Cache Coherent Interconnect for Accelerators

CCIX® Base Specification Revision 2.0 Version 1.0

August 24, 2022

LEGAL NOTICE

This **CCIX Base Specification Revision 2.0 Version 1.0** (this "document") is proprietary to CCIX Consortium, Inc. (sometimes also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE CCIX CONSORTIUM, INC. MEMBERS:

Members of CCIX Consortium, Inc. ("CCIX Member(s)") may use and/or implement this **CCIX Base Specification Revision 2.0 Version 1.0** subject, however, to the CCIX Member's compliance with all of the terms and condition of the Company's Intellectual Property Policy, Bylaws, and all other Company policies and procedures ("CCIX Governing Documents") and the CCIX Member's Participation Agreement.

Each CCIX Member hereby agrees that its use and/or implementation of this **CCIX Base Specification Revision 2.0 Version 1.0** and/or any of the specifications described herein are subject to the following additional terms and conditions:

(i) The Company takes no position regarding, and each CCIX Member is solely responsible for determining on its own, the existence, validity, and/or scope of any intellectual property rights or any other rights (including without limitation any "Essential Claims" under the Company's Intellectual Property Policy) that any third party (including without limitation any other CCIX Member) may own or otherwise hold which may (or may not) pertain to or cover any implementation or other use of this **CCIX Base Specification Revision 2.0 Version 1.0** or any specifications described herein.

(ii) Each CCIX Member is solely responsible for:
   (a) determining whether any license or other consent (including without limitation any "FRAND License" under the Company's Intellectual Property Policy) from any third party (including without limitation from any other CCIX Member) is needed to implement or otherwise use this **CCIX Base Specification Revision 2.0 Version 1.0** or any of the specifications described herein; and
   (b) Negotiating and obtaining, for itself, any such license or other consent (including without limitation any "FRAND License" under the Company's Intellectual Property Policy) from any and all such third parties (including without limitation from other CCIX Members).

NOTICE TO NON-MEMBERS OF CCIX CONSORTIUM, INC.:

If you are not a CCIX Member and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge CCIX Consortium's Inc.'s copyright ownership of this document. The proper copyright citation or reference is as follows: **"©2016-2022 CCIX CONSORTIUM, INC. ALL RIGHTS RESERVED."** When making any such citation or reference to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of CCIX Consortium, Inc. Nothing contained in this document shall be deemed as granting (either expressly or impliedly) to any party that is not a CCIX Member: (ii) any kind of license to implement or use this document or any specifications described therein or any of its contents, or any kind of license in or to any other intellectual property owned or controlled by CCIX Consortium, Inc., including without limitation any trademarks of CCIX Consortium, Inc.; or (ii) any benefits and/or rights as a CCIX Member under any CCIX Governing Documents.

If you are not a CCIX Member but still elect to implement this document or any of the specifications described herein, you are hereby given further notice that your election to do so does not give you any of the rights and/or protections of CCIX Members, including without limitation any of the rights and/or protections of CCIX Members under the Company's Intellectual Property Policy or other CCIX Governing Documents.

LEGAL DISCLAIMERS FOR ALL PARTIES:

THIS DOCUMENT AND ALL SPECIFICATIONS AND/OR OTHER CONTENT PROVIDED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CCIX CONSORTIUM, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT. In the event this document makes any references (including without limitation any incorporation by reference) to another party's ("Third Party") content or work, including without limitation any specifications or standards of a Third Party ("Third Party Content"), you may need to independently obtain a license or other consent from that Third Party in order to have full rights to implement or use that Third Party Content.

Table of Contents

List of Figures

List of Tables

Chapter 1. Document Overview

1.1 Specification Objective

The current document provides all aspects of the specification that impact hardware design.

1.2 Terms and Acronyms

The following is a list of terms and acronyms that are critical for broader understanding of the key aspects of the overall specification. The terms that are used within the context of the detailed specification of a chapter are not listed here and instead are listed within the chapter where they are used.

1 **Port** – Port is associated with physical pins and has two sub-layers: a) CCIX® layer Port, referred to as CCIX Port and b) Transport layer Port, referred to as Transport Port.

   a. CCIX Port – CCIX Port acts as an ingress and egress of the CCIX Protocol Layer messages from a given CCIX Device. Each CCIX Port must have an associated Transport Port as well.

   b. Transport Port – A Transport Port is the Controller that acts as the gateway for ingress and egress of transport layer packets. Transport Port in this version of CCIX specification is based on PCI Express® (PCIe). Transport Port is referred to as PCIe Port for PCIe-based CCIX transport.

2 **CCIX Link** – CCIX Link is a logical connection between a pair of CCIX ports. Each CCIX Link manages credits used for CCIX message communication.

   a. For each CCIX Link there may be one or more CCIX layer Credit Classes for CCIX layer message communication.

3 **PCIe Link** – PCIe Link is a physical connection between the PCIe Ports. PCIe Link conforms to the Link definition in the PCIe specification.

4 **Transport Link** – Transport Link is a generic term to refer to a physical link in the Transport Layer specification used to overlay the CCIX Protocol Layer.

5 **CCIX Components** – CCIX Components are architected building blocks required to define the CCIX coherency protocol. The CCIX protocol defines the interaction between these building blocks to achieve data sharing while conforming to memory consistency requirements.

6 **Acceleration Function** – Acceleration Function is an implementation specific source of a memory access request for coherency protocol that are represented by a CCIX Request Agent for communication by CCIX coherency layer.

7 **CCIX Device** – A CCIX Device is a physical entity consisting of one or more CCIX Components that conform to the CCIX protocol. A CCIX Device must have at least one Port.

8 **CCIX Agent** – Any CCIX protocol Component that can be a source or a target of a transaction is referred to as a CCIX agent. A CCIX Agent can be further classified as one of the following agent types:

a. **Request Agent** – A Request Agent (RA) is a CCIX Agent that is the source of read and write transactions. Each of the CCIX RAs may have one or more internal initiators, Acceleration Functions (AFs).

b. **Home Agent** – A Home Agent (HA) is a CCIX Agent that manages coherency and access to memory for a given address range. An HA manages coherency by sending snoop transactions to the required Request Agents when a cache state change is required for a cache line. Each CCIX Home Agent acts as a Point of Coherency (PoC) and Point of Serialization (PoS) for a given address.

c. **Subordinate Agent** – CCIX enables expanding system memory to include memory attached to an external CCIX Device. When the Home Agent resides on one chip and some or all of the physical memory associated with the Home Agent resides on a separate chip, the resulting new architectural component (the expansion memory) is referred to as Subordinate Agent (SA). A Subordinate Agent is never accessed directly by a Request Agent. A Request Agent always accesses a Home Agent, which in turn accesses the Subordinate Agent.

d. **DVM Agent** – A DVM Agent (DA) receives and processes DVM requests and if required forwards them to other DVM Agents. There can be at most only one DVM Agent in a chip.

e. **IO Coherent Request Agent (RAI)** – RAI is a type of Request Agent that can request access to locations, but will not issue transactions that allow the associated cache line to be allocated in a coherent cache.

f. **Error Agent** – An Error Agent (EA) receives and processes protocol error messages. The protocol error messages are sent from CCIX components.

9 **CCIX Functional Block** – CCIX Functional Blocks are building blocks that are needed to define functionality of CCIX Components; e.g., System Address Map (SAM).

10 **CCIX switch** – CCIX switch is a CCIX Device consisting of two or more CCIX Ports capable of CCIX Port to Port forwarding. A CCIX switch may be embedded in a CCIX Device with Agents, or it may be a CCIX Device which has no CCIX Agents.

11 **Packet** – Packet is the unit of transfer that is routed independently.

12 **SAM** – System Address Map.

13 **G-SAM** – Global System Address Map. This is the global view of the System Address Map of memory accessed by all Home and Subordinate Agents.

14 **G-RSAM** – Global Request Agent System Address Map. This is the global view of the System Address Map of memory accessed by all Request Agents.

15 **G-HSAM** – Global Home Agent System Address Map. This is the global view of the System Address Map of memory accessed by all Home Agents.

16 **RSAM** – Request Agent's view of SAM for a given CCIX Device.

17 **HSAM** – Home Agent's view of SAM for a given CCIX Device for the Requests to Subordinate Agents issued by HA.

18. **PSAM** – SAM associated with a Port for outbound requests to allow mapping of an address range to a carrier CCIX Link for this request.
19. **PHY** – Physical Layer of the interface.
20. **EDR** – Extended Data Rate. The data rates for the PHY that are in addition to the standard PHY speeds defined by the *PCI Express Base Specification* (see Reference Documents).
21. **ESM** – Extended Speed Mode – ESM is a mechanism to allow PCIe Link speed transitions between the standard PCIe speeds and EDR speeds.
22. **DW** – Double Word – CCIX defines Double Word as a 4B-aligned 4B data element.
23. **DWord** – DWord is another representation of DW with identical meaning.
24. **Snoop Hazard** – A condition when a Snoop Request and a copyback request traveling in opposite directions are to the same cache line and have the same Target Agent and Source Agent respectively.
25. **Switches** – CCIX topology supports multiple versions of a switch, these being:
    a. Transport Switch – PCIe switch, where messages are sent between endpoints using the PCIe-compatible packet format.
    b. Protocol Aware Switch (or CCIX Switch) – A switch that can use the optimized CCIX packet format and is aware of the routing of agents; capable of merging of snoop responses for snoop broadcast messages; detecting Snoop Hazards, and managing the coherence conflict.
26. **Message Type** – CCIX uses six Message Types, these being Memory Request, Snoop Request, Misc (credited), Memory response, Snoop response, Misc (Uncredited).
    a. Memory Request, Snoop Request, and Misc (credited) are Credited message types.
    b. Memory Response, Snoop Response, and Misc (uncredited) message types require NO credits to be sent.
27. **CCIX Consortium Identifier (CCID)** – The 16-bit common identifier in CCIX DVSEC structures and CCIX PCIe Compatible Header Protocol Messages.
28. **CCIX Consortium Unique Value (CCUV)** – The hardware default 16-bit value in CCIX DVSEC structures and CCIX PCIe Compatible Header Protocol Messages. CCUV is also the CCID if the CCID has not been re-programmed with a new 16-bit value. CCUV is Decimal: 7724; Hexadecimal: 1E2C.

1.3 Reference Documents

PCI SIG documents, available at https://pcisig.com/specifications:

- [1]PCI Express Base Specification, Revision 5.0, Version 1.0
- PCI Express Card Electromechanical Specification, Revision 4.0
- PCI Local Bus Specification, Revision 3.0
- PCI-X Addendum to the PCI Local Bus Specification, Revision 2.0
- PCI Hot-Plug Specification, Revision 1.1
- PCI Standard Hot-Plug Controller and Subsystem Specification, Revision 1.0
- PCI-to-PCI Bridge Architecture Specification, Revision 1.2
- PCI Bus Power Management Interface Specification, Revision 1.2
- PCI Firmware Specification, Revision 3.0
- Address Translation Services, Revision 1.1
- PCI Express Base Specification, Revision 1.0a, PCI-SIG

OIF Common Electrical I/O (CEI) Electrical and Jitter Interoperability Agreement for 25+ GT/s I/O, IA # OIF-CEI-03.1, February 18, 2014

Unified Extensible Firmware Interface (UEFI) Specification, Version 2.7

[1] All references to the *PCI Express Base Specification* in this document are to the *PCI Express Base Specification*, Revision 5.0, Version 1.0, unless indicated otherwise.

Chapter 2. CCIX Overview

2.1 Introduction

The Cache Coherent Interconnect for Accelerators (CCIX®) enables a new model of data sharing between a server host, accelerators and external interface attached memory. CCIX extends the existing data sharing paradigm for processor-to-processor and processor-to-memory interactions, to include data sharing between processor-to-accelerator, and accelerator-to-accelerator. Additonally, data sharing between processors and accelerators is expanded to include data in memory attached to PCI Express® (PCIe) devices. Thus, CCIX enables data to be shared freely between process threads on Host processor cores and Acceleration Functions (AFs), without software intervention (OS or driver) to manage the data movement between host and accelerators, similar to how data is shared between process threads executing on processor cores.

In addition, CCIX enables expansion of system memory to include peripheral attached memory. Memory attached to different devices, besides the default of being managed by a driver, can be optionally seen as an OS managed pool of memory with non-uniform latencies, as is the case for existing Non-Uniform Memory Access (NUMA) architectures.

CCIX allows all data structures to be referenced by a common set of Virtual Addresses (VA) between all processing entities (hosts, accelerators and IO devices - accelerators and IO device are here on referred to as accelerators). In order to achieve these capabilities, CCIX expands the Shared Virtual Memory (SVM) model to include the following key attributes:

1 **Expanded System Memory:**

CCIX allows expanding the System Memory domain beyond host attached memory. The host memory manager can optionally allocate and manage peripheral attached memory in the same manner that host memory is allocated and managed, as part of system memory.

Thus, with CCIX Memory Expansion, a host can expand its memory capacity and/or support new memory technologies beyond that available with the host’s native memory capabilities. Note that the host’s view of peripheral attached memory is consistent with the existing view of memory in a multi-node host system, the Non-Uniform Memory Access (NUMA) memory model.

2 **Software transparent data movement based on a processor’s or accelerator’s application access patterns:**

CCIX enables hardware coherent caches in accelerators, and maintains a consistent view of shared data for both processor and accelerator accesses without explicit intervention from software.

In a CCIX system, caching allows implicit movement of shared data for further re-use or modification, based on the processor's or accelerator's access patterns, without the overhead of software migrating or maintaining multiple or modified copies between them.

There are use cases, for example the GPU-based co-processing model for numerical analysis, where OS (memory management) assisted implicit data movement is a suitable paradigm, and solutions do exist for these use-cases. Due to the software overhead involved, OS assisted data movement is typically only done for bulk data at a coarse-grained level. However, there are fine-grained data sharing use cases for example, the offloading of large graph transversal, that benefit from hardware managed cache coherency provided by CCIX. Thus, CCIX has an inherent ability to support both coarse-grained and fine-grained data sharing use cases.

3 **Application-managed movement of data from host to accelerator attached memory:**

CCIX enables applications to orchestrate data movement from one memory node to another without requiring any OS assistance, and while still preserving the SVM data sharing model. This allows a producer of data to place that data next to the computational entity (processor or an Acceleration Function) that is the consumer of the data. The optimal memory location for placement of the data can be determined from CCIX provided proximity information of memory to the computational entity.

The combined capabilities form the basis of a new model of seamless data sharing and data movement between host memory and processors, and peripheral attached memory and accelerators. Data either naturally migrates to the optimal location based on an application's access patterns on the CCIX cache coherent network, or the application itself can orchestrate data movement between accelerator-attached and host-attached memory, all without requiring either OS or driver support.

DVM support is provided for systems where components contain MMU functionality and are able to perform independent translation table accesses. Distributed Virtual Memory (DVM) transactions provide in band maintenance of translation caches. CCIX Rev 1.2 has introduced two types of DVM transactions, one which transports invalidations to remote translation caches and a second to synchronize the completion of all invalidation operations.

2.2 Topologies

The CCIX standard enables multiple platform topologies for connecting hosts to I/O devices, starting with support for tree-topologies that are common to the PCI Express® (PCIe) interconnect. CCIX also supports mesh, ring and other flexible topologies that are overlaid over the PCIe transport or, in the future, additional transports. The figures illustrate a few of the direct-attached topologies and illustrate a few of the mesh topologies that can be enabled by CCIX. For illustrations in Figure 2-1 and Figure 2-2 with a CCIX-linked host, the host has a computational entity (Processor or an Acceleration Function) as well as host attached memory.

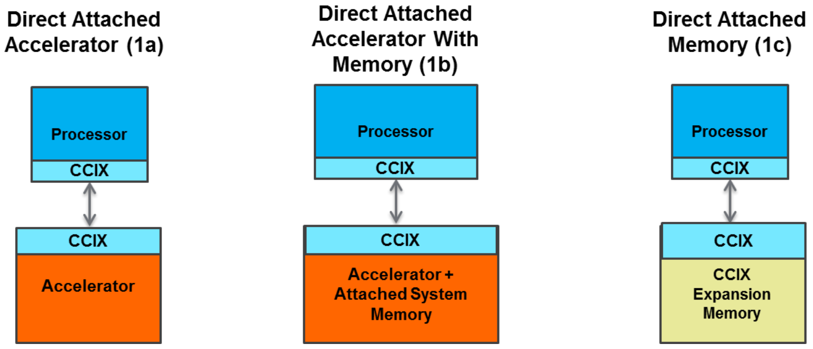

Direct Attached Accelerator (1a)
Processor
CCIX
CCIX
Accelerator
Direct Attached Accelerator With Memory (1b)
Processor
CCIX
CCIX
Accelerator + Attached System Memory
Direct Attached Memory (1c)
Processor
CCIX
CCIX
CCIX Expansion Memory

Figure 2-1: Example CCIX Direct-Attached topologies

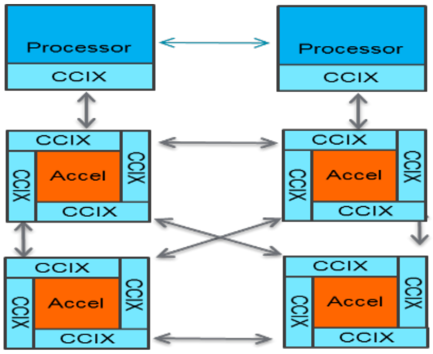


Figure 2-2: Example CCIX Fully Connected topologies

- Figure 2-1 (1a): Illustrates a platform with a CCIX-linked host direct-attached to a peripheral with a CCIX accelerator. This platform enables application acceleration with data sharing between the accelerator and processors, with the data residing only in host attached memory.
- Figure 2-1 (1b): Illustrates a platform with a CCIX-linked host direct-attached to a peripheral that has a CCIX accelerator as well as peripheral memory on the same device that can be host-mapped as part of system memory. This platform extends the application acceleration capabilities of platform 1a to enable sharing between the accelerator and processors such that the data can reside not only in host attached memory but also reside in locally attached accelerator memory as part of expanded system memory.
- Figure 2-1 (1c): Illustrates a platform with a CCIX-linked host direct-attached to a peripheral that has memory that can be made part of the host system map. On this platform, the host has expanded memory capabilities. The expanded memory capability may offer either expansion of memory capacity and/or allow integration of new memory technology beyond that of the host's native capabilities.
- Figure 2-2: Illustrates a platform with the CCIX-linked host and CCIX accelerators, all nodes in a CCIX mesh topology. With CCIX links between the accelerators as well as between the accelerators and the host, fine-grained data sharing can occur across multiple accelerators and processors. In addition, system memory may be expanded to include accelerator attached memory when present, such that the shared-data can reside in either host-attached memory or accelerator-attached memory, for optimized data movement.

2.3 CCIX Architecture Model

CCIX specifies a cache coherent interconnect that is overlaid on a PCIe transport. The overlay nature of the coherency interconnect makes it possible for CCIX coherency interconnect traffic to be carried over transports other than PCIe in the future. For the case of CCIX over a PCIe transport, CCIX also enhances PCIe to allow the carrying of coherency traffic in a manner that minimizes the impact to latency due to the PCIe transaction layer. In order to achieve the lower latency for CCIX communication, CCIX creates a light weight transaction layer that can independently co-exist alongside the standard PCIe transaction layer. Additionally, a CCIX link layer (that is overlaid on a physical transport like PCIe) ensures the availability of sufficient transaction channels necessary for deadlock free communication of CCIX protocol messages.

CCIX's PCIe-based Transport layer utilizes the common data link and Physical (PHY) layers to carry PCIe and CCIX transactions. As such, CCIX's transport definition naturally includes any enhancement done for data link, logical phy or physical layers by the PCI-SIG in future PCI Express Base specifications. For the PHY layer, unless explicitly stated otherwise in a given CCIX revision, the CCIX standard supports all standard data rates (including new data rates as and when they get defined) in the *PCI Express Base Specification*, and additionally defines two new data transfer rates—20GT/s and 25GT/s.

The CCIX model of overlaying the CCIX coherency interconnect over PCIe transport, and the associated enhancements to the PCIe Transport are shown in Figure 2-3.

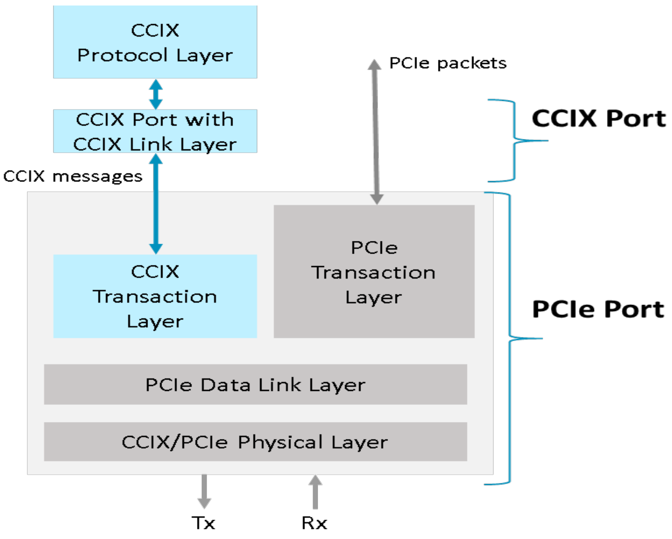


Figure 2-3: CCIX and associated PCIe layers

Chapter 4 provides the CCIX Transaction Layer specification.

The sections below define the key components and features of the CCIX architecture.

2.3.1 Components of the CCIX Architecture

CCIX defines the memory access protocol in terms of CCIX Components. Each CCIX device contains a selection of CCIX Agents, and communication Ports and associated Links with associated resources, as illustrated in Figure 2-4. An Agent can either be a Request Agent, a Home Agent, a Subordinate Agent, or an Error Agent. An Agent is identified within the protocol using an Agent ID value. A brief description of each agent type is given below:

- Request Agent – A Request Agent (RA) performs read and write transactions to addresses within the system. An RA can choose to cache the memory locations accessed.   Each of the CCIX RAs may have one or more internal initiators, referred to as Acceleration Functions (AFs).
- Home Agent – A Home Agent (HA) is responsible for managing coherency and access to memory for a pre-determined address range. It manages coherency by sending snoop transactions to the required Request Agents when a cache state change is required for a cache line. Each CCIX Home Agent acts as a Point of Coherency (PoC) and Point of Serialization (PoS) for a given address.
- Subordinate Agent – CCIX enables the expansion of system memory to include memory attached to peripheral devices. This scenario occurs when the Home Agent resides on one chip and some or all of the physical memory associated with the Home Agent resides on a separate chip. This architectural component (the expansion memory) is referred to as Subordinate Agent (SA). A Subordinate Agent is never accessed directly by a Request Agent. A Request Agent always accesses a Home Agent, which in turn accesses the Subordinate Agent. When the memory or memory interface for a Home Agent is

located on the same chip as the Home Agent, the Subordinate that provides the memory is not exposed as a CCIX Subordinate Agent.

