI was successful 0x01: Request caused an internal error 0x02: Request was invalid
				[31]: Start	Start bit polling sequence: If [31] = 1, pause for 10 µsec and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step
10	MODE READ REQUEST	ERIDATA0	0x2B0000	[15:0]: Status	0x00: Patch applied successfully 0x01: Patch applied successfully but not blown 0x02: Patch is empty 0x03: Checksum Error 0x04: Unrecognized patch file 0x05: Validation failed 0x06: Validation failed after some NVM blown 0x07: Table in NVM is full
				[31:16]: Error Pointer	If Status contains an error, Error Pointer will contain address where error occurred

Patch Load and Patch Apply ERI Example

```
:10000000 1600010000000140A076060000000000 7C
:10001000 00000040227406000000000000000040 C4
:10002000 23740600000000002274001000009010 ED
:10003000 A0760110214700849010000023740010 66
:10004000 15011520010000809F76060000000000 C9
:10005000 02000090001200709010009509A947D0 8E
:00000001FF

#ERI Loading patch 0: 0x00010016 0x40010000 0x000676a0 0x00000000
ERIDATA1 <= 0x00010016;
ERIDATA2 <= 0x40010000;
ERIDATA3 <= 0x000676A0;
ERIDATA4 <= 0x00000000;
ERIREQ <= 0x80000026;

#ERI Loading patch 1: 0x40000000 0x00067422 0x00000000 0x40000000
ERIDATA1 <= 0x40000000;
ERIDATA2 <= 0x00067422;
ERIDATA3 <= 0x00000000;
ERIDATA4 <= 0x40000000;
ERIREQ <= 0x80000026;

#ERI Loading patch 2: 0x00067423 0x00000000 0x10007422 0x10900000
ERIDATA1 <= 0x00067423;
ERIDATA2 <= 0x00000000;
```

```
ERIDATA3 <= 0x10007422;
ERIDATA4 <= 0x10900000;
ERIREQ <= 0x80000026;

#ERI Loading patch 3: 0x100176a0 0x84004721 0x00001090 0x10007423
ERIDATA1 <= 0x100176A0;
ERIDATA2 <= 0x84004721;
ERIDATA3 <= 0x00001090;
ERIDATA4 <= 0x10007423;
ERIREQ <= 0x80000026;

#ERI Loading patch 4: 0x20150115 0x80000001 0x0006769f 0x00000000
ERIDATA1 <= 0x20150115;
ERIDATA2 <= 0x80000001;
ERIDATA3 <= 0x0006769F;
ERIDATA4 <= 0x00000000;
ERIREQ <= 0x80000026;

#ERI Loading patch 5: 0x90000002 0x70001200 0x95001090 0xd047a909
ERIDATA1 <= 0x90000002;
ERIDATA2 <= 0x70001200;
ERIDATA3 <= 0x95001090;
ERIDATA4 <= 0xD047A909;
ERIREQ <= 0x80000026;

#ERI Apply patch
ERIREQ <= 0x80000027;
```

4.6 Debug Guide

When bringing up a new system, it may be necessary to track certain metrics and understand errors which may occur. This section describes how to access, interpret, and react to various metrics or errors in order to ensure robust system functionality.

4.6.1 SerDes Eye Metrics

Each HMC SerDes has a built-in mechanism to measure the incoming data eye metrics: height (in voltage reference taps) and width (in unit delay taps). The values indicate the size of the inner eye at the bit error ratio (BER) of $1E^{-6}$. These metrics can be used to estimate the signal integrity at each HMC receiver lane. They can be read through eye metrics registers on a per-lane basis and are updated continuously by the HMC. Reading the eye metrics registers repeatedly and aggregating the results will provide a higher degree of accuracy.

A link with an increased rate of CRC errors, exhibited by a high rate of link retry events, may have a lane with a smaller eye metric. The individual lane with the smallest eye metric may be the cause of the CRC errors. A small eye metric can be the result of a conductivity or impedance anomaly anywhere from the transmitter to the receiver.