- Error Agent – An Error Agent receives and processes protocol error messages. The protocol error messages are sent from CCIX Components.

Each Agent has an assigned AgentID value. Request Agents, Home Agents, Subordinate Agents and Error Agents are allowed to share the same AgentID value if they are located on the same chip. This ensures that ID routable CCIX protocol messages and error reporting messages can be routed by ID alone without needing the Agent type information.

The routing of messages between chips is done using Ports and Links as defined below.

- CCIX Port – A CCIX Port acts as the ingress and egress port for CCIX messages from a given CCIX Device. Each CCIX Port must have an associated Transport Port. A Transport Port is associated with a set of physical pins that carry both inbound and outbound traffic. A single CCIX Port can only receive one packet at a time from the off-chip interface, and can only transmit one packet at a time to the off-chip interface via the Transport Port.
- CCIX Link – A CCIX Link is defined as the connection between two CCIX Ports and has dedicated resources for communication. Credits are exchanged to indicate the level of resources that are available at the Receiver to accept messages.

Each CCIX Port is associated with a Transport Port. The Transport Port is limited to a PCIe Port. A CCIX Port is responsible for creating a PCIe compatible Vendor Defined Message (VDM) Transaction Layer Packet (TLP), or CCIX Packet with optimized header, which is compatible with the data Link Layer and Physical Layer that the traffic is moving on. CCIX protocol messages are carried in the payload of the PCIe compliant or optimized TLPs.

Each CCIX Port can communicate with one or more other CCIX Ports. To communicate with more than one other Port, the Port must contain multiple Links and an external Transport Switch. The external Transport Switch is required to select and route packets to the targeted CCIX Port via the associated Transport Port. The Transport Switch has no protocol-level awareness and its only function is to route packets between different CCIX Ports.

The physical layer associated with a CCIX Port can either be a PCIe Port with PCIe PHY characteristics or a CCIX PHY that extends the PCIe PHY to higher transfer rates, while maintaining backward compatibility with the PCIe PHY characteristics and lower transfer rates.

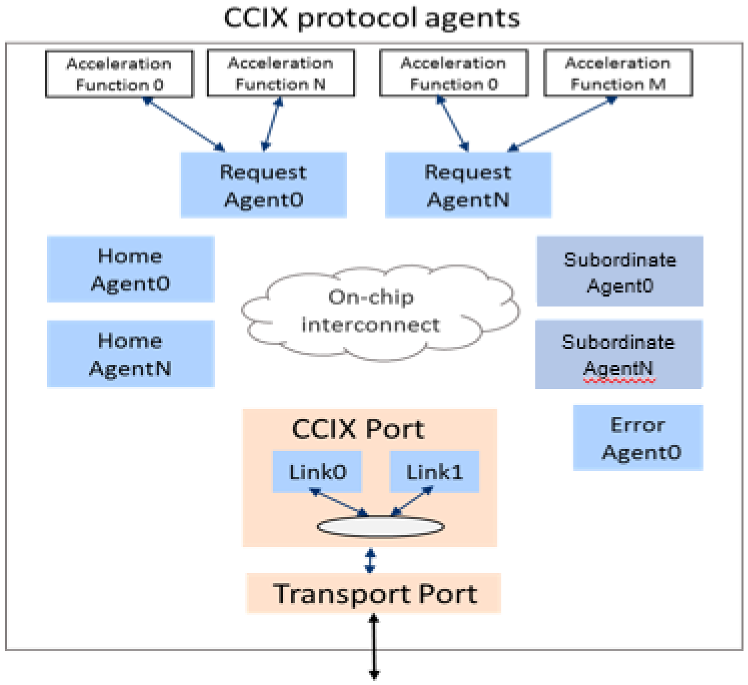


Figure 2-4: Components of the CCIX Architecture

2.3.2 Port Aggregation

CCIX can achieve higher bandwidth connectivity between two CCIX devices by optionally aggregating multiple CCIX ports. The CCIX Architecture defines a method to distribute memory access requests and snoops across multiple CCIX ports, where each CCIX Port maps to a PCIe controller when PCIe is used as a transport, to effectively achieve higher bandwidth between CCIX Agents. Port Aggregation is typically used where the throughput available from a single Port is not sufficient to meet the communication needs between two chips.

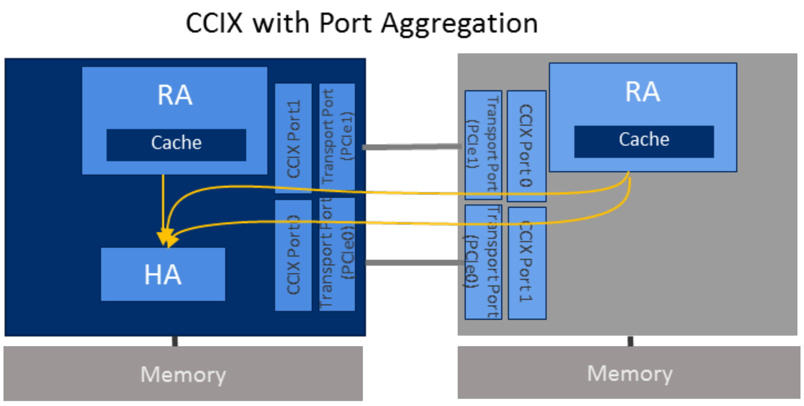


Figure 2-5: Load distribution for address based requests across multiple CCIX Ports

2.3.3 Subordinate Agent Device Aggregation

CCIX can achieve higher bandwidth memory expansion between Home Agent Device and Subordinate Agent by optionally aggregating multiple SA Devices. The CCIX Architecture defines a method to distribute memory access requests from a Home Agent across multiple CCIX ports, where the CCIX Ports map to more than one Subordinate Agent Device, to effectively achieve higher bandwidth between Home and Subordinate Agents. SA Device Aggregation, illustrated in Figure 2-6, is typically used where the throughput available from a single Subordinate Agent Device is not sufficient to meet the bandwidth needs of the Home Agent with memory expansion capability.

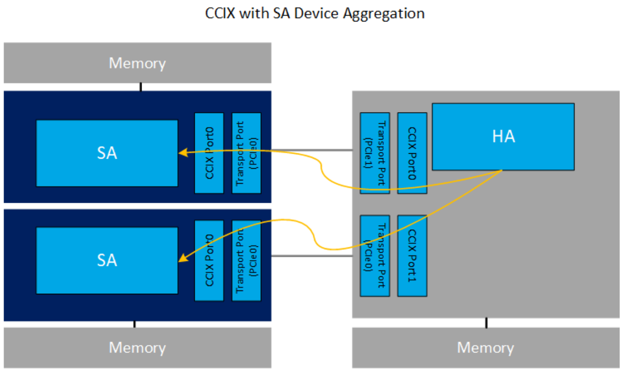


Figure 2-6: Load distribution for HA-to-SA memory expansion across multiple SA Devices

The Home Agent may support Port Aggregation or Subordinate Agent Device Aggregation, or both types of aggregation capabilities. For a particular topology, only one type of HA to SA Aggregation must be enabled, either HA to SA Port Aggregation, where the SA is on a single CCIX Device, or HA to SA Device Aggregation with multiple SA Device destinations. In that regard, the Home Agent can consider SA Device Aggregation the same as Port Aggregation except that the destination of the aggregated ports are on different destination devices. The memory pools of those destination devices are considered as part of one fused address map for the Home Agent with a distribution function between the HA ports also being the distribution function across aggregated SA devices, illustrated in Figure 2-6.

2.3.4 CCIX Extended Data Rate Physical Layer

The CCIX Transport is derived from the PCIe standard. As such, it supports all the interface speeds supported by PCIe. CCIX additionally defines 20GT/s and 25GT/s transfer rates to allow higher bandwidth connectivity between processors and accelerators.

Chapter 4 defines the logical PHY portion of the specification. The specification provides details of enhancements to the LSSTM state machine of logical PHY to support extended data rates.

Chapter 5 defines the electrical specification of the 20GT/s and 25GT/s Physical speeds.

2.4 CCIX Management Framework

CCIX Devices are discovered and managed as PCIe Devices. Components and their capabilities are discovered through a Designated Vendor-Specific Extended Capability (DVSEC) section in the PCIe configuration space. CCIX DVSEC carries a common CCIX Consortium ID (CCID) in the DVSEC Vendor ID field of the PCIe Configuration Header. The CCIX DVSEC defines capabilities and also provides the fields for controls and status.

CCIX coherency interconnect is managed through a standard CCIX driver. Acceleration Functions are managed as PCIe functions through vendor provided drivers. CCIX allows accelerator attached memory to become part of the overall system memory. It is then managed by the kernel memory manager as allocable system memory.

Chapter 6 provides a detailed definition of the Protocol and Transport Layer DVSEC.

2.5 RAS Architecture

CCIX defines a Server-class Reliability Availability and Serviceability (RAS) feature-set, including propogating and reporting uncorrected data errors only on consumption of the data.

The CCIX RAS architecture maintains reporting of PCIe transport errors through the Advanced Error Reporting (AER) mechanism, defined in the *PCI Express Base Specification*. Errors relating to the CCIX coherency interconnect are reported through a separate parallel mechanism to AER. The new mechanism for logging and reporting protocol errors is referred to as the Protocol Error Reporting (PER) mechanism. PER errors in the CCIX devices are logged in the Protocol DVSEC and are reported to a targeted Error Agent via a PER message routed to the AgentID associated with that Error Agent.

The detailed specification of CCIX RAS features and mechanisms is defined in 6.6.3.

2.6 Address Translation Service

CCIX leverages the Shared Virtual Memory (SVM) model for data sharing, where data-structures are shared based on Virtual Addresses (VAs). CCIX utilizes PCIe’s Address Translation Service (ATS) standard to allow CCIX Devices to map VAs to their associated Physical Addresses (PAs) as well as provide access controls, on a per page basis.

CCIX accelerators ensure that all memory requests from AFs pass through an Address Translation Cache (ATC) component of the ATS to map VA-to-PA, and to enforce access rights control.

To achieve the full extent of the capabilities offered by CCIX, described earlier in CCIX Architecture Model, and thereby achieve full flexibility in the data-movement architecture, CCIX requires a richer set of memory attributes than currently offered by the PCIe Gen 4 ATS specification. As a result, CCIX defines extensions to the ATS specification to acquire additional memory attributes on a per request basis. Chapter 8 provides details of additional memory attributes supported by CCIX request messages.

2.7 Signaling Hosts from Accelerators

CCIX uses the PCIe standard’s Message Signaling Interrupt (MSI/MSI-X) specification to signal events from Accelerators to a host processor. In Rev 1.0, there is no specific architected mechanism to signal the embedded functions in the CCIX device from the host.

The CCIX Software guide provides further details on the acceleration framework in the presence of CCIX.

2.8 Establishing Trust with a CCIX Accelerator

CCIX utilizes the PCIe ATS service to map request VA to corresponding PA and also to provide access control on a per page basis. All CCIX devices provide the following assurances:

1 Presence of trusted infrastructure layer – CCIX devices ensure that all requests from AFs through an address translation service to ensure that accesses rights control is applied for the accesses.

2 Device implements industry standard Secure boot mechanism – CCIX device ensure that firmware on device is trusted.

The Software Guide Revision 1.0 provides further details on the mechanism to establish that a CCIX device in use is indeed a genuine CCIX device implementing the required assurances for system robustness.

2.9 IO Coherent Request Agent

CCIX Revision 1.2 adds a formal definition of a simplified Request Agent, which is also referred to as an IO Coherent Request Agent (RAI). This type of Request Agent can request access to locations, but will not issue transactions that allow the associated cache line to be allocated in a coherent cache.

Implementation of a simplified Request Agent is possible with earlier revisions of the specification. Rev 1.2 adds a formal definition, as well as an option vector which permits the suppression of snoops to an IO Cohrerent Request Agent at source.

Chapter 3. Protocol Layer

The CCIX 2.0 Protocol Layer Chapter 3 is covered in a separate document.

Chapter 4. CCIX Transport Layer

4.1 Introduction

The CCIX® Transport layer connects to the physical pins and is the carrier of the packets from one component to another. The relationship of the CCIX Transport layer to the upper layers of CCIX is shown in Figure 4-1. Each of these layers within the CCIX Transport are divided into two sections: one that processes outbound (to be transmitted) information and one that processes inbound (received) information.

CCIX uses packets to communicate information between the CCIX Data Link Layer and the CCIX Transaction Layer. As the transmitted packets flow downstream through the transaction, data link, and physical layers, they are extended with additional information necessary to handle packets at those layers. At the receiving side, the reverse process occurs and packets are transformed from their Physical Layer representation to the Data Link Layer representation and finally (for Transaction Layer Packets) to the form that can be processed by the Transaction Layer of the receiving device.

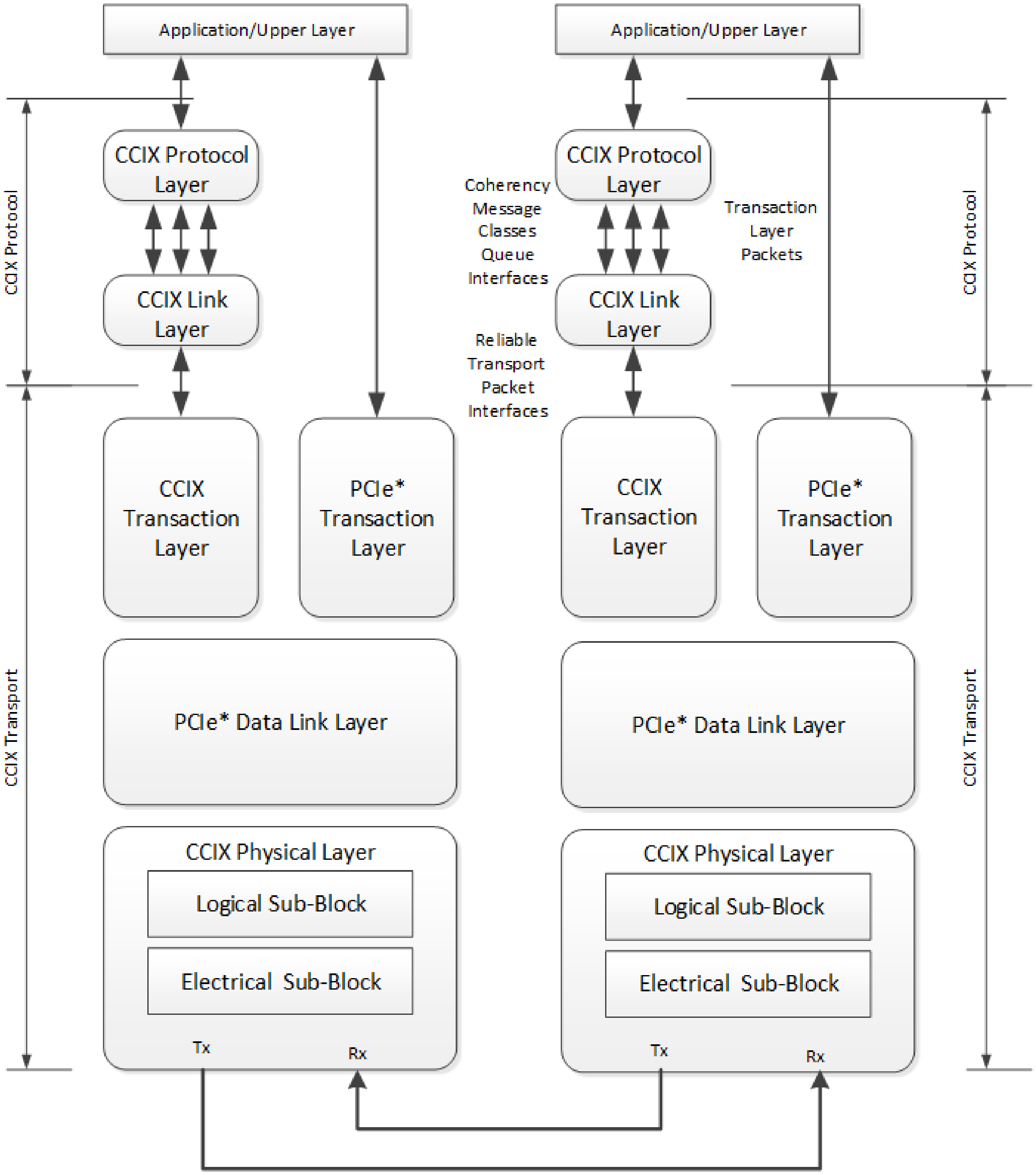


Figure 4-1: CCIX Layering Diagram

4.1.1 CCIX Transaction Layer

The CCIX specification defines a new transaction layer, referred to as the CCIX Transaction Layer. This transaction layer replaces the PCI Express® (PCIe) Transaction Layer in one of the PCIe Virtual Channels (VCs) in a multi-VC implementation. The CCIX Transaction Layer is a reduced PCIe Transaction Layer where only the following posted traffic is supported:

- Optimized TLPs
- PCIe Compatible TLPs

The CCIX Transaction Layer's primary responsibility is the assembly and disassembly of CCIX Transaction Layer Packets (TLPs).

- On the receive path, the CCIX Transaction Layer checks CCIX TLP Integrity, before forwarding the TLP to the CCIX Link Layer.
- For PCIe Compatible TLPs, the PCIe Transaction Layer checks specified in the *PCI Express Base Specification* are applicable.
- For Optimized TLPs, a new set of CCIX Transaction Layer checks are specified.

The CCIX Transaction Layer is also responsible for managing credit-based flow control for CCIX TLPs. On the receive path, posted flow control credits are returned for CCIX TLPs that pass data integrity checks and are forwarded to the Protocol Layer. In the transmit path, a credit gate is implemented to control flow of CCIX TLPs based on available posted credits. These posted credits are defined on a link-wide basis.

4.1.2 PCIe Transaction Layer

The CCIX specification does not modify the PCIe Transaction Layer.

The upper layer of the architecture is the Transaction Layer. The Transaction Layer's primary responsibility is the assembly and disassembly of Transaction Layer Packets (TLPs). TLPs are used to communicate transactions (e.g., read, write), as well as certain types of events. The Transaction Layer is also responsible for managing credit-based flow control for TLPs.

4.1.3 PCIe Data Link Layer

The CCIX specification does not modify the PCIe attirbutes of the Data Link Layer and utilizes it as is.

The PCIe Data Link Layer serves as an intermediate stage between the PCIe and the CCIX Transaction and the CCIX Physical Layers.

The primary responsibilities of the PCIe Data Link Layer include Link management and data integrity, including error detection and error correction.

4.1.4 CCIX Physical Layer

The CCIX Physical Layer includes all circuitry for interface operation, including driver and input buffers, parallel-to-serial and serial-to-parallel conversion, framing and deframing, PLL(s), and impedance matching circuitry. It also includes logical functions related to interface initialization and maintenance.

The CCIX Physical Layer exchanges packet information with the PCIe Data Link Layer in an implementation-specific format. This layer is responsible for converting packet information received from the PCIe Data Link Layer into an appropriate serialized format and transmitting it across the CCIX Link at a data rate and the PCIE width compatible with the device connected to the other side of the Link. In the receive direction, the CCIX Physical Layer converts serial stream received from the CCIX link partner into packet information for the PCIe Data Link Layer.

The CCIX Physical layer defines two PHY types. A CCIX component is required to support only one of the two PHY types.

- PCI Express capable PHY: This PHY type is compliant with all the requirements in the Physical Layer logical block chapter (Chapter 4) and the Electrical sub-block chapter (Chapter 8) of the *PCI Express Base Specification* (see Reference Documents).
- Extended Data Rate (EDR): This PHY type supports all the requirements of *PCI Express Base Specification*, with 16.0 GT/s capability, and extends the supported data rates to 20.0 GT/s and 25.0 GT/s. See Section 4.4.2.2 for more information.

4.2 Transaction Layer

4.2.1 CCIX Transaction Layer Architecture

The CCIX Transaction Layer shall contain at least one PCIe Virtual Channel (VC), used to exchange PCI Express TLPs. This must include VC0.

Where applicable (see section 4.2.2.3) the CCIX Transaction Layer also contains one CCIX Virtual Channel (CCIX VC), used to exchange CCIX TLPs. This cannot be VC0.

- The TC to VC mapping mechanism is used to steer TLPs to VCs.
- The Traffic Class (TC) label on received TLPs is used to steer traffic to various VCs.
- TC association for the CCIX VC is under the control of the system software.

Traffic is presented to the CCIX Transport Layer from the CCIX Protol Layer in one of two forms: 1) Packet or 2) Container.

In Non-Flit Mode, the CCIX Transaction Layer forms TLPs that are in either PCIe Comaptible TLP format or Optimized TLP format. In Flit Mode, the CCIX Transaction Layer forms Flits. See Section 4.2.2.4 for which of these are permitted for which CCIX Transport Layer output.

PCIe compatible TLP packets formed in a Non-Flit Mode and Flits formed in Flit Mode are Vendor Defined Type1 Messages, allowing CCIX TLPs to be forwarded by PCI Express compliant (i.e. CCIX unaware) Switches that implement a minimum of two VCs.

CCIX is not supported in any hierarchy containing a PCI Express to PCI/PCI-X Bridge.

CCIX operation is not impacted by the presence of retimers on the links in the hierarchy.

System software is not permitted to write register fields to change the configuration of, or the behavior of, the CCIX data path when it is enabled to transmit or receive CCIX traffic. Examples of bits that enable elements of the CCIX datapath are VCResourceControl.VCEnable when Set, and ESMControl.ESMEnable when Set.

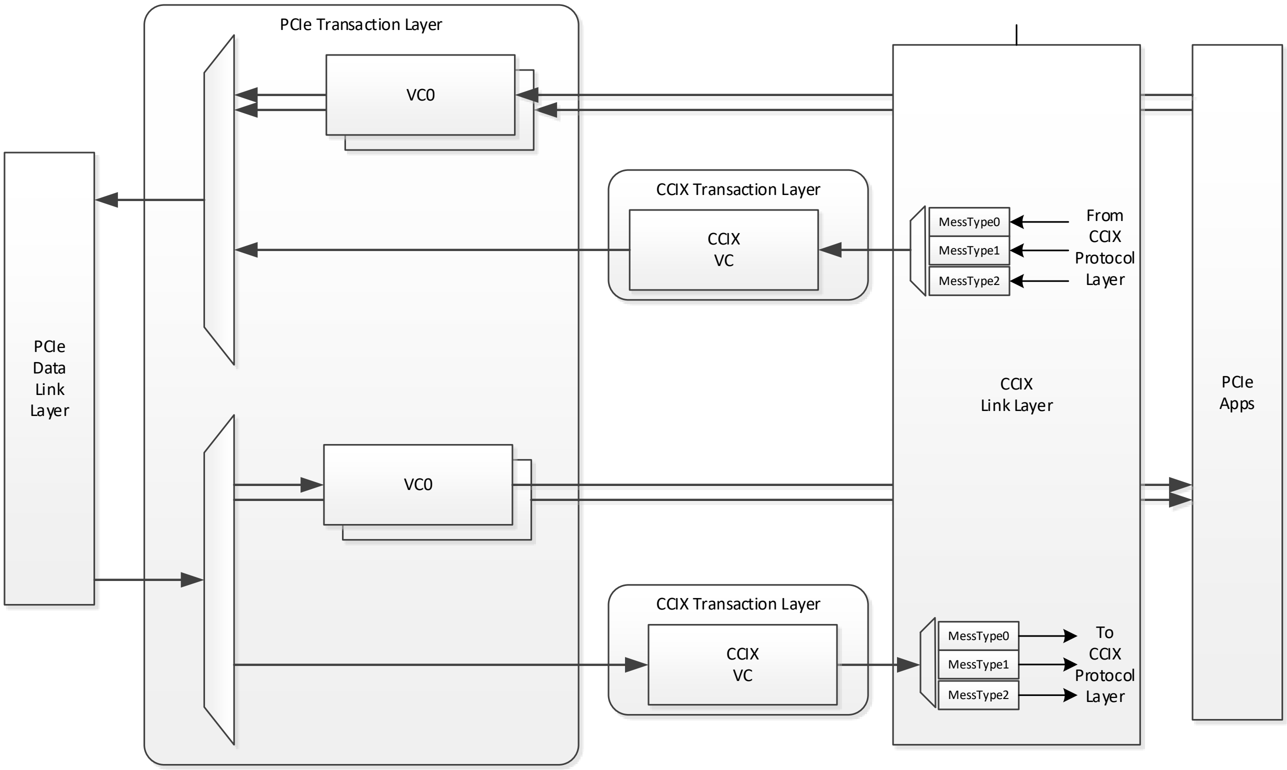


Figure 4-2: CCIX Transaction Layer Architecture

4.2.2 Transaction Layer Protocol - Packet Definition

CCIX Transaction Layer shall follow all requirements in the *PCI Express Base Specification* for the PCIe Transaction Layer (unless otherwise specified).

4.2.2.1 CCIX Transaction Layer Packets

Table 4 2 specifies which CCIX Protocol Layer data form/packet format comnbinations are applicable. required, and optional for each Physical Layer Data Rate and encoding. As specified by that table, components that negotiate to use Non-Flit Mode are not required to support the Flit Mode items required by that table, and components that negotiate to use PCIe Flit Mode are not required to support the Non-Flit Mode items required by that table.

4.2.2.1.1 PCIe Compatible TLP Format

The PCIe Compatible TLP format uses the Vendor-Defined Type 1 Message TLP using MsgD format, specified in the *PCI Express Base Specification*, Section 2.2.8.6 (see Reference Documents) and is used to form Non-Flit Mode TLPs. These TLPs belong to the Posted credit category.

Figure 4-3 shows the PCIe Compatible TLP.

Beyond the rules stated in that specification, the following rules apply to the formation of PCIe Compatible TLPs:

- The Message Routing field must be set to 010b – Routed by ID.

- Vendor ID field for all CCIX TLPs is equal to the CCIX Consortium ID (CCID).
  - Receivers are required to check this value before further processing a CCIX TLP.
- PCIe Compatible TLPs have a total length greater than 4 DW. Therefore, the TLP Fmt field is 011b (MsgD). Length [9:0] is the total number of DWs in the Vendor Defined Message payload.
  - All TLPs must have Length [9:0] in the range 01h <= Length [9:0] <= 7Fh.
  - When carrying a Container, Length must be 38h.
  - When TransactionLayerControl.OptionalLengthCheckEnable is Set, the receiver is permitted to optionally check Length [9:0] to be less than or equal the lesser of [(Device Control register Max_Payload_Size / 4) - 1] or 127. If a Receiver determines that a TLP violates this rule, the TLP is a Malformed TLP, and the error is reported by the Receiving Port.
- Bytes in TLP Header DW3 and in the data payload are specified in Chapter 3.
  - When carrying a Container, the Container starts in DW3.

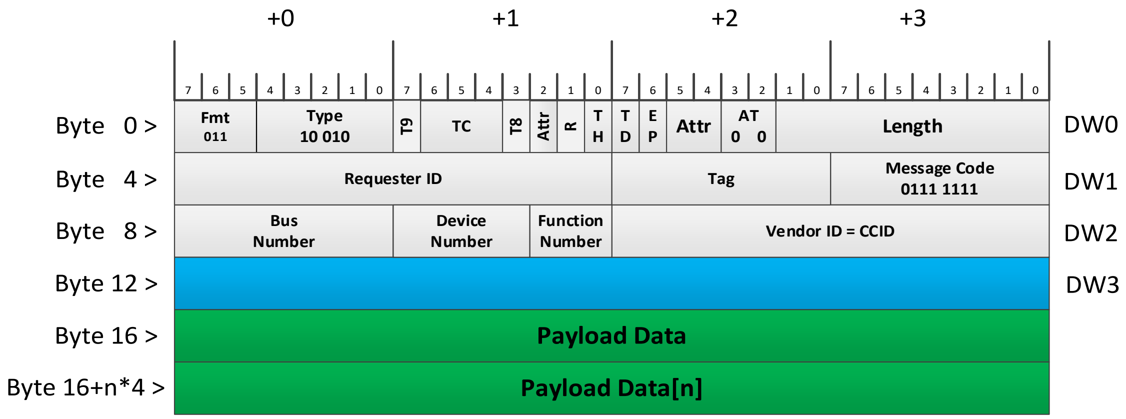


Figure 4-3: PCIe Compatible TLP format

4.2.2.1.2 Optimized TLP format

The Optimized TLP format is used to form Non-Flit Mode TLPs and is an optional feature meant to be used on CCIX links that advertise support for it (e.g., "Direct Attached" CCIX links); see Section 4.4.2.3 for more information. The Optimized TLP format is not compatible with the *PCI Express Base Specification*.

Figure 4-4 shows the Optimized TLP.

The following rules apply to the formation of Optimized TLPs:

- An optimized TLP data must be 4-byte naturally aligned and in increments of 4-byte DWs.
- An optimized TLP is composed of a 1 DW TLP Header section followed by a TLP Payload section that can contain up to 127 DWs.
- Bit 7 of byte 0, is always 0b.
- As indicated in Table 4-1, the 1-bit Type [0] field indicates the CCIX Hardware Specification revision.

Table 4-1: Type [0] Field Values

Type [0]	Optimized TLP Format
0b	CCIX Hardware Specification, Revision 1.0
1b	Reserved

- TC [2:0] – Traffic Class bits [6:4] of byte 1.
  - Transactions with a TC that is not mapped to any enabled VC in an Ingress Port are treated as Malformed TLPs by the receiving device.
  - For a Root Port, transactions with a TC that is not mapped to any enabled VC in the target RCRB are treated as Malformed TLPs.
  - For multi-function devices with an MFVC Capability structure, transactions with a TC that is not mapped to any enabled VC in the MFVC capability structure are treated as Malformed TLPs.
- Length [6:0] – Number of DWs in the TLP Payload section.
  - All TLPs must have Length [6:0] in the range 02h <= Length [6:0] <= 7Fh.
  - When carrying a Container, Length must be 38h.
  - When TransactionLayerControl.OptionalLengthCheckEnable is Set the receiver is permitted to optionally check Length [6:0] to be less than or equal to the lesser value of [(Device Control register Max_Payload_Size / 4) - 1] or 127. If a Receiver determines that a TLP violates this rule, the TLP is a Malformed TLP, and the error is reported by the Receiving Port.

In the Optimized TLP Header (DW0), all remaining bit locations are available for use by the CCIX protocol and are specified in Chapter 3. For an Optimized TLP, the CCIX Transaction Layer logic inspects Bit 7 of byte 0, and the Type [0], TC [2:0], and Length [6:0] fields within DW0 only.

In Optimized TLPs, bit locations in all DWs other than DW0 are available for use by CCIX protocol and are specified in Chapter 3.

All Optimized TLPs consume 1 Header Credit and Data Credits at the rate of (Length [6:0] - 2h) DW. Therefore, a TLP of Size = 3 (Length[6:0] = 000 0010b) shall consume 1 Header and 0 Data credits, while a TLP of Size = 4 (Length[6:0] = 000 0011b) shall consume 1 Header and 1 Data credits. The unit of Flow Control credit is 4 DW for Payload Data.

The bytes in TLP Header DW0 and in the data payload are specified in Chapter 3. When carrying a Container, the Container starts in DW1.

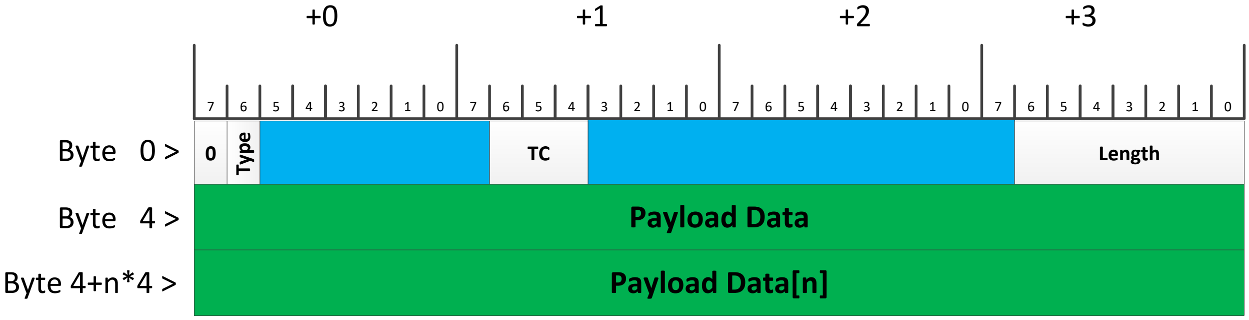


Figure 4-4: Optimized TLP Format

4.2.2.2 Container packing into PCIe 6.0 Flits

This section specifies the packing rules for 64 byte Containers in 256-byte physical layer PCIe 6.0 Flits. There are two formats of container packing; Type 0 and Type 1. Figure 4-6 defines the Type 0 format. Figure 4-7 defines the Type 1 format.

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	PCIe H0/B0	PCIe H0/B1	PCIe H0/B2	PCIe H0/B3	PCIe H1/B0	PCIe H1/B1	PCIe H1/B2	PCIe H1/B3	PCIe H2/B0	PCIe H2/B1	PCIe H2/B2	PCIe H2/B3	C0	C1	C2	C3
16	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
32	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
48	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
64	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	C0	C1	C2	C3
80	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
96	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
112	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
128	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	C0	C1	C2	C3
144	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
160	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
176	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
192	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	C0	C1	C2	C3
208	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
224	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	Dllp0	Dllp1	Dllp2	Dllp3
240	Dllp4	Dllp5	CRC0	CRC1	CRC2	CRC3	CRC4	CRC5	CRC6	CRC7	ECC0	ECC1	ECC2	ECC3	ECC4	ECC5

Figure 4-5: CCIX Container Packing Type 0

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	PCIe H0/B0	PCIe H0/B1	PCIe H0/B2	PCIe H0/B3	PCIe H1/B0	PCIe H1/B1	PCIe H1/B2	PCIe H1/B3	PCIe H2/B0	PCIe H2/B1	PCIe H2/B2	PCIe H2/B3	C32	C33	C34	C35
16	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
32	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	C0	C1	C2	C3
48	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
64	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
80	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
96	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	C0	C1	C2	C3
112	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
128	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
144	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
160	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	C0	C1	C2	C3
176	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19
192	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33	C34	C35
208	C36	C37	C38	C39	C40	C41	C42	C43	C44	C45	C46	C47	C48	C49	C50	C51
224	C52	C53	C54	C55	C56	C57	C58	C59	C60	C61	C62	C63	Dllp0	Dllp1	Dllp2	Dllp3
240	Dllp4	Dllp5	CRC0	CRC1	CRC2	CRC3	CRC4	CRC5	CRC6	CRC7	ECC0	ECC1	ECC2	ECC3	ECC4	ECC5

Figure 4-6: CCIX Container Packing Type=1

Packing rules:

1. The first 12 bytes of the flits are a VDM header as defined in the PCI Express Specification. The fields in this header are populated with values as defined in Section 4.2.2.1.1, with the exception that:
    a. the length field is always set to 224 bytes (56 DW's)
    b. CCIX H0 is placed in the VDM Header at PCIe H1/B2. This field is defined as 'Reserved' in the PCIe 6.0 spec. It is defined as 'Tag' in the PCIe 5.0 and earlier spec revisions. The field is valid for decode by CCIX when the header is a VDM and the VC field in the VDM header = the value of TransactionLayerCapabilities. CCIXVCResourceCapabilityIndex, and the VID field in the VDM header = CCID

2. CCIX H0 is defined as:

Bit #	7 (MSB)	6	5	4	3	2	1	0 (LSB)
Definition	R	R	R	R	Valid			Flit Type

   a. Flit Type identifies the format of the Flit

      0b = type 0

      1b = type 1

   b. The Valid field defines which Containers are populated or empty.

      a. For Type 0, the encodings are:

         000b = all Containers are empty

         001b = the Container starting at byte 12 is populated. The Containers starting at bytes 76, 140, and 204 are empty

         010b = the Containers starting at bytes 12 and 76 are populated. The Containers starting at bytes 140 and 204 are empty

         011b = the Containers starting at byte 12, 76, and 140 are populated. The Container starting at byte 204 is empty

         100b = the Containers starting at byte 12, 76, 140, and 204 are populated

         All other encodings are Reserved

      b. For Type 1, the encodings are:

         000b = all Containers are empty

         001b = the rest of the Container starting at byte 12 is populated. The Containers starting at bytes 44, 108, and 172 are empty

         010b = the rest of the Container starting at byte 12 and the Container starting at byte 44 are populated. The Containers starting at bytes 108 and 172 are empty

         011b = the rest of the Container starting at byte 12 and the Containers starting at bytes 44 and 108 are populated. The Container starting at byte 172 is empty

         100b = the rest of the Container starting at byte 12 and the Containers starting at byte 44, 108, and 172 are populated

         All other encodings are Reserved

3. If Type0OnlyPackingEnable.TransactionLayerControl is Set:
   a. packing with Type 1 formatting must not be used
   b. only the first 3 Containers are permitted to be valid
   c. the Container starting at byte 204 must be empty.
4. If Type0OnlyPackingEnable.TransactionLayerControl is Clear:
   a. a Type 1 Flit is permitted to be the next Flit after a Type 0 Flit only.
5. A Type 0 Flit is permitted to be the next Flit after a Type 0 Flit where the Container starting at byte 204 is empty
6. A Type 0 Flit must be the next Flit after a Type 1 Flit
7. A Data Link Protocol Error is reported when a Type 0 Flit is the next Flit received after a Type 0 Flit that has the Container starting at byte 204 populated
8. Each Flit contains 3 ½ containers
9. An empty Container is filled with 0’s, and Receivers ignore the empty Container’s contents.

4.2.2.3 Support for Multiple Protocol Messages in one CCIX TLP

CCIX enables minimizing overhead for transport of protocol messages by allowing packing of two or more protocol messages in one TLP. To the CCIX Transaction Layer, the packed messages still look like a single TLP with payload. Information for multiple protocol messages is encoded/decoded inside the CCIX Link layer, and is transparent to the Transaction Layer.

4.2.2.4 Use of PCIe Compatible or Optimized TLP formats or the Container

Table 4-2 specifies which TLP formats are applicable to which physical layer.

When using a PCIe Link running in Non-Flit Mode, all devices are required to support the PCIe Compatible TLP format for the CCIX VC. A CCIX device shall advertise if it supports the optional Optimized TLP Format feature, by setting the TransportDVSEC.TransactionLayerCapability.OptimizedTLPFormatSupported bit, shown in Table 6-73. If the OptimizedTLPFormatSupported bit is Clear, then the CCIX VC is only capable of transmitting and receiving PCI Express Compliant CCIX Compatibility TLPs.

Optimized TLPs may be exchanged on the CCIX VC only if both link partners support the optional Optimized TLP Format feature. To enable the Optimized TLP format on a CCIX link, the TransportDVSEC.TransactionLayerControl.OptimizedTLPGenerationReceptionEnable bit, shown in Table 6-74 must be Set on both link partners before software Sets the VC Enable bit in the CCIX VC, VC Resource Control register, see the *PCI Express Base Specification*, Section 7.15.7 (see Reference Documents).

A CCIX device using a PCIe Link running in Non-Flit Mode operates in one of two modes: a) PCIe Compatible TLP format enabled, or b) Optimized TLP format enabled.

The Container is the only form or format used when using a PCIe Link running in Flit Mode for the CCIX VC.

CCIX support is discovered and enabled through reporting and control registers described in Chapter 6.

For cases when CCIX devices using a PCIe Link are connected without a dedicated CCIX VC, CCIX traffic is not permitted to be enabled.

Optimized TLPs cannot be used if a CCIX unaware Switch is present on the Link.

For cases when CCIX devices using a PCIe Link are connected with a dedicated CCIX VC through CCIX unaware Switches, the CCIX Compatibility TLP format must be selected for carrying CCIX TLPs on the CCIX VC. Behavior is undefined if a CCIX device is connected through a CCIX unaware Switch and Optimized TLPs are used.

For the cases when CCIX devices using a PCIe Link running in Non-Flit Mode are connected with a dedicated CCIX VC without any intervening CCIX unaware Switches, Optimized TLP or PCIe Compatible TLP formats may be selected for carrying CCIX TLPs. In normal operation, only one of these formats is permitted to be used.

Message traffic, regardless of format (Optimized or PCIe Compatible), is excluded from the non-D0 TLP gating requirements in the *PCI Express Base Specification*, Sections 5.3.1.2 D1 State, 5.3.1.3 D2 State, and 5.3.1.4.1 $D3_{hot}$ State (see Reference Documents).

4.2.3 Identifying CCIX Coherency Traffic

Table 4-2 specifies how CCIX coherency traffic is identified based on the physical interface that is being run, defines what forms or Formats are applicable to which physical interface, and how the decoding of CCIX Messages at the Receiver are enabled.

Physical Interface	Means by which coherency traffic is identified	PCIe Compatible TLP Format Supported	Optimized TLP Format Supported	Container Supported	Enabling the decode
Any PCIe data rate running in Non-Flit Mode	The TC field is mapped to the CCIX VC	Required	Optional	For the PCIe Comaptible TLP Format, the Container will be carried[1] as specified in Section 4.2.2.1.1. If the Optimized TLP Format is supported, the Container is carried[1] as specified in Section 4.2.2.1.2.	The VC Enable bit in the CCIX VC, VC Resource Control register is Set.
Any PCIe data rate running in Flit Mode	The TC field is mapped to the CCIX VC	Not supported	Not supported	Required	The VC Enable bit in the CCIX VC, VC Resource Control register is Set.
Any other transport	Defined by that transport	Defined by that transport	Defined by that transport	The expectation is that the Container will be the most amenable	Defined by that transport
Note(s): 1. *A Container can be carried by this Format because it appears to the Transaction Layer to be a normal payload for that Format with a fixed length of 64 bytes and the TC field mapped to the CCIX VC.*					

Table 4-2: CCIX Coherency Traffic Differentiation Mechanisms

The following bullets are applicable when utilizing the CCIX VC only:

- Under normal operating conditions, the CCIX VC will only transmit and receive CCIX TLPs. All CCIX TLPs utilize the Posted Credit category. The CCIX VC implements a single First in First out (FIFO) Queue for CCIX TLPs.
- The CCIX VC shall advertise non-infinite Posted Flow Control Credit values. These Credit values determine the size of FIFO for the CCIX TLP Queue in the implementation.
- The CCIX VC shall advertise non-infinite Non-Posted Flow Control Credit values. Non-Posted TLPs are not anticipated on the CCIX VC, but some resources are needed in the event they are received and must be processed.
- The CCIX VC shall advertise infinite Completion Header and Data Credits. The CCIX VC shall not transmit Flow Control Update DLLPs for Completion credits.
- The CCIX VC shall follow all requirements in the PCI Express Base Specification for Posted Credit Handling by a Virtual Channel.
- The CCIX VC shall follow all requirements in the PCI Express Base Specification for Non-Posted Credit Handling by a Virtual Channel.

- The CCIX VC shall silently drop any received Flow Control Update DLLPs for Completion credits.

4.2.4 Handling of Received TLPs

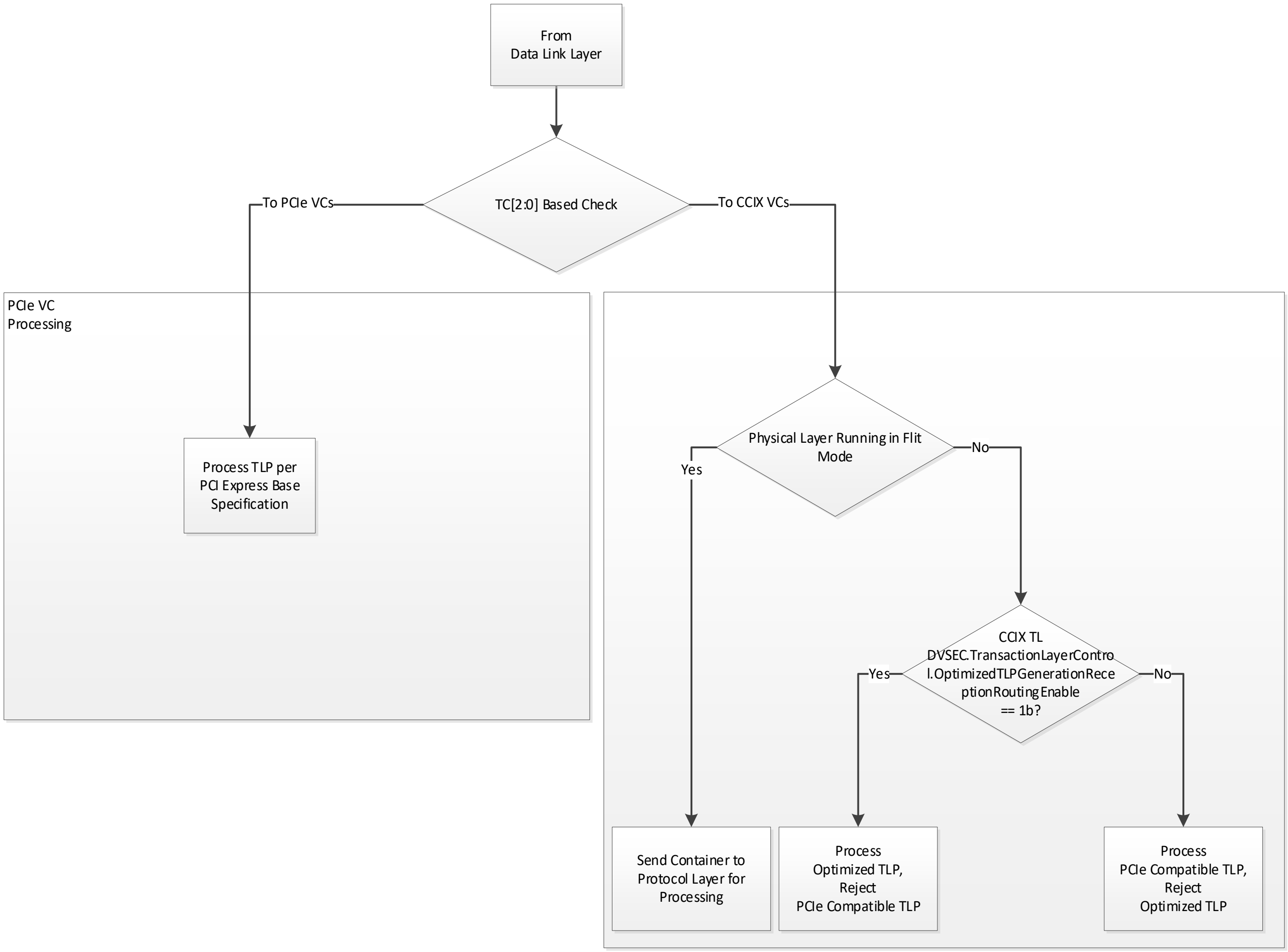


Figure 4-7: CCIX Transaction Layer Received TLP Processing Flow

Figure 4-7 shows the overall CCIX Transaction Layer received TLP processing flow. When running in Non-Flit Mode, CCIX Transaction Layer behavior is controlled by the TransactionLayerControl. OptimizedTLPGenerationReceptionRoutingEnable bit in Table 6-74.