The eye metrics of a high speed serial link are measured at the receiver. The eye metric reported by the HMC is the HMC receiver half of the link. The eye metrics for the half of the link where the HMC is the transmitter need to be collected at the associated receiver.

The eye metrics should be compared to the expected metrics from the simulation results to ensure that the board is fabricated and assembled according to the simulated design.

The SerDes RX eye width registers provide the width of the received data eye, as measured at the HMC receiver lane. The value read will be a five-bit hexadecimal value representing the eye width in number of delay taps and can be converted to UI by dividing by the value 32 (decimal). See Table 74 for per-lane addresses.

The SerDes eye metrics registers are runtime registers and require that the Link Configuration ERI and INIT Continue ERI be run prior to reading or writing the registers or ERI in this section.

Table 73: SerDes RX Eye Width

Name	Start Bit	Size	Type	Reset Value	Description	Notes
Reserved	0	10	RO	0x000	Reserved	1
RX eye width	10	5	RO	0x00	The latest available eye width measurement	
Reserved	15	17	RO	0x00000	Reserved	1

Note: 1. Value may be nonzero during runtime.

Table 74: SerDes RX Eye Width Addresses

Individual lane addresses to be read for SerDes RX eye width

Lane	Link 0	Link 1	Link 2	Link 3
Lane 0	0x2000AA	0x2100AA	0x2200AA	0x2300AA

Table 74: SerDes RX Eye Width Addresses (Continued)

Individual lane addresses to be read for SerDes RX eye width

Lane	Link 0	Link 1	Link 2	Link 3
Lane 1	0x2000EA	0x2100EA	0x2200EA	0x2300EA
Lane 2	0x2001AA	0x2101AA	0x2201AA	0x2301AA
Lane 3	0x2001EA	0x2101EA	0x2201EA	0x2301EA
Lane 4	0x2002AA	0x2102AA	0x2202AA	0x2302AA
Lane 5	0x2002EA	0x2102EA	0x2202EA	0x2302EA
Lane 6	0x2003AA	0x2103AA	0x2203AA	0x2303AA
Lane 7	0x2003EA	0x2103EA	0x2203EA	0x2303EA
Lane 8	0x2008AA	0x2108AA	0x2208AA	0x2308AA
Lane 9	0x2008EA	0x2108EA	0x2208EA	0x2308EA
Lane 10	0x2009AA	0x2109AA	0x2209AA	0x2309AA
Lane 11	0x2009EA	0x2109EA	0x2209EA	0x2309EA
Lane 12	0x200AAA	0x210AAA	0x220AAA	0x230AAA
Lane 13	0x200AEA	0x210AEA	0x220AEA	0x230AEA
Lane 14	0x200BAA	0x210BAA	0x220BAA	0x230BAA
Lane 15	0x200BEA	0x210BEA	0x220BEA	0x230BEA

The SerDes RX eye height registers provide the height of the received data eye, as measured at the HMC receiver lane. The value read will be an eight-bit hexadecimal value representing the eye height in voltage taps and can be converted to mV by multiplying by 7.8 (decimal). See Table 76 for per-lane addresses.

Table 75: SerDes RX Eye Height

Name	Start Bit	Size	Type	Reset Value	Description	Notes
RX eye height	0	8	RO	0x32	The latest available eye height measurement	
Reserved	8	24	RO	0x000000	Reserved	1

Note: 1. Value may be nonzero during runtime.