- When Set:
  - The Transmitter generates CCIX TLPs in the Optimized TLP format, Switches in the path and Receivers accept Optimized TLPs.
  - All Receivers reject a PCIe Compatible TLP as a Malformed TLP.

- When Clear:
  - The Transmitter generates CCIX TLPs in the PCIe Compatible TLP format, Switches in the path and Receivers accept PCIe Compatible TLPs.
  - All Receivers reject an Optimized TLP as a Malformed TLP.

System Software (SSW) must program the TransactionLayerControl.OptimizedTLPGenerationReceptionRoutingEnable bit before CCIX traffic starts flowing and must not change value after it does. Behavior is undefined if the value of the OptimizedTLPGenerationReceptionRoutingEnable bit changes while CCIX traffic is flowing.

Figure 4-8 shows the steps in the processing of the Received CCIX TLPs.

These requirements are in addition to the standard processing flow for received TLPs, see the *PCI Express Base Specification,* Section 2.3, Handling of Received TLPs (see Reference Documents), and should be performed after the TLP checks specified in that section. Figure 4-8 is only for PCIe Compatible TLPs.

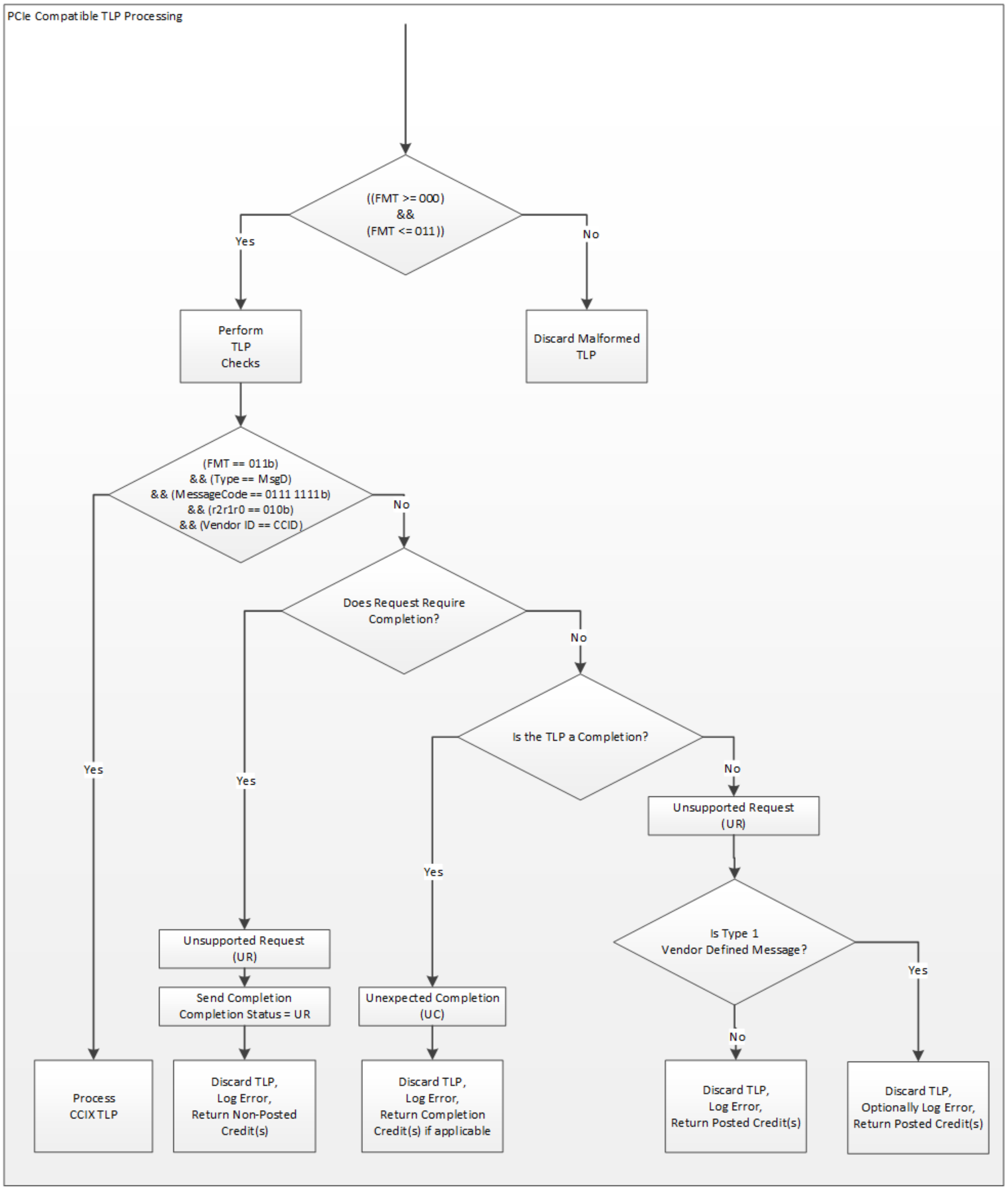


Figure 4-8: PCIe Compatible TLP Processing Flow

The CCIX TLP checks include TLP Format, TLP Type, TLP Length, TLP Message Code, TLP Message Routing field, and the Vendor ID field as shown in Figure 4-8. Only a valid CCIX TLP shall be forwarded to the CCIX VC storage.

If a TLP with a Prefix or undefined Format field is steered to a CCIX VC, it will be discarded as an "Invalid CCIX TLP", the port will log a Malformed TLP error and no flow control credit will be returned.

Other non-CCIX TLPs that are steered to a CCIX VC will be discarded (before they enter VC storage). An error will be logged and credit will be returned. Posted header and data credits shall be returned for all misdirected Posted TLPs, Non-Posted header and data credits shall be returned for all misdirected Non-Posted TLPs, and Completion header and data credits shall be returned for all misdirected Completion TLPs, if the component previously advertised non-infinite credits only. Since the CCIX VC advertises infinite Completion credits, these types of TLPs will not update credits. Type 1 Vendor Defined Messages on the CCIX VC that are not CCIX PCIe Compatible TLPs are permitted to be optionally logged as an error. The determination of whether to log or not is done by implementation specific means.

Figure 4-9 shows the processing flow for received Optimized TLPs. These requirements are in addition to the appropriate standard processing flow for received TLPs. See the *PCI Express Base Specification,* Section 2.3, Handling of Received TLPs (see Reference Documents) and should be performed after the TLP checks specified in that section.

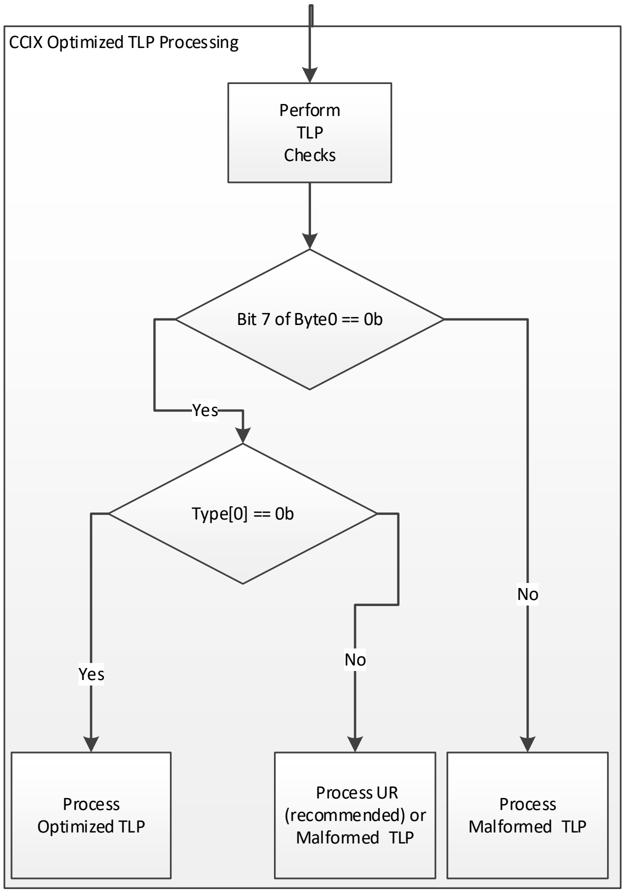


Figure 4-9: Optimized TLP Processing Flow

Optimized TLP checks include Bit 7 of Byte 0, Type [0] state (as shown in Figure 4-9) and optionally, valid Length[6:0] with the corresponding data payload. Bit 7 of Byte 0 must be zero. Receiving an Optimized TLP with a non-zero value for Bit 7 of Byte 0 is reported as a Malformed TLP. The Type bit must be zero. Receiving an Optimized TLP with a non-zero value for the Type bit is reported as a UR error (recommended) or a Malformed error (permitted). Only a valid Optimized TLP shall be forwarded to the CCIX VC storage.

4.2.5 Transaction Ordering Rules

The CCIX Transaction Layer shall follow all applicable requirements in the *PCI Express Base Specification* for Transaction Ordering rules.

Under normal operating conditions, the CCIX VC will only transmit and receive either PCIe Compatible TLPs or Optimized TLPs. These TLPs utilize the Posted Credit category.

On a CCIX VC, a Posted Request is not allowed to pass another Posted Request. All other ordering scenarios involving Non-Posted Requests or Completions are not expected to be handled by a CCIX VC.

4.2.6 Virtual Channel (VC) Mechanism

The CCIX Transaction Layer shall follow all the requirements in the *PCI Express Base Specification* for Virtual Channel Mechanism.

The VC Mechanism makes no distinction between a PCIe VC and a CCIX VC.

The CCIX VC is configured via the PCIe VC Resource Control Register associated with the VC Resource Capability register that is identified by the TransactionLayerCapabilities.CCIXVCResourceCapabilityIndex field.

It is required that a unique non-zero Traffic Class (TC) is mapped to the CCIX VC, and that CCIX TLPs receive differentiated QoS with respect to PCIe TLPs by use of this unique TC throughout the PCIe hierarchy of connected components as described in the Implementation Note “TC-VC Map in a PCIe Interconnect Hierarchy with PCIe and CCIX Devices”.

IMPLEMENTATION NOTE

VC Arbitration – Arbitration between VCs

Strict Priority for CCIX VC: Software may configure the supported VCs into two priority groups. The lower priority group may contain all non CCIX VCs, while the upper priority group contains just the CCIX VC (highest VC ID), enabling minimal latency for CCIX transactions. There is however a potential danger of bandwidth starvation with such a strict priority scheme.

It is therefore recommended that CCIX implementations set (Extended VC Count == Low Priority Extended VC Count) in the VC Extended Capability, Port VC Capability Register 1, so that all VCs (including the CCIX VC) are governed by the VC arbitration schemes indicated by the VC Arbitration Capability field. The VC Arbitration Capability may support additional arbitration discipline that on average allows achievement of strict priority, while at the same time, allows starvation free operation for lower priority traffic. The choice of arbitration scheme supported is an implementation choice.

Please refer to the *PCI Express Base Specification*, Section 6.3.3 (see Reference Documents) for more information regarding VC Arbitration.

IMPLEMENTATION NOTE

TC-VC Map in a PCIe Interconnect Hierarchy with PCIe and CCIX Devices

In a PCIe hierarchy containing both PCIe and CCIX Devices, CCIX traffic must receive differentiated QoS vs. PCIe traffic, and not have any ordering dependencies with PCIe traffic through the PCIe hierarchy. This can be achieved by associating CCIX traffic to a unique TC that is carried on its own VC resource, which is not shared with other TCs throughout the path between the CPU and the CCIX device. Figure 4-10 shows an example of a possible TC to VC mapping.

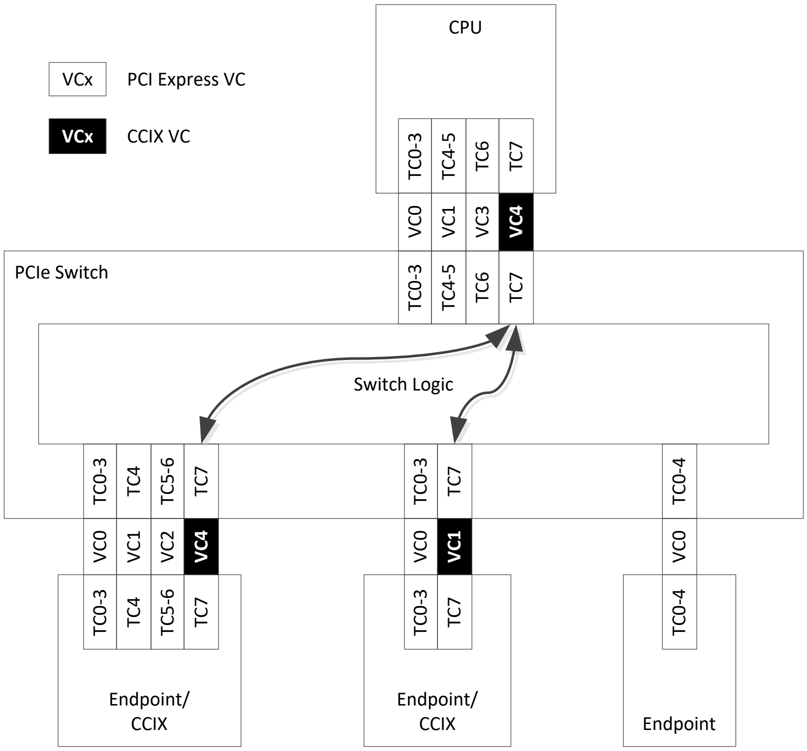


Figure 4-10: Possible TC to VC Mapping

4.2.7 Transaction Layer Flow Control

The CCIX Transaction Layer shall follow all the requirements in the *PCI Express Base Specification* for Flow Control with exceptions that apply only to the CCIX VC behavior.

The CCIX VC shall initially advertise infinite Completion Credits, and shall not transmit Completion Flow Control Credit Updates.

4.2.8 Data Integrity

The CCIX Transaction Layer shall follow all the requirements in the *PCI Express Base Specification* for Data Poisoning mechanisms. PCIe Data Poisoning is not supported for the Optimized TLP format.

If supported, the CCIX Transaction Layer shall follow all the requirements in the *PCI Express Base Specification* for End-to-End CRC (ECRC) based Data Integrity mechanisms. ECRC is not supported for the Optimized TLP format.

Refer to Section 6.2.10, Downstream Port Containment, *PCI Express Base Specification*, for DPC considerations (see Reference Documents).

4.2.9 Completion Timeout Mechanism

The CCIX Transaction Layer shall follow all the requirements in the *PCI Express Base Specification* for Completion Timeout, with an exception that they apply only to the CCIX VC behavior.

The CCIX VC exchanges only CCIX TLPs that use the Posted Credit category. Hence, Completion Timeout requirements shall not be applicable to the CCIX VC.

4.2.10 Link Status Dependencies

The CCIX Transaction Layer shall follow all the requirements in the *PCI Express Base Specification* for Link Status Dependencies.

4.3 CCIX Data Link Layer

4.3.1 REPLAY_TIMER Limits for 20.0 GT/s and 25.0 GT/s

Implementations that support 20.0 GT/s or 25.0 GT/s must use the Simplified REPLAY_TIMER Limits specified in the *PCI Express Base Specification,* Section 3.6.2.1, (see Reference Documents) for operation at all data rates.

4.3.2 AckNak_LATENCY_TIMER Limits for 20.0 GT/s and 25.0 GT/s

Implementations that support 20.0 GT/s or 25.0 GT/s must use the AckNak_LATENCY_TIMER Limits specified in the *PCI Express Base Specification,* Table 3-10 (see Reference Documents) when operating at 20.0 GT/s or 25.0 GT/s data rates.

4.4 CCIX Physical Layer Logical Block

4.4.1 Introduction

The Physical Layer isolates the Transaction Layer and Data Link Layers from the signaling technology used for Link data interchange. The Physical Layer is divided into the logical and electrical sub-blocks. CCIX Transport Specification extends both logical and electrical sub-blocks as specified in the *PCI Express Base Specification* (see Reference Documents). This chapter specifies the CCIX logical sub-block requirements, while Chapter 5 contains the CCIX electrical sub-block requirements.

4.4.2 CCIX Logical Sub-block

The CCIX Physical Layer optionally extends the PCIe Physical Layer, adding 20.0 GT/s and 25.0 GT/s data rates. CCIX devices that optionally support Extended Data Rates must support both 20.0 GT/s and 25.0 GT/s date rates of operation in addition to all required data rates supported by a 16.0 GT/s capable *PCI Express Base Specification* compliant component.

4.4.2.1 CCIX Extended Speed Mode (ESM)

A CCIX device optionally supports Extended Data Rates. A CCIX device that supports Extended Data Rates enters ESM when it is instructed to transition to one of the extended data rates via the ESMControl.ESMEnable bit transitioning from 0b to 1b, when operating at 2.5 GT/s or 5.0 GT/s. Once entered, ESM rules redefine the data rate for the 8.0 GT/s and/or 16.0 GT/s speeds, to ESM Data Rate0 and ESM Data Rate1 respectively.

A CCIX port must exit ESM following fundamental reset, link down reset, or hot reset event and return to PCIe mode. Function Level Reset (FLR) does not affect ESM. $D3_{hot}$ to D0 transition events do not affect ESM.

4.4.2.2 CCIX PHY Types

A CCIX device must support one of two PHY Types: PCI Express capable or Extended Data Rate (EDR).

PCI Express capable PHY:

- A PCI Express capable PHY must be compliant with the *PCI Express Base Specification* and support 16.0 GT/s operation.
- Figure 4-11 shows data rate transitions supported by a PCI Express capable PHY.

Components implementing the PCI Express capable PHY must hardwire the CCIXTransportCapabilities.ESMModeSupported bit in CCIX Transport Capabilities register to 0b.

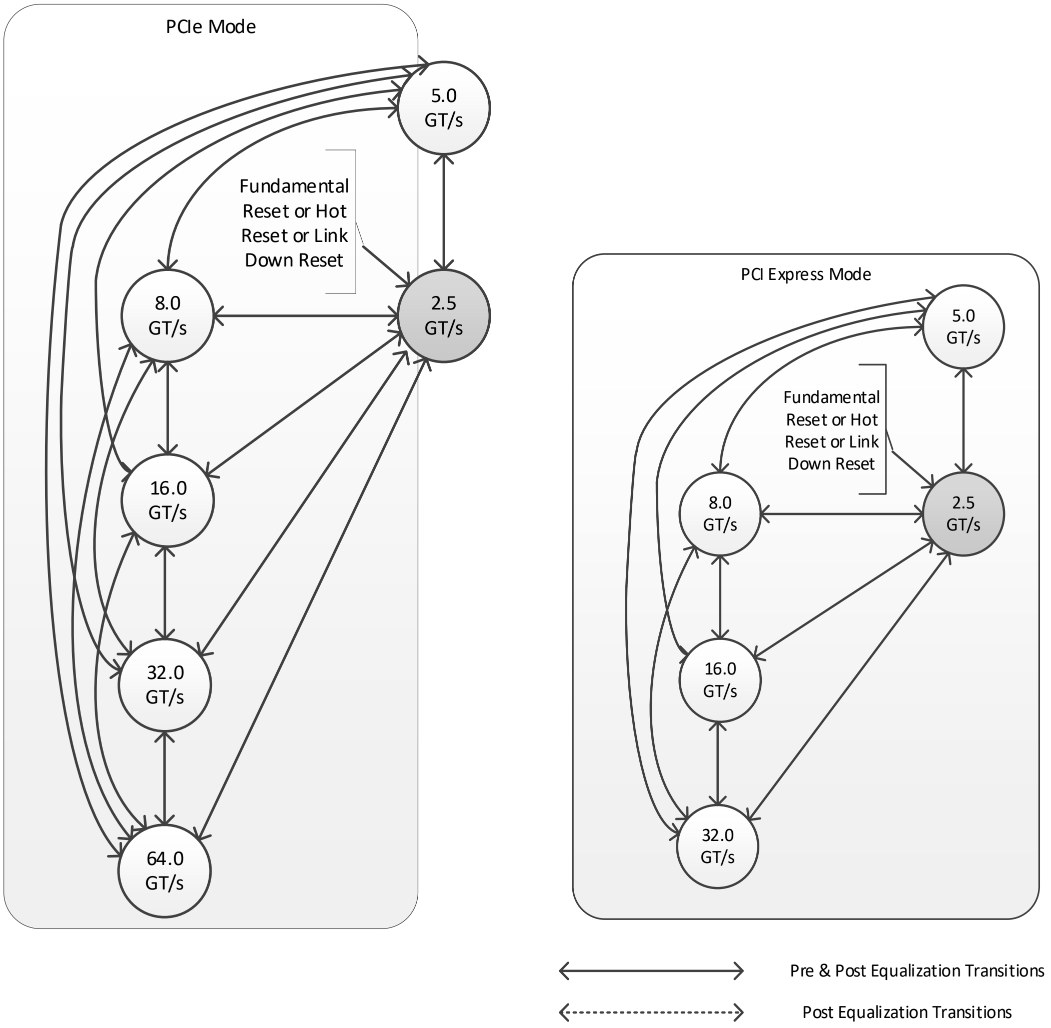

PCIe Mode
5.0 GT/s
Fundamental Reset or Hot Reset or Link Down Reset
8.0 GT/s
2.5 GT/s
16.0 GT/s
32.0 GT/s
64.0 GT/s
PCI Express Mode
5.0 GT/s
Fundamental Reset or Hot Reset or Link Down Reset
8.0 GT/s
2.5 GT/s
16.0 GT/s
32.0 GT/s
Pre & Post Equalization Transitions
Post Equalization Transitions

Figure 4-11: PCI Express Capable PHY

Extended Data Rate (EDR) PHY:

- The EDR PHY Type must support the capabilities of a PCI Express capable PHY as well as supporting the 20.0 GT/s and 25.0 GT/s data rates.
- Two ESM data rates are architected in this specification; ESMDataRate0 and ESMDataRate1. While only 2 rates will be used (programmed) at any given time, rate selection is chosen from several available rates. These rates are identified in the ESMMandatoryDataRateCapability and the ESMOptionalDataRateCapability registers. See Section 6.3.2 and 6.3.3. Data rates that an EDR PHY is required to support are specified in Section 5.1.
- The Initial Link-Up Phase is compliant with the *PCI Express Base Specification*, followed by Flow Control Initialization (DLCSM in DL Active), and the ESM Phase.
- The ESM Phase that follows continues to use the PCI Express Link Training and Status State Machine (LTSSM) states for Data Rate transitions to 20.0 GT/s or 25.0 GT/s, as well as other important functions (such as lane-lane de-skew at new data rate(s) etc.).

- Figure 4-12 shows data rate transitions during ESM PHY operation. Transitions between PCIe and ESM Modes only occur when operating at 2.5 GT/s or 5.0 GT/s data rates.
- A new software-based method is defined in the ESM Phase to select and change ESM Data Rates.
- Components implementing the EDR PHY Type must hardwire the CCIXTransportCapabilities.ESMModeSupported bit to 1b (see Table 6-67).

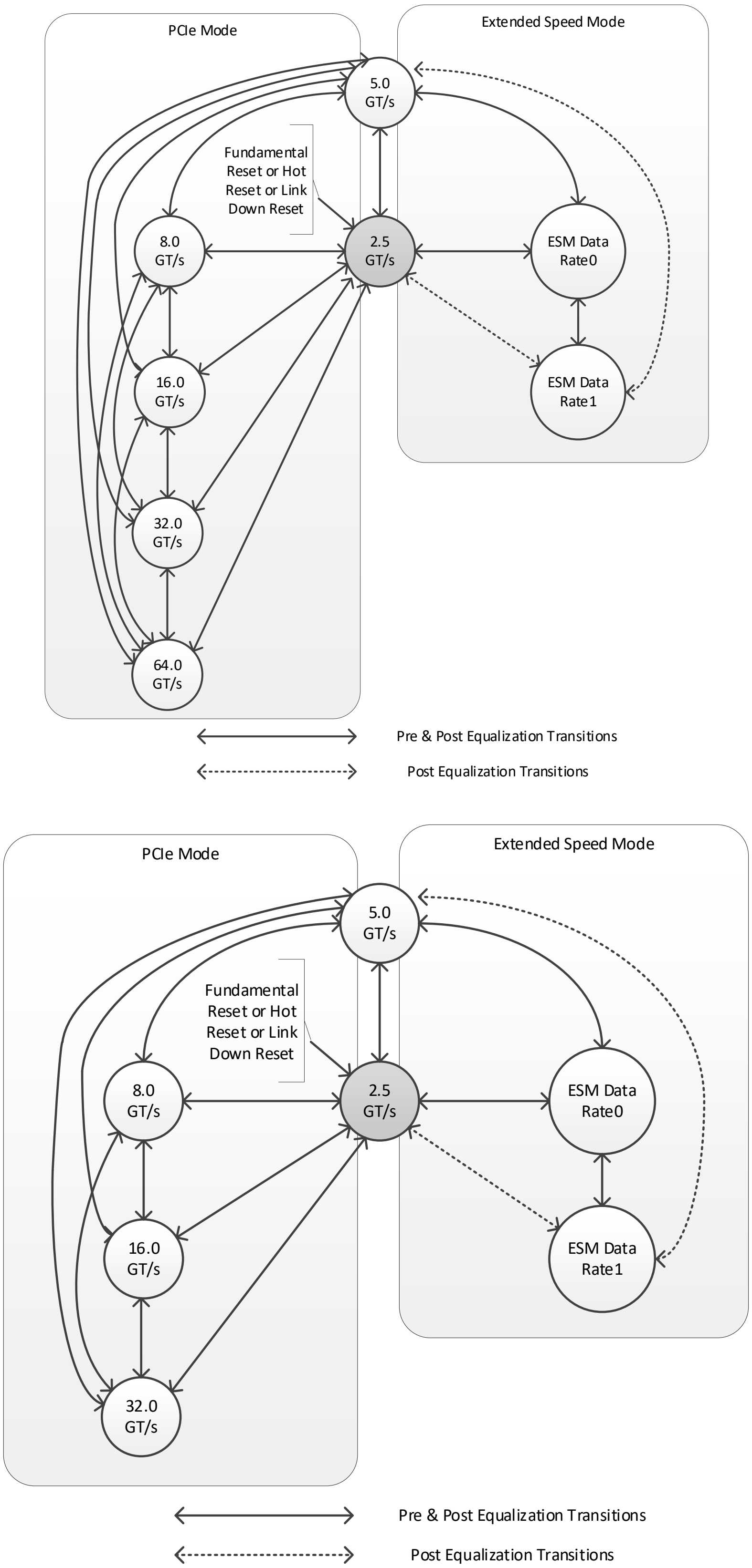

PCIe Mode
Extended Speed Mode
5.0 GT/s
Fundamental Reset or Hot Reset or Link Down Reset
8.0 GT/s
2.5 GT/s
ESM Data Rate0
16.0 GT/s
ESM Data Rate1
32.0 GT/s
64.0 GT/s
Pre & Post Equalization Transitions
Post Equalization Transitions
PCIe Mode
Extended Speed Mode
5.0 GT/s
Fundamental Reset or Hot Reset or Link Down Reset
8.0 GT/s
2.5 GT/s
ESM Data Rate0
16.0 GT/s
ESM Data Rate1
32.0 GT/s
Pre & Post Equalization Transitions
Post Equalization Transitions

Figure 4-12: Extended Data Rate PHY

4.4.2.3 ESM Operation

The steps involved in ESM operation are described below. Refer to Figure 6-85, Figure 6-86, Figure 6-87, Figure 6-88, and Figure 6-89; the corresponding definitions in Table 6-68, Table 6-69, Table 6-70, and Table 6-71, respectively.

1 **Step 1** - PCIe Compliant Phase

a The Physical Link is fully compliant to the *PCI Express Base Specification*.

b Initial transition to LTSSM L0 state.

c Link-Up to DL_Active state and Flow Control Initialization is complete.

2 **Step 2 - Software-based discovery and PHY Calibration in L1 state.**

a System Software (SSW) probes configuration space to determine whether the CCIX Transport DVSEC capability (Section 6.2.2.10) is present and the ESM Mode Supported bit is Set in both the Downstream Port (DSP) and the Upstream Port (USP). When these conditions are met in both USP and DSP ports, SSW uses LinkControl2.TargetLinkSpeed in the DSP to transition the link to 2.5 GT/s data rate. If either DSP or USP does not support the CCIX Transport DVSEC or ESM Mode, then an ESM transition must not be initiated, and remaining steps are not taken.

b SSW determines the Link Reach Target, Short Reach (SR) or Long Reach (LR), according to the CCIXTransportCapabilities.ESMPHYReachLengthCapability bit on both USP and DSP, as well as platform-specific channel capability (whether SR or LR compliant), and sets the LinkReachTarget bit, for SR or LR accordingly, on both the USP and the DSP.

c SSW reads ESMControl.LinkReachTarget (SSW must configure this bit before entering ESM Operation) on both the DSP and the USP. If LinkReachTarget is 0b (Short Reach) on both DSP and USP, SSW Clears ESMControl.ESMExtendedEqualizationPhase2Timeout and ESMControl.ESMExtendedEqualizationPhase3Timeout on both the USP and the DSP. If LinkReachTarget is 1b (Long Reach) on either the DSP or the USP, SSW reads the USP's ESMControl.ESMExtendedEqualizationPhase2Timeout value and writes the corresponding value into the DSP's ESMControl.ESMExtendedEqualizationPhase2Timeout field. SSW then reads the DSP's ESMControl.ESMExtendedEqualizationPhase3Timeout value and writes the corresponding value to the USP's ESMControl.ESMExtendedEqualizationPhase3Timeout field.

d SSW reads the Supported bits in the ESMMandatoryDataRateCapabilities and ESMOptionalDataRateCapabilities registers in DSP and USP, and determines the appropriate data rates to be programmed into the ESMControl.ESMDataRate fields. The actual data rate(s) selection algorithm is platform specific.

e SSW sets the selected data rates in the ESMControl.ESMDataRate fields. Figure 4-13 shows the procedure detailed below.

1. SSW programs ESMDataRate1 to the data rate selected in step 2d.
2. After ESMDataRate1 is programmed, hardware is permitted to change the data rates advertised in the ESMMandatoryDataRateCapabilities and

ESMOptionalDataRateCapabilities registers. The updated values must be present before the completion is generated for any read of those registers after ESMDataRate1 is written.

3. SSW does (or redoes) Step 2d to determine the data rate for ESMDataRate0 and programs ESMDataRate0.

4. If the value for ESMDataRate0 would have been ‘No Speed’, reprogram ESMDataRate0 to the value in ESMDataRate1, and program ESMDataRate1 to a value of ‘No Speed’.

5. When setting the ESMDataRate fields, the rules specified in Section 6.3.5.1 must be followed.

6. Changing the values of either ESMDataRate0 or ESMDataRate1 from their previous values (including from No Speed) must force an entry into the Equalization states in Step 3 below.

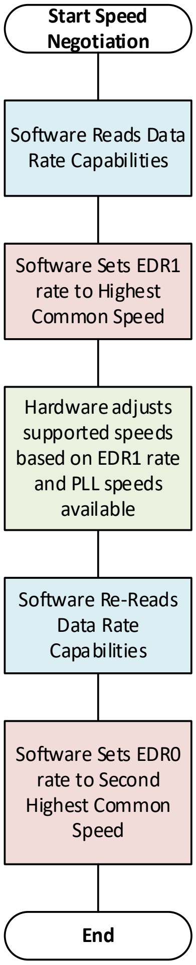

Start Speed Negotiation
Software Reads Data Rate Capabilities
Software Sets EDR1 rate to Highest Common Speed
Hardware adjusts supported speeds based on EDR1 rate and PLL speeds available
Software Re-Reads Data Rate Capabilities
Software Sets EDR0 rate to Second Highest Common Speed
End

Figure 4-13: Programming ESM Data Rates

f SSW reads the CCIXTransportCapabilities.ESMCalibrationTime values on the DSP and the USP. SSW then selects the maximum of the two capability values as the upper limit for time spent in the calibration step.

g SSW reads and stores for later restoration, the values the PCI-PM L1.1 Enable, PCI-PM L1.2 Enable, ASPM L1.1 Enable, ASPM L1.2 Enable bits in the L1 PM Substates Control 1 register, and the Enable Clock Power Management bit of the Link Control register, then clears those bits.

h SSW initiates the calibration step by Setting the ESMControl.ESMPerformCalibration bit on the DSP and the USP.

i SSW then drives the link to the L1 Low Power State, using the PCI Express software-managed Power Management (*PCI Express Base Specification,* Section 5.6.2) (PPM) mechanism.

  i. A transition to the L1 state may be optionally used to calibrate the PHY logic for the data rates programmed into ESMDataRate0 and ESMDataRate1.

  ii. PHY calibration activities include, but are not limited to, PLL spin-up and stabilization, PLL reprogramming and reset, transmit lane-lane de-skew etc.

  iii. Device hardware performs calibration and Sets the ESMStatus.ESMCalibrationComplete bit when calibration is complete.

j. SSW drives an exit from the L1 Low Power State using the PPM mechanism, after the timeout specified by ESMCalibrationTime (Step (f) above) has expired.

k. On exit from L1 state, SSW must check that the ESMStatus.ESMCalibrationComplete bit is Set on both the DSP and USP, to ensure that the ESM calibration step is complete. If the ESMCalibrationComplete bit is Clear on either the USP or the DSP, SSW does not execute the rest of the ESM transition, an error is logged by SSW in an implementation-specific manner, and the physical layer remains in PCIe mode at the 2.5 GT/s data rate.

l. SSW restores the bits that were cleared in Step 2g.

m. SSW Sets the ESMControl.ESMEnable bit on both the DSP and the USP. LTSSM variables related to the 8.0 GT/s and 16.0 GT/s Data Rate Identifiers must be remapped for ESM Data Rate0 and ESM Data Rate1, respectively. Equalization status bits are reset at this point. See Section 4.4.2.3.1 for more details.

n. If Link-Up is Set after a Hot Reset event on the link, SSW may optionally set the ESMControl.QuickEqualizationTimeoutSelect field on both the DSP and the USP. Before setting the ESMControl.QuickEqualizationTimeoutSelect field, SSW must first read the CCIXTransportCapabilities.ESMQuickEqualizationTimeout field on both the USP and DSP, and select the greater of the two encoded values. SSW will then program that value into the ESMControl.QuickEqualizationTimeoutSelect field on both the USP and the DSP. See the ESMControl.QuickEqualizationTimeoutSelect field description and the Implementation Note “ESM Control Register Quick Equalization Timeout Select Field” that follows. Implementations are permitted to use the Quick Equalization mechanism to redo equalization due to correctable errors detected on the link or for other reasons. Use of the Quick Equalization mechanism is prohibited for Link-Up after a Cold Reset, Warm Reset, Link Disable and Link Down Reset events.

3 **Step 3 - Link transition to selected ESM Data Rates**

a SSW sets LinkControl2.TargetLinkSpeed to 0100b (Supported Link Speeds Vector bit 3), on the DSP, and sets LinkControl.LinkRetrain on the DSP.

b At this point, a Data Rate change occurs based on PCI Express LTSSM.

c The following LTSSM states have been modified. These changes are applicable to Data Rate transitions beyond those specified in the *PCI Express Base Specification*.

i. **Recovery.Speed** LTSSM State (*PCI Express Base Specification*):

i. The Transmitter enters Electrical Idle and stays there until the Receiver Lanes have entered Electrical Idle, and then additionally remains there for at least 800 ns on a successful speed negotiation (i.e., successful_speed_negotiation = 1b) or at least 6 µs on an unsuccessful speed negotiation (i.e., successful_speed_negotiation = 0b), but stays there no longer than an additional 2 ms.

ii. **Recovery.Equalization** (*PCI Express Base Specification*)

i. Downstream Lanes, Phase 2 of Transmitter Equalization (*PCI Express Base Specification*, Section 4.2.6.4.2.1.2, change the final ‘Else’ clause from ‘Else, next state is Recovery.Speed after a 32 ms timeout with a tolerance of -0 ms and +4 ms.’ to ‘Else if the current data rate is greater than 16.0 GT/s, next state is Recovery.Speed, after a timeout, as indicated by ESMControl.ESMEqualizationPhase2Timeout, else next state is Recovery.Speed after a 32 ms timeout with a tolerance of -0 ms and +4 ms.’

ii. Downstream Lanes, Phase 3 of Transmitter Equalization (*PCI Express Base Specification*, Section 4.2.6.4.2.1.3, change the last ‘Else’ clause from ‘Else, next state is Recovery.Speed after a 24 ms timeout with a tolerance of -0 ms and +2 ms.’ to ‘Else if the current data rate is greater than 16.0 GT/s, next state is Recovery.Speed, after a timeout, as indicated by ESMControl.ESMEqualizationPhase3Timeout, else next state is Recovery.Speed after a 24 ms timeout with a tolerance of -0 ms and +2 ms.’

iii. Upstream Lanes, Phase 2 of Transmitter Equalization (*PCI Express Base Specification*, Section 4.2.6.4.2.2.3, change the last ‘Else’ clause from ‘Else, next state is Recovery.Speed after a 24 ms timeout with a tolerance of -0 ms and +2 ms.’ to ‘Else if the current data rate is greater than 16.0 GT/s, next state is Recovery.Speed, after a timeout, as indicated by ESMControl.ESMEqualizationPhase2Timeout, else next state is Recovery.Speed after a 24 ms timeout with a tolerance of -0 ms and +2 ms.’

iv. Upstream Lanes, Phase 3 of Transmitter Equalization (*PCI Express Base Specification*, Section 4.2.6.4.2.2.4, change the final ‘Else’ clause from ‘Else, next state is Recovery.Speed after a 32 ms timeout with a tolerance of -0 ms and +4 ms.’ to ‘Else if the current data rate is greater than 16.0 GT/s, next state is Recovery.Speed, after a timeout, as indicated by ESMControl.ESMEqualizationPhase3Timeout, else next state is Recovery.Speed after a 32 ms timeout with a tolerance of -0 ms and +4 ms.’

d When the data rate programmed in ESMControl.ESMDataRate1 is achieved, and when Link-Up is achieved after a Hot Reset event and SSW has previously programmed the ESMControl.QuickEqualizationTimeoutSelect field to a value greater than 000b, then SSW must clear the ESMControl.QuickEqualizationTimeoutSelect field by programming it to 000b on both the USP and the DSP.

4.4.2.3.1 ESM Operational Mechanisms

On entering ESM mode, actions associated with Symbol 4 of PCI Express TS1 Ordered Set and TS2 Ordered Set change as shown in Table 4-3. Actions associated with Bit 3 and Bit 4 of Symbol 4 are performed according to the requirements in *PCI Express Base Specification*.

By programming one of the supported data rates into ESMControl.ESMDataRate0/1 fields, the hardware must take the appropriate data rate change actions as required by the *PCI Express Base Specification*, except that the target data rate is governed by the value programmed into the ESMDataRate0/1 fields.

Section 6.3.5.1 specifies additional rules governing programming of ESMDataRate0/1 fields.

Unless otherwise specified, when transitioning to, or operating at, an ESM Data Rate of 8.0 GT/s, all reguirements corresponding to transitioning to, or operating at, the 8.0 GT/s Data Rate specified in the *PCI Express Base Specification* must be adhered to.

Unless otherwise specified, when transitioning to, or operating at, an ESM Data Rate of 16.0 GT/s, 20.0 GT/s, or 25.0 GT/s, all requirements corresponding to transitioning to, or operating at, the 16.0 GT/s Data Rate specified in the *PCI Express Base Specification* must be adhered to.

Unless otherwise specified, when transitioning to, or operating at, an ESM Data Rate of 32.0 GT/s, all reguirements corresponding to transitioning to, or operating at, the 32.0 GT/s Data Rate specified in the *PCI Express Base Specification* must be adhered to.

Unless otherwise specified, when transitioning to, or operating at, an ESM Data Rate of 64.0 GT/s, all reguirements corresponding to transitioning to, or operating at, the 64.0 GT/s Data Rate specified in the *PCI Express Base Specification* must be adhered to.

Unless otherwise specified, Alternate Protocol operates with an EDR PHY exactly as Alternate Protocol is specified for a PCIe PHY in the *PCI Express Base Specification.*

Ordered Sets and speed changes behave as specified in the *PCI Express Base Specification*, with ESMDataRate0 replacing 8.0 GT/s in the definition of the 8.0 GT/s Data Rate Identifier (DRI), ESMDataRate1 replacing 16.0 GT/s in the definition of the 16.0 GT/s DRI, and DRI’s greater than 16.0 GT/s are reserved.

- When operating at an ESM data rate:
  - At 20.0 GT/s or 25.0 GT/s data rates, EIEOS’s shall be transmitted and received as defined in Table 4-4 and Figure 4-14.
  - Lane Margining at Receiver as defined in the *PCI Express Base Specification* is permitted but not required when the link is operating in ESM Data Rate0 at 16.0 GT/s.
  - When performing equalization at an ESM Data Rate of 20.0 GT/s, parameters are to be used from the Lane Equalization Control registers in the ESMLaneEqualizationControl registers for 20.0 GT/s.

    - When performing equalization at an ESM Data Rate of 25.0 GT/s, parameters are to be used from the Lane Equalization Control registers in the ESMLaneEqualizationControl registers for 25.0 GT/s.
    - Equalization status bits pertaining to 8.0 GT/s in the Link Status register are used for ESM Data Rate0 when it is programmed to 16.0 GT/s.
    - Equalization status bits pertaining to 16.0 GT/s in the 16.0 GT/s Status register are used for an ESM Data Rate of 20.0 GT/s or 25.0 GT/s, or ESM Data Rate1 when it is programmed to 16.0 GT/s.

- Electrical Compliance testing uses LTSSM states and mechanisms defined in the *PCI Express Base Specification*.
    - Electrical Compliance test procedures will depend on implementation-specific mechanisms to write and read the ESMControl and ESMStatus registers in the Transport DVSEC.
    - Compliance-related additions are required in the following LTSSM states.
        - Polling.Compliance
            - When in Polling.Compliance operating at the 2.5 GT/s Data Rate with a De-emphasis Level = -3.5dB, a transition to Detect.Quiet must occur if all of the following conditions are true:
                - ESMControl.ESMCompliance is Set
                - ESMControl.ESMPerformCalibration has been Set, arming this process
                - An exit from Electrical Idle is detected on any configured lanes
        - Detect.Quiet

- On entry into Detect.Quiet, if ESMControl.ESMCompliance is Set, the 12 ms timer is held in reset, the Electrical Idle input is ignored, and PHY Calibration is performed.
- When PHY Calibration is complete, the 12 ms timer and Electrical Idle inputs are enabled.

Table 4-3: PCI Express TSx Symbol 4 Description

Symbol Number	PCI Express Compliant Phase Description	ESM Phase Description
4	Data Rate Identifier Bit 0 – Reserved for future Data Rate Bit 1 – 2.5 GT/s Data Rate Supported. Must be set to 1b. Bit 2 – 5.0 GT/s Data Rate Supported. Must be set to 1b if Bit 3 is 1b. Bit 3 – 8.0 GT/s Data Rate Supported. Must be set to 1b if Bit 4 is 1b Bit 4 – 16.0 GT/s Data Rate Supported. Bit 5 – Reserved for future Data Rate	Data Rate Identifier Bit 0 – Reserved for future Data Rate Bit 1 – 2.5 GT/s Data Rate Supported. Must be set to 1b. Bit 2 – 5.0 GT/s Data Rate Supported. Must be set to 1b. Bit 3 – ESM Data Rate0 Supported. Must be set to 1b if Bit 4 is 1b Bit 4 – ESM Data Rate1 Supported. Bit 5 – Reserved for future Data Rate

Table 4-4: Electrical Idle Exit Ordered Set (EIEOS) for 20.0 GT/s and 25.0 GT/s Data Rates

Symbol Numbers	Value	Description
0, 1, 2, 3, 8, 9,10,11	00h	Symbol 0: EIEOS Identifier A low frequency pattern that alternates between thirty-two 0s and thirty-two 1s.
4, 5, 6, 7,12,13,14,15	FFh	A low frequency pattern that alternates between thirty-two 0s and thirty-two 1s.