Table 76: SerDes RX Eye Height Addresses

Individual lane addresses to be read for SerDes RX eye height

Lane	Link 0	Link 1	Link 2	Link 3
Lane 0	0x200093	0x210093	0x220093	0x230093
Lane 1	0x2000D3	0x2100D3	0x2200D3	0x2300D3
Lane 2	0x200193	0x210193	0x220193	0x230193
Lane 3	0x2001D3	0x2101D3	0x2201D3	0x2301D3
Lane 4	0x200293	0x210293	0x220293	0x230293
Lane 5	0x2002D3	0x2102D3	0x2202D3	0x2302D3

Table 76: SerDes RX Eye Height Addresses (Continued)

Individual lane addresses to be read for SerDes RX eye height

Lane	Link 0	Link 1	Link 2	Link 3
Lane 6	0x200393	0x210393	0x220393	0x230393
Lane 7	0x2003D3	0x2103D3	0x2203D3	0x2303D3
Lane 8	0x200893	0x210893	0x220893	0x230893
Lane 9	0x2008D3	0x2108D3	0x2208D3	0x2308D3
Lane 10	0x200993	0x210993	0x220993	0x230993
Lane 11	0x2009D3	0x2109D3	0x2209D3	0x2309D3
Lane 12	0x200A93	0x210A93	0x220A93	0x230A93
Lane 13	0x200AD3	0x210AD3	0x220AD3	0x230AD3
Lane 14	0x200B93	0x210B93	0x220B93	0x230B93
Lane 15	0x200BD3	0x210BD3	0x220BD3	0x230BD3

4.6.2 SerDes PRBS Verification ERI

While the Link configuration ERI is used to initiate a specific PRBS pattern to be generated and checked, additional steps must be taken to verify the synchronization of the PRBS checking circuitry and check the results. The PRBS Verification ERI Sequence is used to check whether each lane is synchronized and if there are errors beyond a bit error ratio (BER) of 1E^{-6}. Steps 3 through 6 can be truncated to read only the desired link's status by reading only the associated ERIDATA register after step 2 if desired. Each time the PRBS Verification ERI sequence is executed, the status will be reset after the values are read. Therefore, the results from the first time the PRBS Verification ERI is run after link training may not be valid, and one or more subsequent runs is required.

Table 77: PRBS Verification ERI Sequence

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
1	MODE WRITE REQUEST	ERIREQ	0x2B0004	[7:0]: Register request command	0x25: PRBS Verification	
				[15:8]: Type	0x00: Reserved	
				[21:16]: Target location	0x00: Ignored	
				[25:22]: Size	0x00: Ignored	
				[30:26]: External request status	0x0: Read only	
				[31]: Start	0x1: Inform HMC of valid request	
2	MODE READ REQUEST	ERIREQ	0x2B0004	Start bit polling sequence If [31] = 1, pause for 10µs and issue subsequent MODE READ REQUEST to ERIREQ If [31] = 0, continue to next step		

Table 77: PRBS Verification ERI Sequence (Continued)

Step	Command	Register Name	Register Address	Register Fields	Register Field Value	Notes
3	MODE READ REQUEST	ERIDATA0	0x2B0000	[31:0]: Link 0 Synchronization and Error Status	Bit 0: Lane 0 Synchronization Status Bit 1: Lane PRBS Errors ... Bit30: Lane 15 Synchronization Status Bit 31: Lane PRBS Errors	1
4	MODE READ REQUEST	ERIDATA1	0x2B0001	[31:0]: Link 1 Synchronization and Error Status	Bit 0: Lane 0 Synchronization Status Bit 1: Lane PRBS Errors ... Bit30: Lane 15 Synchronization Status Bit 31: Lane PRBS Errors	1
5	MODE READ REQUEST	ERIDATA2	0x2B0002	[31:0]: Link 2 Synchronization and Error Status	Bit 0: Lane 0 Synchronization Status Bit 1: Lane PRBS Errors ... Bit30: Lane 15 Synchronization Status Bit 31: Lane PRBS Errors	1
6	MODE READ REQUEST	ERIDATA3	0x2B0003	[31:0]: Link 3 Synchronization and Error Status	Bit 0: Lane 0 Synchronization Status Bit 1: Lane PRBS Errors ... Bit30: Lane 15 Synchronization Status Bit 31: Lane PRBS Errors	1

Note: 1. Error status is only valid when lane is synchronized.

4.6.3 Debug of General Errors

Table 78: First Error

Register address: 0x2B0005

Name	Start Bit	Size	Description
ERRSTAT	0	16	[15:0]: Error code of error that caused this logging.
RESERVED	16	16	RESERVED. May be non-zero.