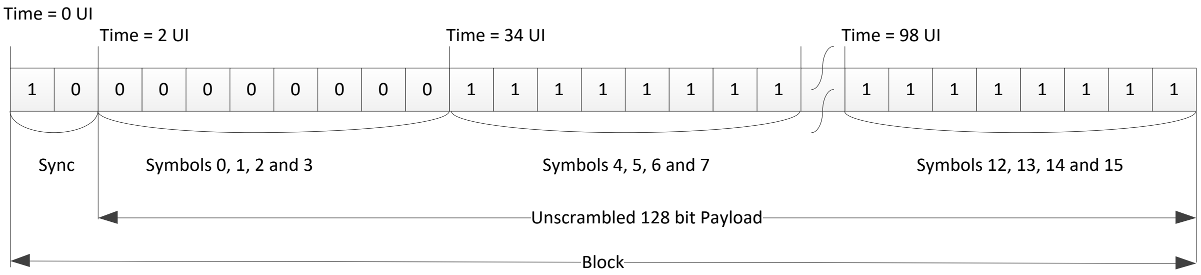


Figure 4-14: Electrical Idle Exit Ordered Set (EIEOS) for 20.0 GT/s and25.0 GT/s Data Rates

4.4.2.3.1.1 ESM Initial Link-Up to 25.0 GT/s ESM Operation

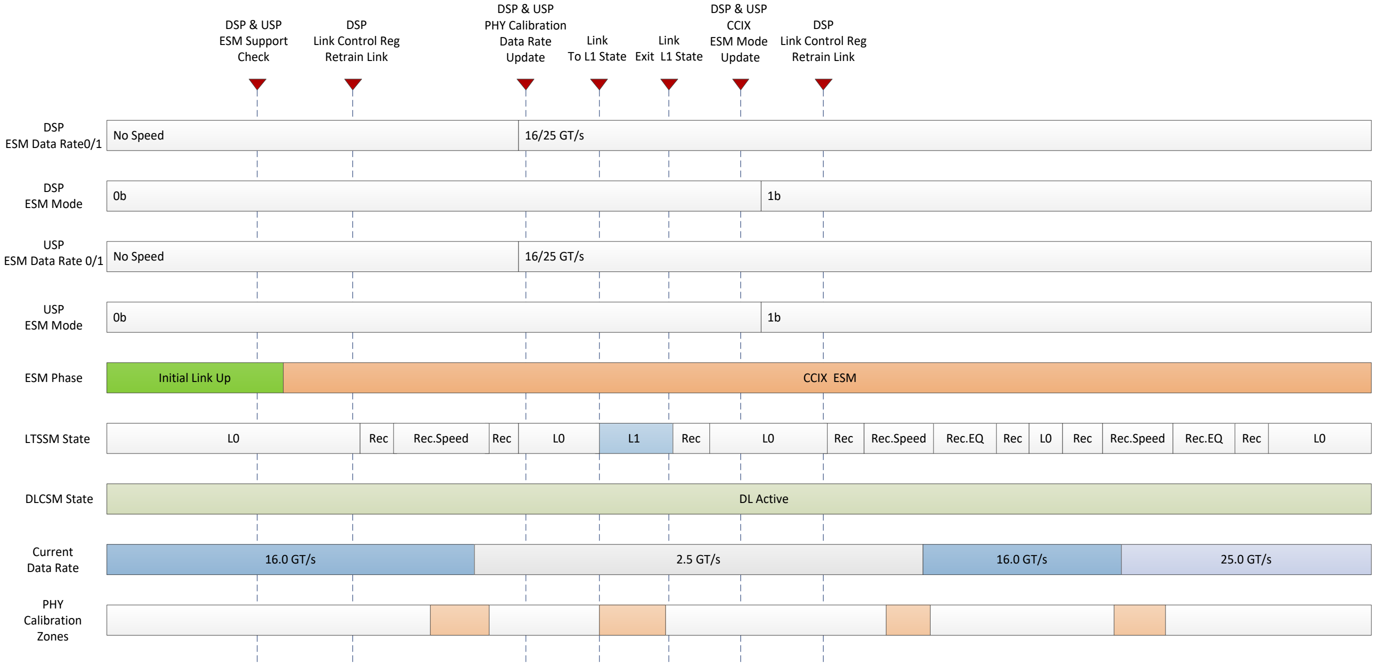


Figure 4-15: Link-Up to ESM 25.0 GT/s through ESM 16.0 GT/s data rate

4.4.3 Retimers

This section defines the requirements for Retimers that are Physical Layer protocol aware and that interoperate with a pair of Components and a compliant channel on each side of the Retimer. Retimers on CCIX links up to 16.0 GT/s Data Rates must be compliant to requirements specified in the *PCI Express Base Specification*, Section 4.3, (see Reference Documents). Retimers on CCIX links that support greater than 16.0 GT/s (EDR) Data Rates must be compliant to requirements in this section.

Requirements for retimers supporting greater than 16.0 GT/s data rates are expected to be available in future revisions of this specification.

Chapter 5. Electrical PHY Layer

5.1 Introduction

A CCIX® 1.1 device must support one of two PHY types: PCI Express® (PCIe) 5.0 or CCIX Extended Data Rate (EDR) PHY.

The PCIe 5.0 PHY type must be compliant to *PCI Express Base Specification 5.0*. The EDR PHY type supports 20 GT/s and 25 GT/s in addition to *PCI Express Base Specification 5.0*. Table 5-1 defines the required and optional data rates for each CCIX 1.1 PHY type.

Table 5-1: CCIX PHY Types and Line Rates

<table>
<tr><th rowspan="2"><em>Line Rate (GT/s)</em></th><th colspan="2"><em>CCIX 1.1 PHY Type</em></th></tr>
<tr><th><em>PCIe 5.0 PHY</em></th><th><em>EDR PHY</em></th></tr>
<tr><td><em>2.5</em></td><td colspan="2"><em>Required</em></td></tr>
<tr><td><em>5.0</em></td><td colspan="2"><em>Required</em></td></tr>
<tr><td><em>8.0</em></td><td colspan="2"><em>Required</em></td></tr>
<tr><td><em>16</em></td><td colspan="2"><em>Required</em></td></tr>
<tr><td><em>20</em></td><td><em>NA</em></td><td><em>Required</em></td></tr>
<tr><td><em>25</em></td><td><em>NA</em></td><td><em>Required</em></td></tr>
<tr><td><em>32</em></td><td colspan="2"><em>Optional</em></td></tr>
</table>

Note: The PHY follows PCIe 5.0 electrical specifications for PCIe data rates, which allows for compliance to PCIe 4.0 electrical specifications for PHYs that do not support 32 GT/s. EDR PHYs must comply to CCIX 1.1 or CCIX 1.0 electrical specifications for 20 GT/s and 25 GT/s.

EDR PHY at 20 GT/s and 25 GT/s has two different electrical variants to support short reach or long reach links. Bits 2:1 of CCIX Transport Capabilities Register in Table 6-67 is used to configure the type of reach supported.

EDR PHY at other line rates has a single electrical variant that is defined at each electrical specification.

The compliant channel at each line rate in EDR PHY can be different. Each line rate must support the compliant channel based on corresponding electrical specification.

5.2 EDR25-SR Electrical Specification

The EDR25-SR electrical specification applies to the short reach application of ESM Data rates in EDR PHY which are 20GT/s and 25GT/s.

5.2.1 General Specification

5.2.1.1 Line Rate

EDR25-SR PHY for the high speed must support 20GT/s and 25GT/s.

5.2.1.2 Line Coding

EDR25-SR PHY for the high speed uses 128b/130b coding, same as PCIe Gen3 and Gen4.

5.2.1.3 Crosstalk

Crosstalk defined in EDR high speed specification is contributed from the elements such as connector and PCB of component-to-component channel. It should be minimized and at least less than the specified number which is defined informatively through Integrated Crosstalk Noise (ICN) from CEI-28G-SR standard.

5.2.1.4 Baud Rate Tolerance

EDR25-SR PHY is required to operate with common clock and tolerate up to +/-300 ppm from the nominal baud rate without spread spectrum clock.

Spread Spectrum Clock (SSC) is allowed for common clock operation.

Separate Reference Clocks with Independent SSC (SRIS) system will be added in future revision as an optional feature.

5.2.1.5 AC Coupling Capacitor

EDR25-SR link requires AC coupling only. The AC coupling cap value is 176nF to 265nF and its placement is determined by the form factor specification.

5.2.1.6 Target Bit Error Rate

The target Bit error rate in EDR25-SR link is 1.0E-12. All jitter calculations and compliance tests must use 1.0E-12 target BER.

5.2.1.7 Reference Model

The EDR High Speed common reference model is defined in Figure 5-1. EDR25-SR link has four reference points at both die pads and BGA balls. The socket is included in the package.

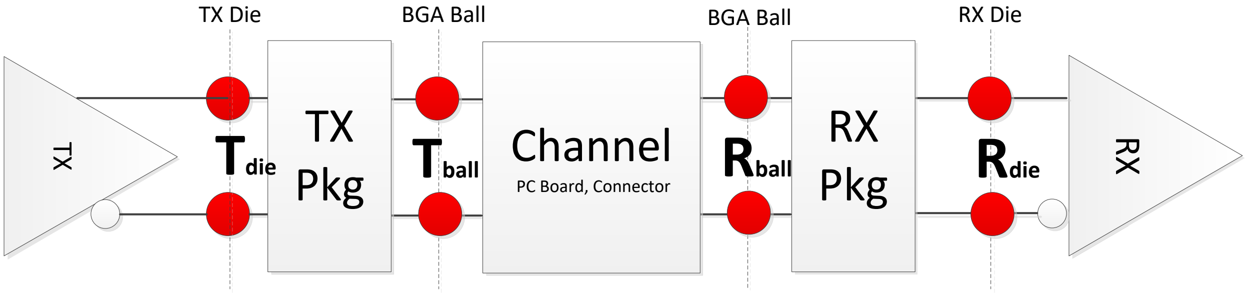


Figure 5-1: Reference Model of EDR25-SR Link

Figure 5-2 shows reference points on a downstream direction of EDR25-SR link.

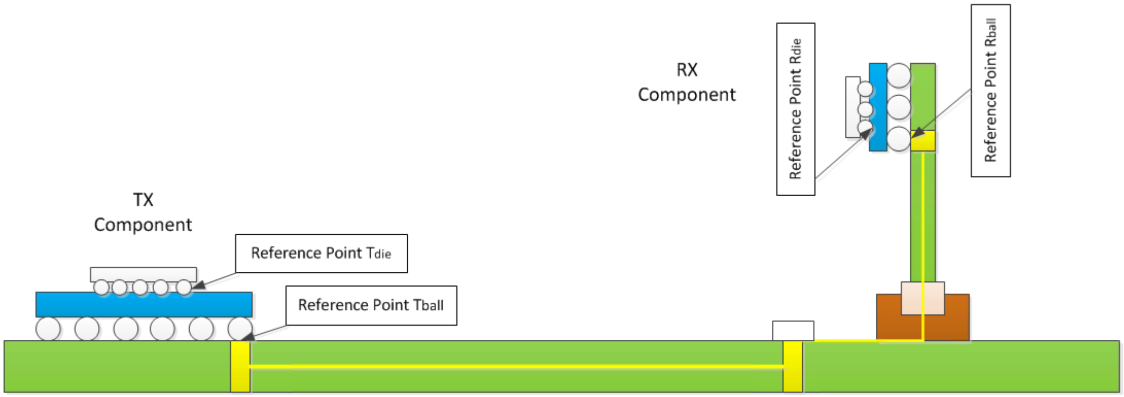


Figure 5-2: Reference Points on a downstream direction of EDR25-SR link

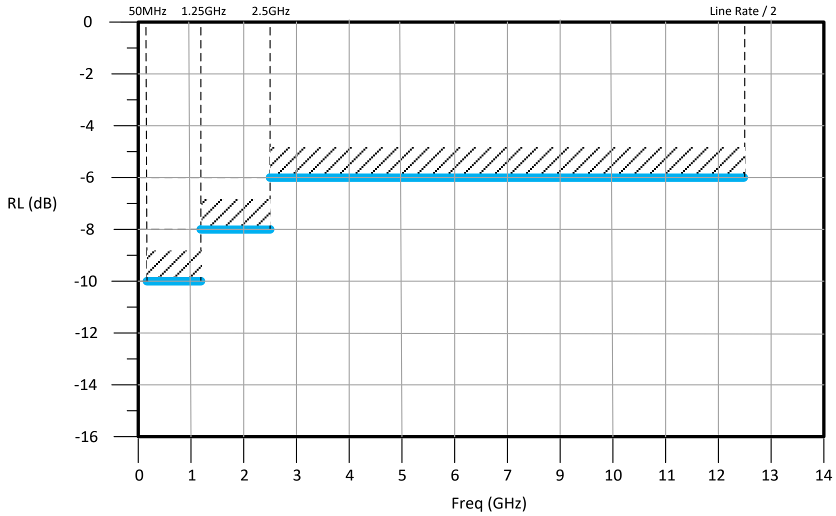
5.2.2 Transmitter Specification

5.2.2.1 Driver Termination

EDR25-SR PHY does not specify differential source termination at the driver as long as the TX return loss specification is met.

5.2.2.2 Differential Return Loss of Transmitter

The transmitter differential return loss at T_{ball} including package and socket is given in Figure 5-3.

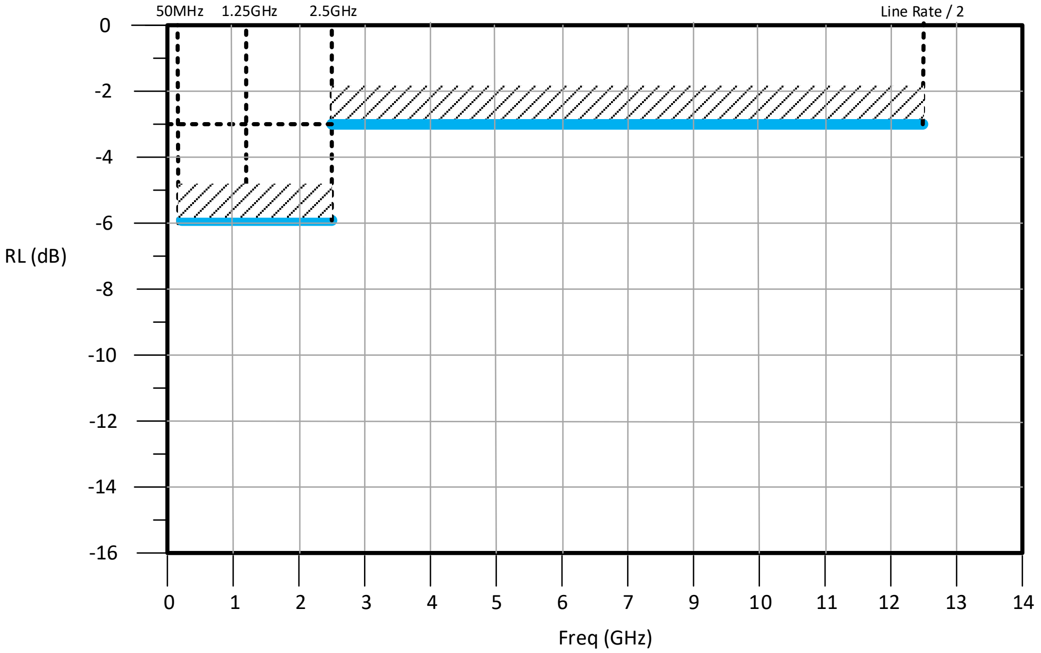


Note: Reference impedance is 100ohm

Figure 5-3: Transmitter Differential Return Loss Mask for EDR25-SR PHY

5.2.2.3 Common Mode Output Return Loss

The common mode return loss for T_{ball} including package and socket is given in Figure 5-4. To attenuate noise and absorb even/odd mode reflections, both transmitter and receiver is required to satisfy the Common Mode Return Loss requirement.



Note: Reference impedance is 25ohm

Figure 5-4: Transmitter Common Return Loss Mask for EDR25-SR PHY

5.2.2.4 Transmitter PLL Bandwidth and Peaking

Table 5-10 defines transmitter PLL bandwidth and peaking.

5.2.2.5 TX Voltage Parameters

5.2.2.5.1 Signal Definition

The signal definition in EDR25-SR PHY reuses voltage level definition in Section 8.3.2 of the *PCI Express Base Specification* (see Reference Documents).

5.2.2.5.2 Transmitter AC Specification

The transmitter electrical specifications are given in Table 5-2. The measurement of these parameters in EDR High Speed must support the methodology in Section 8.3 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-2: The Electrical Specification at Reference Point T_{ball}

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Differential peak-peak Tx voltage swing for full swing operation	$V_{TX\text{-}DIFF\text{-}PP}$	Measured with compliance test load	800		1300	mVpp
Differential peak-peak Tx voltage swing for low swing operation	$V_{TX\text{-}DIFF\text{-}PP\text{-}LOW}$	Measured with compliance test load	400		1300	mVpp
Minimum voltage swing during EIEOS for full swing signaling	$V_{TX\text{-}EIEOS\text{-}FS}$		250			mVpp
Minimum voltage swing during EIEOS for reduces swing signaling	$V_{TX\text{-}EIEOS\text{-}RS}$		232			mVpp
Pseudo package loss of root complex device (Informative)	$ps21_{TX\text{-}ROOT\text{-}DEVICE}$				7.0	dB
Pseudo package loss of non-root complex device (Informative)	$ps21_{TX\text{-}NON\text{-}ROOT\text{-}DEVICE}$				2.0	dB

Note: All of measurement should be referenced to 100ohm termination.

5.2.2.5.3 Transmitter Equalization

The EDR25-SR transmitter must support the equalization by using 3-tap FIR.

The transmitter equalization of EDR25-SR PHY reuses the specification of 16.0GT/s in Section 8.3.3.2 of the *PCI Express Base Specification* (see Reference Documents).

The control feature of each cursor is required for re-driver option.

5.2.2.5.4 TX Equalization Presets

The EDR High Speed transmitter must support the full range of presets in Section 8.3.3.3 of the *PCI Express Base Specification* (see Reference Documents).

The initial TX Preset must be set to achieve a minimum BER of 1.0E-4 for phase 1 of Recovery.Equalization. The methodology in Section 4.7.6 of the *PCIe 4.0 CEM Specification*, is a good practice to achieve the minimum BER when the inter-changeable add-in card slot is implemented for a CCIX system.

5.2.2.5.5 Measuring Presets at EDR25-SR PHY

The methodology to measure transmitter presets is defined in Section 8.3.3.5 of the *PCI Express Base Specification* (see Reference Documents), where 16.0GT/s Preset measurement is specified. The presets shall be measured at EDR25-SR PHY rates.

5.2.2.5.6 EIEOS and $V_{TX\text{-}EIEOS\text{-}FS}$ and $V_{TX\text{-}EIEOS\text{-}RS}$ Limit

The EIEOS patterns at EDR25-SR PHY consists of 32 consecutive ones followed by 32 consecutive zeros, where the pattern is repeated for a total of 128 UI.

Measuring $V_{TX\text{-}EIEOS\text{-}FS}$ and $V_{TX\text{-}EIEOS\text{-}RS}$ is defined in Section 8.3.3.9 of the *PCI Express Base Specification* (see Reference Documents), where 16.0GT/s measurement is specified except for measurement interval. The measurement interval is UI number 9-28 at 20 GT/s and 25 GT/s. The measured voltage level is averaged over this interval over more than 500 repetitions of the compliance pattern.

5.2.2.6 TX Timing Parameters

5.2.2.6.1 Jitter Parameters

The jitter parameters in EDR High Speed must support the definition and methodologies in Section 8.3.5 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-3 shows the jitter parameters in EDR25-SR PHY specification.

Table 5-3: The Jitter Specification at Reference Point T_{ball}

Characteristic	Symbol	Condition	20 GT/s	25 GT/s	Unit
			Max	Max	
TX Uncorrelated total jitter	$T_{TX\text{-}UTJ}$	At 1.0E-12	12.5	10.0	ps
TX Uncorrelated Dj for non-embedded Refclk	$T_{TX\text{-}UDJDD}$		5	4.0	ps

Total uncorrelated pulse width jitter	$T_{TX\text{-}UPW\text{-}TJ}$	At 1.0E-12	12.5	10.0	ps
Deterministic DjDD uncorrelated pulse width jitter	$T_{TX\text{-}UPW\text{-}DJDD}$		4	3.2	ps
TX Random Jitter	$T_{TX\text{-}RJ}$	Informative Parameter only	0.36 – 0.804	0.288 – 0.643	ps rms
Lane-to-Lane Output Skew	$L_{TX\text{-}SKEW}$		500ps + 8UI	500ps + 8UI	ps

5.2.2.6.2 De-Embedding the Breakout Channel for TX Jitter Measurement

When the TX jitter is measured at TP1, the replica channel can be used to de-embed the breakout channel.

The de-embedding methodology is defined in Section 8.3.5.2 of the *PCI Express Base Specification* (see Reference Documents), where 16.0GT/s de-embedding is specified.

5.2.2.6.3 Behavioral CDR Characteristics

A behavioral CDR of EDR25-SR PHY is the 1st order with 10MHz bandwidth. The transfer function of the 1st order CDR is defined at Section 8.3.5.5 of the *PCI Express Base Specification* (see Reference Documents).

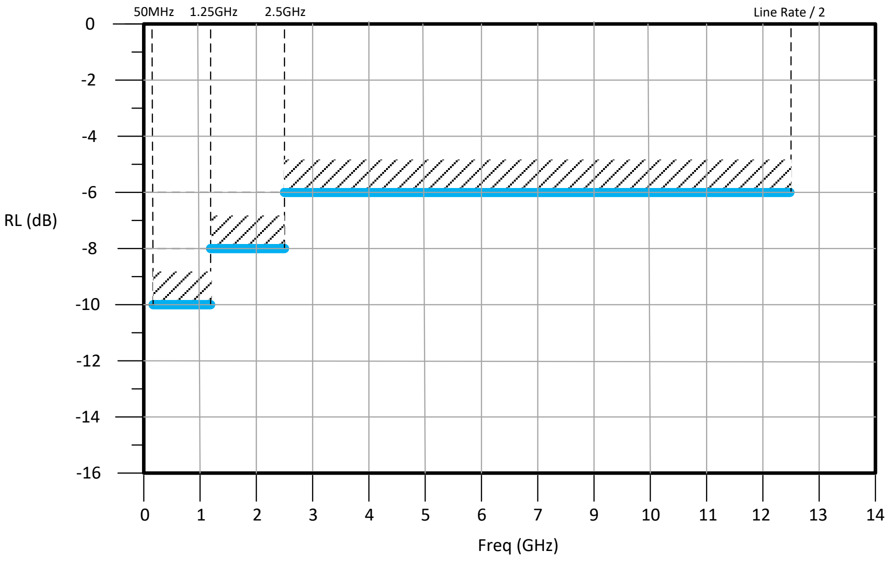
5.2.3 Receiver Specification

5.2.3.1 Receiver Termination

It specifies a nominal 100ohm differential load termination at DC at the receiver. The scheme for the termination needs to provide both differential and common mode termination to effectively absorb differential or common mode noise and reflections.

5.2.3.2 Differential Return Loss of Receiver

The receiver differential return loss at R_{ball} including package and socket is given in the Figure 5-3.