Table 79: Last Error

Register address: 0x2B0006

Name	Start Bit	Size	Description
ERRSTAT	0	16	[15:0]: Error code of error which caused this logging.
RESERVED	16	16	Reserved. May be non-zero.

The following table contains the codes and descriptions of the first and last (most recent) errors that occurred after cold reset of the HMC.

Table 80: First and Last Error Register, Error Decode and Description

ERRSTAT	Description	Details and Recommended Steps
0x0001	Operating temperature threshold	The maximum junction temperature for normal operating conditions has been reached. Reduce the temperature of the device.
0x0002	TX token overrun	There are two possible causes of this error: 1. The HMC has detected that too many tokens have been returned from the host. Check that the host has not sent too many tokens through RTC. If operating in open loop response mode, tokens must not be sent to the HMC (per specification). 2. Undetected errors on incoming packets. Ensure that link CRC detection is properly set to 1 in the link configuration register.
0x001F	Multi-bit uncorrectable error (MUE)	A multi-bit uncorrectable error was detected in the DRAM data either in the read response packet data or during patrol scrubbing. An issue with the DRAM array can be repaired using the MUE repair feature of the vault controller. Manual repair can be used to repair this MUE as well. Refer to the Register Definitions section of this guide for details on both functions.
0x0020	Retry on Link 0	The HMC link logic has detected errors in the incoming packet. Check that the host has properly formed the packet. If error is occurring frequently, check the BER and begin debug for channel, power, power integrity, and clocking concerns.
0x0021	Retry on Link 1	
0x0022	Retry on Link 2	
0x0023	Retry on Link 3	
0x006X	CA parity retry limit (vault critical error)	X indicates vault number that generated this error. Check the integrity of the link, power delivery network, and PCB. Particularly focus on V_{DD} and V_{DDM} to ensure that they remain within specification. Run the BIST With Repair ERI to test and repair (as necessary) the through-silicon vias (TSVs) if this error occurs.
0x0070	Link retry failed Link 0	Link retries were attempted up to the specified limit (see Register Description section for details on setting retry limit), but retry was unsuccessful in these attempts. Check the integrity of the link, power delivery network, and PCB. Particularly focus on V_{TT}, V_{TR}, and V_{DD} to ensure that they remain within specification. Refer to the PDN guide in chapter 2 for details and references to related simulation models. An IBIS-AMI model is available for simulating the channel.
0x0071	Link retry failed Link 1	
0x0072	Link retry failed Link 2	
0x0073	Link retry failed Link 3	

Table 80: First and Last Error Register, Error Decode and Description (Continued)