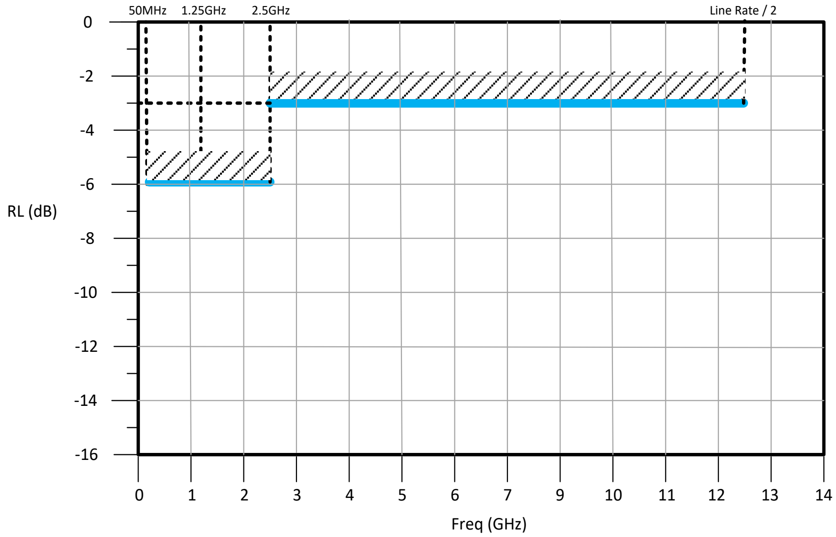


Note: The termination of the measurement is 100ohm

Figure 5-5: Receiver Differential Return Loss Mask for EDR25-SR PHY

5.2.3.3 Common Return Loss of Receiver

The common mode return loss for R_{ball} including package and socket is given in Figure 5-6. To attenuate noise and absorb even/odd mode reflections, receiver is required to satisfy the Common Mode Return Loss requirement.



Note: The termination of the measurement is 25ohm

Figure 5-6: Common Mode Return Loss Mask for EDR25-SR PHY

5.2.3.4 Common Receiver Parameters

The common receiver parameters are in Table 5-4. EDR25-SR PHY must follow the test condition in Section 8.4.3 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-4: Common receiver parameters

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Rx PLL bandwidth corresponding to PKGRX-PLL1	$BW_{RX-PKG-PLL1}$		2.0		4.0	MHz
Rx PLL bandwidth corresponding to PKGRX-PLL2	$BW_{RX-PKG-PLL2}$		2.0		5.0	MHz
Maximum Rx PLL peaking corresponding to BWRX-PKG-PLL1	$PKG_{RX-PLL1}$				2.0	dB
Maximum Rx PLL peaking corresponding to BWRX-PKG-PLL2	$PKG_{RX-PLL2}$				1.0	dB
Rx termination float time	$RX_{GND-FLOAT}$				500	ns
Rx AC common Mode Voltage	$V_{RX-CM-AC-P}$	Measured at RX pins into a pair of 50ohm terminations to GND			75 for EH <100mVPP 125 for EH >= 100mVpp	mVp
Electrical Idle Detect threshold	$V_{RX-IDLE-DET-DIFF-PP}$		65		175	mV
Unexpected Electrical Idle Enter Detect Threshold Integration Time	$T_{RX-IDLE-DET-DIFF-ENTERTIME}$				10	ms
Lane to Lane skew	$L_{RX-SKEW}$				4	ns

5.2.3.5 Jitter Tolerance

The jitter tolerance is given in Table 5-5. The receiver shall tolerate the calibrated stress eye and sinusoidal jitter from the below table. The calibration of stressed eye must follow same methodology in Section 8.4.2 of the *PCI Express Base Specification* (see Reference Documents).

The jitter tolerance measurement in CCIX-25G must follow the procedures in Section 8.4.2 of the *PCI Express Base Specification* (see Reference Documents).

Refclk in jitter tolerance measurement must include the spurs from SSC for the data-based CDR. It's typical to apply triangular frequency modulation between 30 and 33kHz to common Refclk as defined in the *PCI Express Base Specification* (see Reference Documents).

Table 5-5: Jitter Tolerance Specification

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Sinusoidal Jitter, Maximum	R_sj_max		1			UIpp
Sinusoidal Jitter, High Frequency	R_sj_hf		0.1			UIpp

The jitter tolerance mask is given in Figure 5-7.

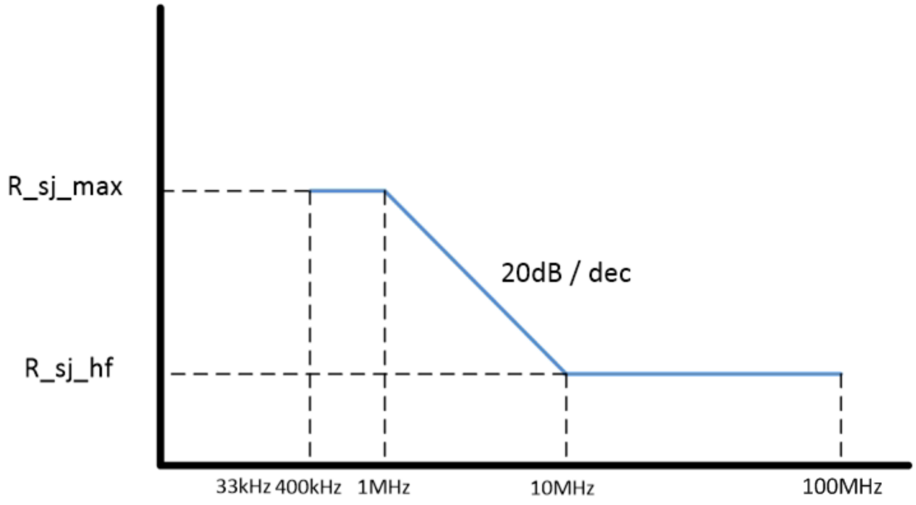


Figure 5-7: Jitter Tolerance Mask of EDR25-SR Receiver

5.2.3.6 Receiver Stressed Eye Specification

All ESM Data rates of EDR PHY are required to be tested by a stressed eye over calibration channel which closely approximates the worst-case loss characteristic in the real CCIX-SR application. The recovered eye is defined at TP2P which is at the input of the receiver’s analog latch.

5.2.3.6.1 Breakout and Replica Channels

Replica channel which is closely match to breakout channel of DUT board makes the measurement of the signal at RX DUT’s pin possible. The impedance target of replica channel is required to match with the impedance of breakout channel which can be either 100ohm differential or 85ohm with +/-5% or better.

Figure 5-8 shows the RX Test board topology for CCIX-SR. The stressed eye needs to be measured at TP2 point with external signal source. After the stressed eye has been calibrated, this signal is applied to DUT over the breakout channel.

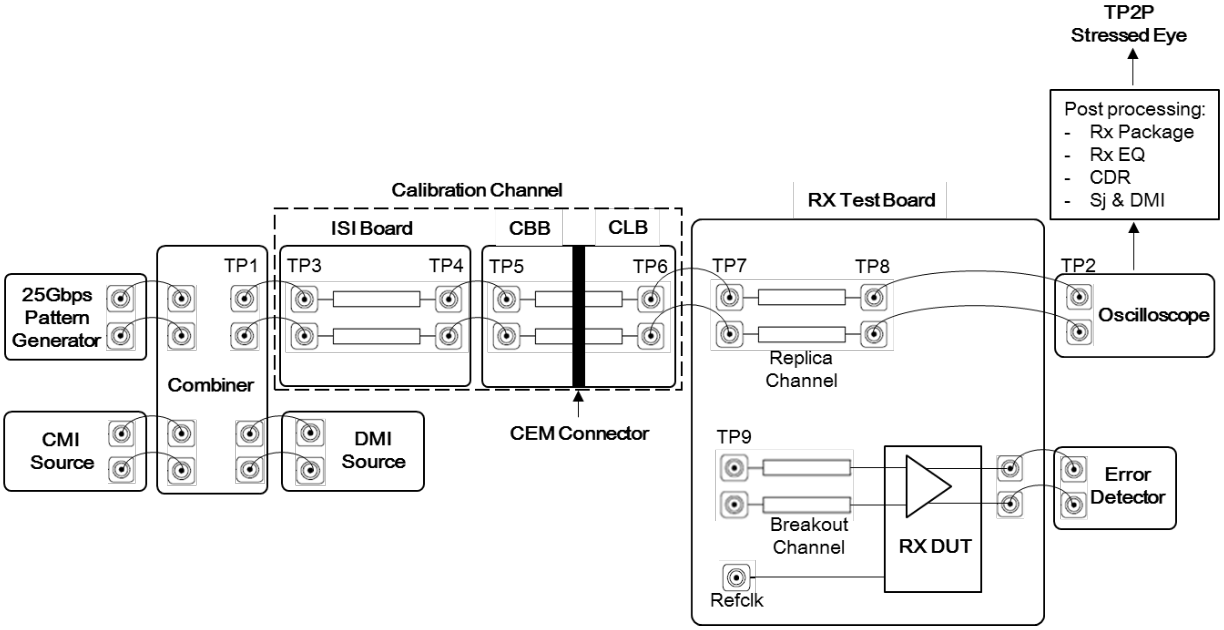


Figure 5-8: RX Test Board Topology

Note1: Reference boards are using 2.92mm connector to connect each test points.

Note2: All cables and discrete broadband termination are 50ohm.

5.2.3.6.2 Calibration Channel Insertion Loss Characteristics

The calibration channel loss specification reuses the definition in Section 8.4.1.2 in of the *PCI Express Base Specification* (see Reference Documents).

Table 5-6 shows the loss range of the initial calibration channel for EDR25-SR PHY.

Table 5-6: Calibration Channel IL Limits for CCIX-SR

ESM Data Rate	$F_{LOW\text{-}IL\text{-}MIN}$	$F_{LOW\text{-}IL\text{-}MAX}$	$F_{HIGH\text{-}IL\text{-}MIN}$	$F_{HIGH\text{-}IL\text{-}MAX}$
20.0 GT/s Root Port	1.5 dB @ 1 GHz	2.5 dB @ 1 GHz	12.5 dB @ 10 GHz	13.5 dB @ 10 GHz
20.0 GT/s Non-Root Port	2.0 dB @ 1 GHz	3.0 dB @ 1 GHz	18.0 dB @ 10 GHz	19.0 dB @ 10 GHz
25.0 GT/s Root Port	1.5 dB @ 1 GHz	2.5 dB @ 1 GHz	12.5 dB @ 12.5 GHz	13.5 dB @ 12.5 GHz
25.0 GT/s Non-Root Port	2.0 dB @ 1 GHz	3.0 dB @ 1 GHz	18.0 dB @ 12.5 GHz	19.0 dB @ 12.5 GHz

Note1: Calibration channel plus RX reference package is 22dB nominally at Nyquist frequency which is informative.

The impedance target of RX Test Board topology except for Replica channel is 85ohm differential and 42.5ohm single-ended with +/-5% or better.

5.2.3.6.3 Post Processing Procedures

Section 8.4.1.3 of the *PCI Express Base Specification* (see Reference Documents) is incorporated by reference.

5.2.3.6.4 Behavioral Rx Package Models

Separate package loss models are required to be used to represent the root complex device and the end point device in Figure 5-8. The effective package loss reuses the definition in Section 8.3.3.11 of the *PCI Express Base Specification* (see Reference Documents).

If the actual Rx package performance is worse than that of the behavioral package, then the actual package models are permitted to be used. If the actual package models are used, the calibration channel must be adjusted such that the total Rx test channel loss described in Figure 5-8 including the embedded actual package must be same with the loss including the behavioral package model. Note that form factor overall requirements still need to be met. The Rx package performance is assessed using the methodology in Section 8.5.1.2 of the *PCI Express Base Specification* (see Reference Documents).

5.2.3.6.5 Behavioral CDR Model

Post processing should include a behavioral CDR model. EDR25-SR PHY has a first order CDR transfer function with 10MHz loop bandwidth, as described in Section 8.4.1.5 of the *PCI Express Base Specification* (see Reference Documents).

5.2.3.6.6 Behavioral CTLE

EDR25-SR PHY has a first order CTLE with fixed poles and adjustable DC gain. For details on CTLE, refer to Section 5.2.6.1.2.

5.2.3.6.7 Behavioral DFE

EDR25-SR PHY defines four taps DFE with the combination of a first order CTLE. For details on DFE tap size, refer to Table 5-11.

5.2.3.7 Calibration Stress Eye for Jitter Tolerance

The stress eye for jitter tolerance test needs to be calibrated to present the receiver with the worst-case situation. EDR High Speed must follow the same methodology to calibrate the stressed eye for jitter tolerance test in Section 8.4.2 of the *PCI Express Base Specification* (see Reference Documents).

The reference receiver for the calibration procedures is defined in Section 5.2.6.

- Calibrated EH / EW = 30mV / 0.3UI

Table 5-7: Calibration Parameters for Jitter Tolerance Test

Characteristic	Symbol	Condition	20 GT/s	25 GT/s	Unit
Eye Width at TP2P	T_{RX-ST}		0.3	0.3	UI
Eye Height at TP2P	V_{RX-ST}		30	30	mV

Characteristic	Symbol	Condition	20 GT/s	25 GT/s	Unit
Random Jitter	$T_{RX\text{-}ST\text{-}RJ}$	Note 1,2	0.75 (max)	0.6 (max)	ps rms
Differential Noise	$V_{RX\text{-}DIFF\text{-}INT}$	Adjust to control EH Frequency = 2.1GHz Note 2, 3	14	14	mVpp
Common mode Noise	$V_{RX\text{-}COMM\text{-}INT}$	Defined as a single tone at 120MHz. Measured at TP2 without post-processing Note 4	150	150	mVpp
Minimum RX calibration channel loss	$ISI_{RX\text{-}ISI\text{-}SR\text{-}min}$	The minimum differential loss for SR calibration channel from TP1 to TP2P	16	16	dB
Maximum RX calibration channel loss range	$ISI_{RX\text{-}ISI\text{-}SR\text{-}max}$	The maximum differential loss for SR calibration channel from TP1 to TP2P	24	24	dB

Note1: Rj is measured at TP1. The low frequency limit of Rj may be between 1.5MHz and 10MHz, and the upper limit is 1.0GHz.

Note2: Both $T_{RX\text{-}ST\text{-}RJ}$ and $V_{RX\text{-}DIFF\text{-}INT}$ are limited to prevent the stressed eye from excessive amount of jitter or noise distortion.

Note3: The frequency of $V_{RX\text{-}DIFF\text{-}INT}$ is chosen to be slightly higher than the first pole of the reference CTLE.

Note4: Common mode noise is turned off during $T_{RX\text{-}ST}$ and $V_{RX\text{-}ST}$ calibration and then turned on for the stressed eye jitter test.

5.2.3.7.1 Procedure to Calibrate Stressed Eye

Same as that described in Section 8.4.2.1 of the *PCI Express Base Specification* (see Reference Documents), the goal of calibrating a stressed voltage/jitter eye is to present the receiver under test with simultaneously worst case margins whose distortion characteristics are similar to an eye produced by a real channel. Much of the distortion consists of the ISI produced by the calibration channel. Incremental changes of differential voltage from nominal values may be used to adjust the EH while the EW is post processed to evaluate the worst case of Sj that will be used onto RX DUT test, which is different from the *PCI Express Base Specification* (see Reference Documents).

The reference point where EH/EW is defined corresponds to input to the receiver latch at 20.0 GT/s and 25.0 GT/s. Since this point is not physically accessible it is necessary to construct its equivalent by means of a post-processing procedure. A step response that has been averaged 1024 times at TP2 is first post processed in order to mathematically include the additional signal distortion caused by the behavioral receiver package. As the averaged step response is used, the stresses of differential voltages are turned on, but the stresses of jitters will not be included and are turned off. Then the resulting signal is recovered by means of Rx equalization, and a behavioral CDR function, resulting in an equivalent eye. The requirements for the waveform post processing tool used for the EH/EW calibration are described further in Section 5.2.3.7.3.

As the calibration procedure of the signal generator output contains steps where the generator is connected directly to measurement instrumentation, the transition time of the output waveform can be very fast. Therefore, it is important that the bandwidth of instrumentation used to calibrate the generator be matched appropriately to the edge rate of the generator output. This specification requires the use of a generator whose outputs have less than 19ps rise time (20% / 80%) which also requires a minimum oscilloscope bandwidth of 32GHz. This oscilloscope bandwidth is also the minimum required bandwidth for transmitter measurements. The oscilloscope must have enough low noise floor up to the bandwidth of it and the measured step response needs to show reasonable response in the frequency domain.

The loss of combiner at 12.5GHz including cables from the output of generator to TP1 should be less than 2dB. The 3dB cutoff frequency of combiner is required to be at least 20GHz.

Note that for the eye calibration process, the Tx equalization is fixed to the preset that gives the optimal eye area with the post processing tool being used for calibration. Once the testing procedure is under way the Tx preset may be adjusted to yield the best eye margins with the DUT. For calibration at ESM rates the following process is used to calibrate the eye:

1. *Calibrate the stress values as shown below at TP1:*
    a. *Tx Differential Voltage Swing to 800 mV PP (+ 10 mV / - 0 mV).*
    b. *TX presets that specified in Table 8-1 in PCI Express Base Specification* (see Reference Documents) *within +/- 0.1 dB.*
    c. *Sj by 0.1 UI (4 ps) at 100 MHz. As the step response is used, this Sj is included to post-processing tool.*
    d. *Rj to 0.6 ps rms. As the step response is used, this Rj is included to post-processing tool.*
    e. *DMI to 14mV at TP2P. As the step response is used, this DMI is included to post-processing tool.*
2. *Connect an initial test channel with selecting the ISI length that give a loss of TP1 to TP2P:*
    a. *The loss of an initial test channel for 20GT/s: 16dB +/-0.5dB at 10GHz.*
    b. *The loss of an initial test channel for 25GT/s: 16dB +/-0.5dB at 12.5GHz.*
3. *Measure the eye diagram for each TX EQ preset using the calibrated TX EQ and select the TX EQ preset that gives the largest eye area.*

IMPLEMENTATION NOTE: REQUIRMENT OF COMBINER

To combine the signal from pattern generator and DMI noise source either RF combiner or directional coupler can be used. Figure 5-9 is the recommended frequency mask for the combiner block.

“Nominal loss” means DC or low frequency loss at less than couple hundreds kHz. Usually RF combiner has 6dB and directional coupler has 0dB.

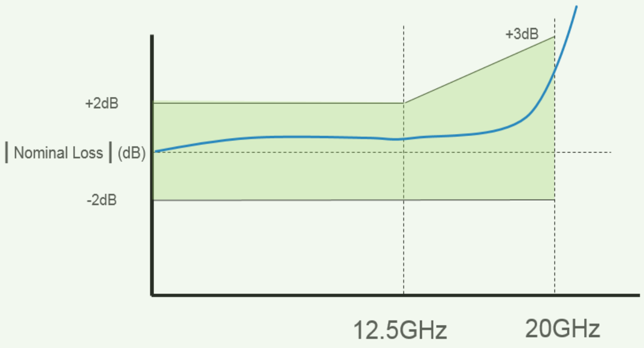


Figure 5-9: The frequency mask of combiner block

For all EH, EW and eye area measurements performed in receiver calibration the DC Gain in the reference receiver CTLE is varied over its minimum to maximum range in 0.25 dB steps. This is done to improve repeatability and accuracy in automated Rx calibration software and is only done for stressed eye calibration (not for channel compliance, etc.)

4. *Increase the calibration channel loss to the next available length/loss and measure the new eye diagram at the selected preset. Continue to increase the length/loss until either the height or width have fallen below the targets (EH – 30 mV +/-3mV, EW – 0.3 UI +/-0.0125UI) then the previous calibration channel length/loss is selected. If neither the height or width have fallen below the targets and the TP1 to TP2P loss at 10 GHz for 20 GT/s and 12.5 GHz for 25 GT/s has reached 24.0 dB then advance to the next step.*
5. *For the selected calibration channel length/loss, measure the eye diagram for each TX EQ preset and select the preset that gives the largest eye area. Note that this may be a different preset than step 3 due to the length/loss change. Also the noises in differential and common modes are calibrated as following:*
6. *For the fixed calibration channel length/loss and Tx EQ in step 5, adjust DMI$_{TP2P}$ and Voltage Swing to make final adjustments to the eye by sweeping them through the following ranges:*
   a. *Sj in post-processing tool – 4 to 8ps*
   b. *DMI$_{TP2P}$ in post-processing tool – Equivalent range with (10 mV to 25 mV) DMI at TP2*
   c. *Differential Voltage Swing at generator – 720 mV to 800 mV at TP1*

The range of DMI$_{TP2P}$ in post-processing tool is calculated by DMI range at TP2 (10mV ~ 25mV) and the gain of adapted CTLE. The following table can be used to map DMI and DMI$_{TP2P}$

$$DMI_{TP2P} = DMI * 10^{((Gain\_dB + Rxpkg\_dB) / 20)}$$

Table 5-8: The gain of adapted CTLE at 2.1GHz (dB)

Adc	fp1 = 2 GHz
-12 dB	-2.592416763
-11 dB	-2.531968322
-10 dB	-2.457045407
-9 dB	-2.364524550
-8 dB	-2.250782177
-7 dB	-2.111698919
-6 dB	-1.942708404

7. *If the final calibrated Sj is less than 0.1 UI then the Rj level in post processing tool is reduced so the eye width meets the target eye width with 0.1 UI of Sj in post processing tool*
8. *If there are multiple combinations of DMI and Voltage Swing that give valid solutions first pick the combination that is closest to the target eye width (0.3 UI). The selected values must give a mean eye height and width (over at least 5 measurements – exact number of 15 measurements needed for stable values will depend on lab set-up and tools) within the following ranges at BER 1E-12:*
    a. *Eye height – 30 mV +/- 3 mV*
    b. *Eye width – 0.3 UI +/- 0.0125 UI*

Figure 5-10 shows block diagram for calibrating the stressed eye for EDR25-SR PHY. If the calibration set up with the specific pattern generator and the oscilloscope cannot achieve calibration at the specific loss range, this set up is not valid and cannot be used for RX calibration procedures. Table 5-7 shows the loss range of RX calibration for EDR25-SR PHY.