ERRSTAT	Description	Details and Recommended Steps
0x0078	Input buffer overflow Link 0	There are two possible causes of this error: 1. The input buffer was overflown by the host. In this case, verify that the host is processing tokens properly such that the HMC input buffer tokens are available prior to sending request packets to the HMC. This will ensure that the HMC input buffer has sufficient depth for the incoming request packets. 2. Multiple in-band mode read requests were made prior to the previous mode read request completing. If this is the cause, ensure that only one mode read request takes place at a time.
0x0079	Input buffer overflow Link 1	
0x007A	Input buffer overflow Link 2	
0x007B	Input buffer overflow Link 3	
0x007D	Reliability temperature threshold	The device has reached a junction temperature which can cause long term reliability issues. Reduce the temperature of the device.
0x007E	Packet length greater than 9	LEN field indicated a request packet has greater length than HMC Gen2 (HMCC protocol 1.1) allows. This indicates a malformed packet (the CRC checked as valid). Check the host logic and ensure that the length is properly encoded for Gen2 devices.
0x007F	Internal fatal error	Indicates that the internal logic has encountered an error from which it can not recover without reset. Attempt to warm reset the device to recover. Ff this error does not clear or if it recurs, a cold reset is required.
0x0100	Timeout: DFE synchronization Link 0	Decision feedback equalizer (DFE) did not converge on link during link initialization. Ensure that valid signals are present and scrambled NULL packets are being sent prior to running INIT Continue. If so, investigate the channel integrity using the built in signal integrity test functionality of the device (that is, PRBS generation and check and/or eye metric tests) and/or other lab debug methods.
0x0101	Timeout: DFE synchronization Link 1	
0x0102	Timeout: DFE synchronization Link 2	
0x0103	Timeout: DFE synchronization Link 3	
0x0200	Timeout: link state Link 0	Link failed to reach an active link state prior to the completion of INIT Continue. Use the optional timeout feature if the host requires longer than ^{t}INITCONTINUE MIN to complete link training (see Register Description section for details). If link fails to train, verify the training sequence and the quality of the link channel.
0x0201	Timeout: link state Link 1	
0x0202	Timeout: link state Link 2	
0x0203	Timeout: link state Link 3	
0x0300	Timeout: signal detect Link 0	Link failed to detect signal on one or more receiver lanes in the link. Ensure that the host is transmitting valid signals (scrambled NULLs) prior to executing the INIT CONTINUE command per the HMC Gen2 data sheet specification for power-on and initialization.
0x0301	Timeout: signal detect Link 1	
0x0302	Timeout: signal detect Link 2	
0x0303	Timeout: signal detect Link 3	
0x8100	Timeout: NVM write	A write to the NVM was attempted but failed to complete.
0x8200	NVM log full	The NVM allocated for logging has filled. The device can typically run normally, but no further events will be logged.

Table 80: First and Last Error Register, Error Decode and Description (Continued)

ERRSTAT	Description	Details and Recommended Steps
0x8300	Timeout: PLL lock Link 0	One of the link's high speed serial phase-locked loops (PLLA/PLLB) failed to lock. Ensure that the reference clock is stable at power-up, as described in the HMC Gen2 data sheet. Verify that all PLLx power pins are connected to a valid supply.
0x8301	Timeout: PLL lock Link 1	
0x8302	Timeout: PLL lock Link 2	
0x8303	Timeout: PLL lock Link 3	
0x8400	Timeout: port ready Link 0	SerDes PLLs locked, but the SerDes itself did not indicate ready before the timout. Ensure that the power is stable on all HSSIO supplies (V_{TT}, V_{TR}, V_{DDPLLx}, and V_{DD}).
0x8401	Timeout: port ready Link 1	
0x8402	Timeout: port ready Link 2	
0x8403	Timeout: port ready Link 3	
0x8500	Timeout: internal DRAM error	Ensure that the power-on sequence has been followed and power supplies are stable and within specifications.
0x86XX	Repairs unavailable	No additional DRAM repair resources remain. A repair would have been executed at the following bank and vault, where XX is defined as: [7:4] = Vault, [3:0] = Bank.
0x8700	Processor internal error	The internal processor has encountered an unexpected and unknown error.
0x88XX	Processor internal error	The internal processor has encountered a serious error. XX is defined as: [6:0] = ERRSTAT
0x89XX	Processor internal error	The internal processor has encountered a serious error.
0x8AXX	Processor internal error	The internal processor has encountered a serious error.
0x8CXX	ECC error on internal SRAM	An ECC error was detected on an internal SRAM on the logic die. If a low rate of occurrence, this may be caused by soft errors.
0x8DXX	MUE and SBE in same packet	An MUE and a single-bit error (SBE) were detected in the same outgoing read response packet. Check voltage and PDN stability; verify that the specifications for delta-T_J and voltage change are being followed after INIT Continue. Run DRAM BIST after cold reset to attempt to repair DRAM.
0x8E00	Timeout: TSV portion of BIST	The TSV portion of built-in self test (BIST) timed out when the BIST ERI was executed.
0x8F00	Power loss detected during log write	Ensure that the shutdown ERI is used prior to removing power. NVM log corruption can occur if power is lost during the act of writing to NVM.
0x9100	Timeout: DFE convergence Link 0	The decision feedback equalizer (DFE) did not converge on link following emergence from down mode. Ensure that valid signals are present and scrambled NULL packets are being sent prior to exit of down mode.
0x9101	Timeout: DFE convergence Link 1	
0x9102	Timeout: DFE convergence Link 2	
0x9103	Timeout: DFE convergence Link 3	

Table 80: First and Last Error Register, Error Decode and Description (Continued)

ERRSTAT	Description	Details and Recommended Steps
0x9200	Timeout: link activate Link 0	Link failed to emerge from down mode and reach an active state.
0x9201	Timeout: link activate Link 1	
0x9202	Timeout: link activate Link 2	
0x9203	Timeout: link activate Link 3	
0x9300	Timeout: signal detect Link 0	Link did not detect a valid signal when emerging from down mode. Ensure that valid signals are present and scrambled NULL packets are being sent prior to exit of down mode.
0x9301	Timeout: signal detect Link 1	
0x9302	Timeout: signal detect Link 2	
0x9303	Timeout: signal detect Link 3	

4.6.4 Multi-Bit Uncorrectable Error

Multi-bit uncorrectable errors (MUEs) are found in the data as it is stored or read from the DRAM before it is reported to the host. If an MUE occurs, it will be reported in the response packet with the ERRSTAT, and the DINV flag will be set. Because the host utilizes tags to track transactions, the DRAM address at which the error occurred may be determined. The following registers store the addresses where the first and most recent (if applicable) MUEs occurred.

Table 81: First and Last MUE Registers

Register addresses: First MUE = 0x002C80B6; Last MUE = 0x002C80B7

Register Field	Description
MUE[0]	Reserved
MUE[19:1]	DRAM address [19:1]
MUE[23:20]	Bank address [3:0]
MUE[27:24]	Vault address [3:0]
MUE[30:28]	Reserved
MUE[31]	0x1: Entry is valid; 0x0: Entry is not valid

4.6.5 I^{2}C Bus Debug

The I^{2}C bus is an industry standard bus, but it may occasionally need to be debugged. If the HMC cannot be contacted via the I^{2}C bus, check the following:

- Are the CUB2:0 pins strapped according to the desired CUB address, per the HMC Gen2 data sheet?

If all pins are strapped to 0, then the base address will be [0x10][CUB[2:0]] = 0010 000, or 0x10. If you include the Wr/Rd bit as the LSB, making it 8-bits in length, this would be 0010 000 0 (0x20 for CFG Write), or 0010 000 1 (0x21 for CFG Read).

- Are REFCLKSEL and REFCLK_BOOT set for the appropriate REFCLK coupling and frequency options?
  Ensure that P_RST_N is asserted at power-up, de-asserted after REFCLK is valid, and de-asserted before trying to communicate via I^2C.
- Is the voltage 1.5V (V_{DDK}) and are the appropriate pull-up resistors being used?
- Is a slave on the bus pulling SCL and/or SDA LOW?
  A bus clear (9 clocks) can be issued to instruct a slave to release SDA if held LOW.
- Are the I^2C commands appropriate for HMC, as outlined in the HMC data sheet?
- Is the required 32b transaction with addresses present?
- Using an oscilloscope, are the V_{IH}/V_{IL} levels being met at the HMC (and other devices), or is there excessive capacitance on the lines causing slow rising edges?
  The latter could be a particularly issue if the I^2C bus traces are long or have many devices (loads).

If none of the potential issues listed above are the problem, the following tips may help with debugging:

- If possible, set the device driving the I^2C bus so that the SCK line will drive both HIGH and LOW levels, instead of driving HIGH through an open drain. This should be possible if the I^2C controller is implemented in a programmable I/O, such as that available in an FPGA package.
- 400 kHz can sometimes work better than lower frequencies (though 400 kHz is the maximum frequency).
- If illegal commands are issued (such as non-32b transactions), issuing a STOP will recover the HMC's internal state machine. Repeated START will not reset the HMC's I^2C controller.
- The I^2C should not be used to access the HMC during the warm reset completion time: doing so may result in a NAK on I^2C.
- The I^2C slave controller inside HMC can get stuck when unsupported commands are used, such as single-byte reads or when an extra STOP is issued before repeated START. After getting stuck from an event like this, the HMC I^2C slave controller may not respond to future correct I^2C commands until it is reset. After the HMC I^2C slave controller gets stuck, the following actions can clear/reset it:

  1. I^2C master should issue a STOP. However, if SDA is being held LOW by a slave, the master may need to issue a bus clear operation (9 clock pulses) and then the STOP.
  2. Toggling P_RST_N (however, this will reset the entire HMC configuration and stop all traffic on the links).

  Note that issuing START or repeated START without a prior STOP will not reset the HMC's slave I^2C controller, per the I^2C specification.
- If the I^2C bus does not have the correct pull-up resistor values, the I^2C bus may have corrupt transactions or no transactions. Calculate the pull-up resistor values, Rp MAX and Rp MIN, based on Chapter 7 of the I^2C bus specification:
  http://www.nxp.com/documents/user_manual/UM10204.pdf

  Expected typical values range from 1 kΩ to 10 kΩ.

Include level shifters appropriately in the sum of capacitance on the bus. Selectively adding capacitance may improve noise immunity. Series resistance may help for some noisy systems.

Micron recommends simulating the I^{2}C bus to ensure signal integrity at the device pins, and to potentially help optimize the pull-up resistor values. The HMC Gen2 IBIS model contains the SCL and SDA pins, and is located on Micron's website: https://www.micron.com/~/media/documents/products/sim-model/hmc/gen2/signal-models-ibis-and-s-parameter/gpio_ibis/hmc_gen2_gpio_ibis.zip

4.6.6 CUB ID Header Field Debug

The CUB field in the request packet headers from the host must always be used correctly, even in non-chained topologies. If the CUB field is set improperly by the host, the HMC may report an internal fatal error (ERRSTAT = 0x7F), and assert the FERR_N pin.

Setting the CUB field in the request packet header is required, but the CUB ID value can be overridden in the cube to match what's in the request header generated by the host, as outlined in the Register Definitions section of this document.

4.6.7 P_RST_N and TRST_N Debug

P_RST_N must be asserted and held LOW during power-up of all HMC power supply rails, and only released (de-asserted) in accordance with the "Power-On and Initialization" section in the HMC data sheet. If the correct procedures are not followed, there can be no guaranteed functionality of the HMC, and it could partially or totally fail to initialize.

TRST_N must be asserted LOW through power-up or cold reset (P_RST_N assertion). Failure to do so could cause false DRAM failures, and/or could cause the die temperature sensors to falsely calibrate. The latter can cause issues that include false temperatures reported during Temperature Monitor ERIs and/or false temperature threshold warnings/errors reported in error response packets (either false high-temperature warnings, or not receiving the high-temperature warnings/alarms when they should be given). If boundary scan testing will be run while the rest of the HMC device is in a reset state, the following sequence is recommended:

1. Power-up or cold reset normally
2. De-assert P_RST_N HIGH followed by TRST_N
3. Re-assert P_RST_N LOW
4. Run boundary scan

5 Revision History

5.1 Rev. A – 7/16

- Initial release combining previous documents.
- This document supercedes
  - Register Addendum
  - TN-43-03: HMC (Gen2) Routing Guidelines
  - TN-43-04: HMC Gen2 PDN Requirements
  - TN-43-05: HMC Thermal Management
  - TN-43-07: Optimizing HMC Performance
  - TN-43-08: HMC Pin Connections Guide
  - TN-43-09: HMC Gen2 Thermal Modeling Guide
- Changed nomenclature for static repair and dynamic repair to MUE repair and SBE repair.
  - Explaining the events that cause a MUE repair to get logged to NVM
  - Explaining the events that cause a persistent permanent SBE repair to get logged to NVM.

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-4000
www.micron.com/products/support Sales inquiries: 800-932-4992