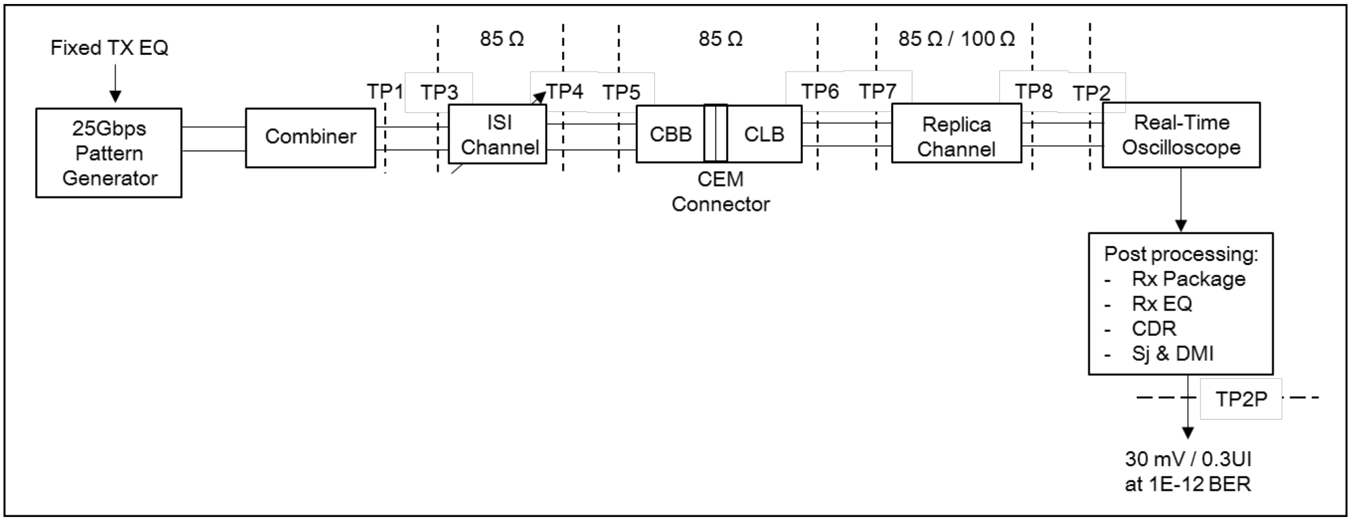


Figure 5-10: Layout for Calibrating the Stressed Jitter Eye at ESM RatesProcedure for Testing Rx DUT

5.2.3.7.2 Procedure for Testing Rx DUT

Once a calibrated EH and EW have been obtained, the cable connected at TP7 in Figure 5-10 is moved to TP9 so that the Rx DUT is connected to the far end of calibration channel as shown in Figure 5-11. The cable from TP8 to TP2 during the calibration is not included to the Rx DUT test. So, the cable from TP8 to TP2 should be picked carefully to have enough low loss at 10 GHz for 20 GT/s and 12.5 GHz for 25 GT/s to be ignored or deembedded during the calibration process.

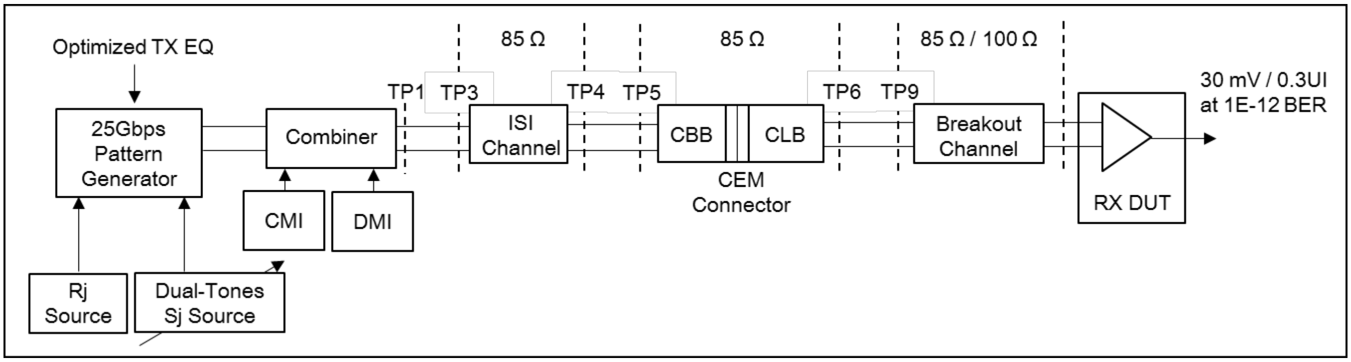


Figure 5-11: Layout for Testing Rx DUT

The Tx equalization may then be optimized with the assumption that the DUT Rx will also optimize its equalization. Sj is set to an initial value of 0.1 UI at 100 MHz and the receiver CDR must achieve lock. The 100 MHz Sj initial tone is removed and then the appropriate swept Sj profile is tested. Also an additional Sj tone at 210 MHz is present for all testing. The amplitude of this additional tone is equal to the amplitude of the 100 MHz Sj required to achieve the target eye width minus 0.1 UI. If the calibration Sj level was less than 0.1 UI then no additional tone at 210 MHz is used. Receivers operating the CC Refclk mode shall utilize the Sj profile shown in Figure 8-36 in *PCI Express Base Specification* (see Reference Documents).

5.2.3.7.3 Waveform Post Processing Tool Requirements

The waveform post processing tool to calibrate EH/EW for RX stressed eye test must be consistent with the channel compliance methodology in Section 8.5.1 of the *PCI Express Base Specification* (see Reference Documents). This tool needs to be used during RX stressed eye calibration and there are few notes for using the tool as follows:

- ✓ *The reference package model of Rx need to be embedded by either the oscilloscope or the post processing tool.*
- ✓ *All step measurements are done at TP2.*
- ✓ *A step pattern with 512 ones and zeroes is captured through the test channel by averaging 1024 times on a real time oscilloscope. The step is saved with an x-axis resolution of 1 ps or less to be used as the transmit waveform for the channel compliance methodology. The step pattern is captured with the calibrated presets of TX EQ enabled at Generator. When the measured step is converted to the frequency domain by using appropriate mathematical function, it is required to show reasonable frequency domain response*
- ✓ *A captured step is simulated in the post processing tool with TX EQ of the tool itself disabled. The equivalent jitters and DMI are set in the tool at the same time.*
- ✓ *For the preset that gives the largest eye area with the waveform post processing tool the EH and EW @ 1E-12 BER must match 30 mV +/- 10% and 0.3 UI +/- 0.0125 UI.*

A general flow for using post processing tool is shown in Figure 5-12.

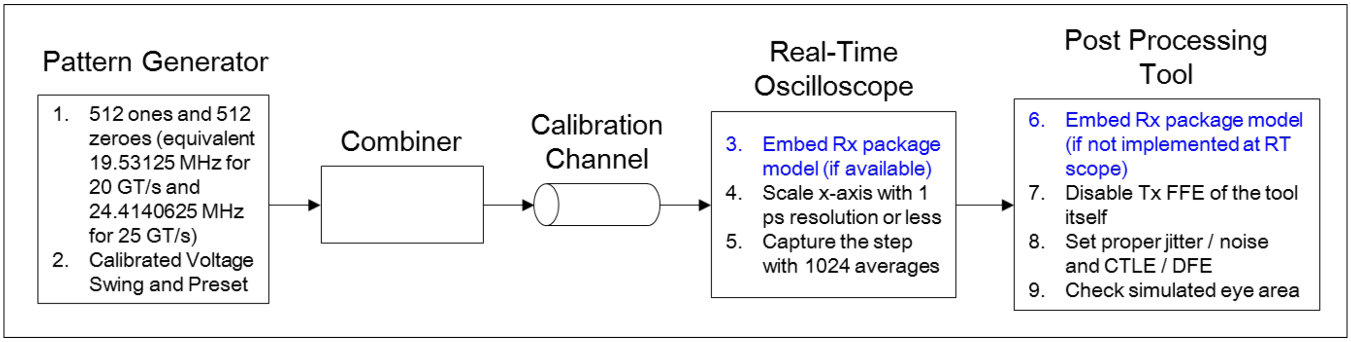

Pattern Generator
1. 512 ones and 512 zeroes (equivalent 19.53125 MHz for 20 GT/s and 24.4140625 MHz for 25 GT/s)
2. Calibrated Voltage Swing and Preset
Combiner
Calibration Channel
Real-Time Oscilloscope
3. Embed Rx package model (if available)
4. Scale x-axis with 1 ps resolution or less
5. Capture the step with 1024 averages
Post Processing Tool
6. Embed Rx package model (if not implemented at RT scope)
7. Disable Tx FFE of the tool itself
8. Set proper jitter / noise and CTLE / DFE
9. Check simulated eye area

Figure 5-12: General Flow for Using Post Processing Tool

5.2.3.8 Receiver Refclk Modes

CCIX supports Common Clock (CC) mode only. Figure 5-13 shows the Refclk connection for a receiver. A single Refclk source drives both the external signal source and DUT.

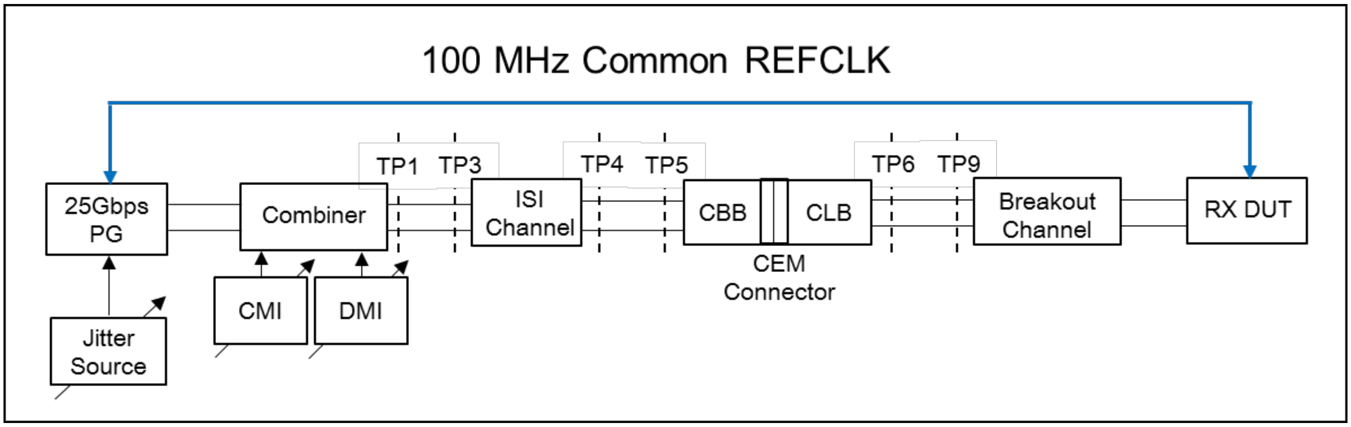


Figure 5-13: Layout for Jitter Testing for Common Refclk Rx

5.2.4 PCIe-Specific Specifications

5.2.4.1 Electrical Idle

Section 8.4.5.5 of the *PCI Express Base Specification* (see Reference Documents), is incorporated by reference.

5.2.4.2 Receiver Detection

Section 8.4.5.7 of the *PCI Express Base Specification* (see Reference Documents), is incorporated by reference.

5.2.4.3 Receiver Margining

Section 8.4.4 of the *PCI Express Base Specification* (see Reference Documents) is incorporated by reference.

5.2.4.4 Link Training for TX EQ with Back Channel

Section 4.2.3 of the *PCI Express Base Specification* (see Reference Documents), is incorporated by reference.

5.2.5 Reference Clock Specification

The reference clock specification in EDR25-SR PHY reuses the requirements for common reference clock (CC) in Section 8.6 of the *PCI Express Base Specification* (see Reference Documents).

5.2.5.1 Reference Clock Electrical Specification

Table 5-9 shows the specification of reference clock.

Table 5-9: Refclk Specification

Symbol	Description	Min.	Max.	Unit
F_{refclk}	Reference Clock frequency	99.97	100.03	MHz
F_{SSC}	SSC Frequency Range	30	33	kHz
$T_{SSC\text{-}FREQ\text{-}DEVIATION}$	SSC Deviation	-0.5	0	%
T_{REFCLK}	Residual Jitter Limit for CC using 2 port without system noise		0.35	ps rms
T_{REFCLK_noise}	Residual Jitter Limit which can be used channel simulations to account for additional noise in a real system		0.5	ps rms
$T_{TRANSPORT\_DELAY}$	TX-RX Transport delay		12	ns

5.2.5.2 Reference Clock Compliance Measurement

EDR25-SR PHY must use same methodology to calculate the residual jitter of Refclk for the compliance in Section 8.6 of the *PCI Express Base Specification* (see Reference Documents).

The jitter measurement of Refclk shall be made with a capture at least 100,000 clock cycles by a real-time oscilloscope with a sample rate of 20GS/s or greater.

Broadband oscilloscope noise must be minimized in the measurement. The PLL and CDR filters in Table 5-10 should be applied to time domain raw data measurement.

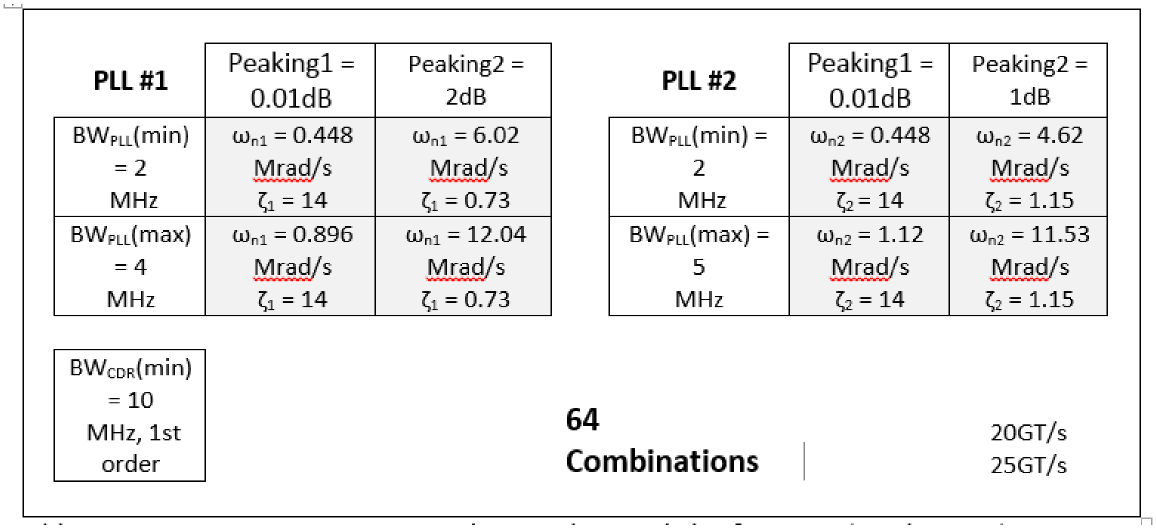
SSC spurs from the fundamental and harmonics are removed up to a cutoff frequency of 2MHz taking care to minimize removal of any non-SSC content.

5.2.5.3 CDR and PLL Bandwidth and Peaking for Reference Clock Compliance

CDR and PLL BW and Peaking for reference clock measurement need to be defined and EDR25-SR PHY reuses the methodology in Sections 8.6.5.1 and 8.6.5.2 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-10 shows CDR and PLL BW and peaking of reference TX PLL, RX PLL and CDR for EDR25-SR PHY.

Table 5-10: Common Refclk PLL and CDR Characteristics for EDR25-SR PHY



PLL #1	Peaking1 = 0.01dB	Peaking2 = 2dB
BW_{PLL}(min) = 2 MHz	ω_{n1} = 0.448 Mrad/s ζ_1 = 14	ω_{n1} = 6.02 Mrad/s ζ_1 = 0.73
BW_{PLL}(max) = 4 MHz	ω_{n1} = 0.896 Mrad/s ζ_1 = 14	ω_{n1} = 12.04 Mrad/s ζ_1 = 0.73

PLL #2	Peaking1 = 0.01dB	Peaking2 = 1dB
BW_{PLL}(min) = 2 MHz	ω_{n2} = 0.448 Mrad/s ζ_2 = 14	ω_{n2} = 4.62 Mrad/s ζ_2 = 1.15
BW_{PLL}(max) = 5 MHz	ω_{n2} = 1.12 Mrad/s ζ_2 = 14	ω_{n2} = 11.53 Mrad/s ζ_2 = 1.15

BW_{CDR}(min) = 10 MHz, 1st order

64 Combinations

20GT/s
25GT/s

5.2.6 Channel Compliance

5.2.6.1 Channel Compliance Methodology

5.2.6.1.1 Using Internal EW/EH with reference receiver

The EDR25-SR PHY must follow same methodology to check the channel compliance as defined in Section 8.5.1 of the *PCI Express Base Specification* (see Reference Documents).

Figure 5-14 shows the key components and processes of channel tolerance at ESM rates. This flow must be implemented in channel tolerance tools.

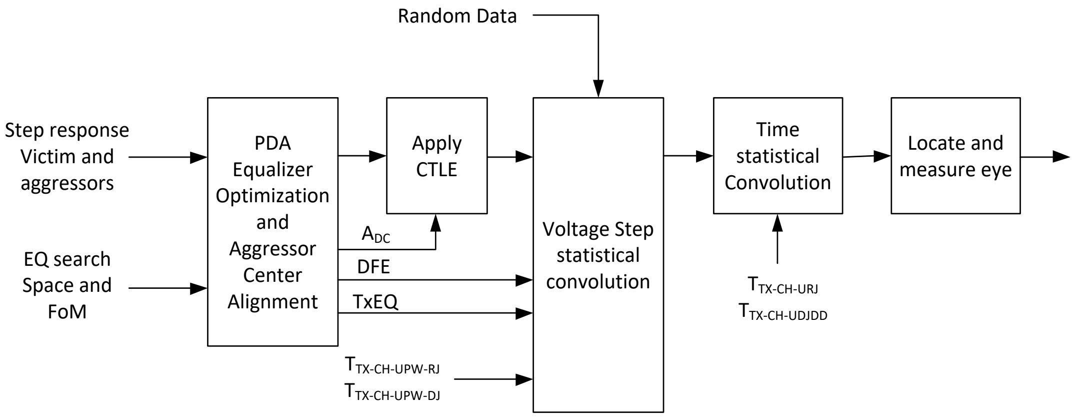

Random Data
Step response Victim and aggressors
EQ search Space and FoM
PDA Equalizer Optimization and Aggressor Center Alignment
Apply CTLE
A_{DC}
DFE
TxEQ
$T_{TX-CH-UPW-RJ}$
$T_{TX-CH-UPW-DJ}$
Voltage Step statistical convolution
Time statistical Convolution
$T_{TX-CH-URJ}$
$T_{TX-CH-UDJDD}$
Locate and measure eye

Figure 5-14: Flow Diagram for Channel Tolerance at ESM Rates

Table 5-11 shows reference receiver parameters for EDR25-SR channel compliance. $V_{RX-CH-EH}$ is eye height for the nominal channel.

Table 5-11: Reference Receiver Parameters for Channel Compliance

Symbol	Description	Value	Unit	Note
$V_{RX-CH-EH}$	Eye height at nominal channel	30 (min)	mVpp	Eye height at BER=10-12 Note1
$T_{RX-CH-EW}$	Eye width at zero crossing	0.3 (min)	UI	Eye width at BER=10-12
$T_{RX-DS-OFFSET}$	Peak EH offset from UI center	+/- 0.1	UI	
R_{H1_H0}	H1 / H0 Ratio	0.7 (max)		The ratio between data cursor and the 1st post cursor
$V_{RX-DFE-D1}$	Range for DFE d1 coefficient	+/-30	mV	
$V_{RX-DFE-D2}$	Range for DFE d2 coefficient	+/-20	mV	
$V_{RX-DFE-D3}$	Range for DFE d3 coefficient	+/-20	mV	
$V_{RX-DFE-D4}$	Range for DFE d4 coefficient	+/-20	mV	

Note1: The variation of channel must be considered in the simulation for channel compliance

5.2.6.1.2 Behavioral CTLE

The EDR25-SR PHY behavioral Rx equalization defines a 1st order CTLE with fixed LF and HF poles, and an adjustable DC gain (A_{DC}) specified to the family of curves shown in Figure 5-15. A_{DC} is adjustable over a minimum range of -6 to -12dB in steps of 1.0dB. Transfer function of CTLE is shown in Equation 1.

$$\mathbf{H(s)} = g_{dc}\,\frac{(\frac{s}{\omega_Z}+1)}{\left(\frac{s}{\omega_{P1}}+1\right)*(\frac{s}{\omega_{P2}}+1)}$$

$$\omega_{P1} = pole1 = 2\pi * 2GHz$$

$$\omega_{P2} = pole2 = 2\pi * 25\ GHz$$

$$\omega_Z = Zero\ Frequency = \omega_{P1} * g_{dc}$$

$$A_{DC} = from\ (-12dB)\ to\ (-6\ dB)\ in\ steps\ of\ 1dB$$

$$g_{dc} = \ 10^{A_{DC}/20}$$

Equation 1: Transfer Function for CCIX Behavioral CTLE of EDR25-SR PHY

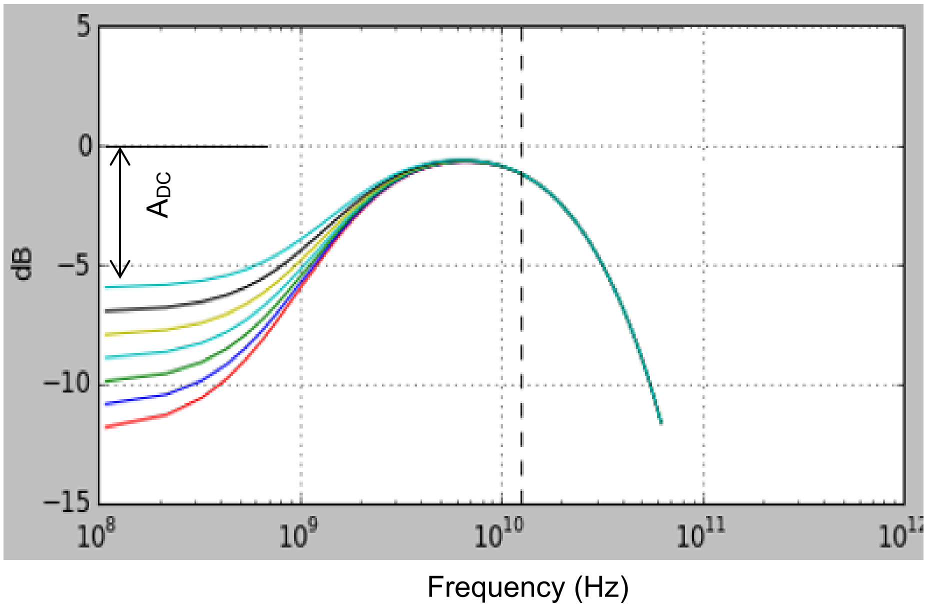


Figure 5-15: Loss Curves for CCIX Behavioral CTLE of EDR25-SR PHY

5.2.6.1.3 Behavioral Transmitter and Receiver Package Models (Informative)

The behavioral package models are defined to represent the combined die and package loss. Table 5-12: Package Model Capacitances has the parasitic capacitances at die pin and BGA ball.

Table 5-12: Package Model Capacitances

Symbol	Description	Value	Unit	Note
C_{pin}	BGA Ball	0.25	pF	
C_{pad}	Die-pad	0.25	pF	

The package model which is used during SR channel compliance procedures must include the parasitic capacitances in Table 5-12 or the equivalent model.

5.2.6.1.4 Behavior Transmitter Parameters

Table 5-13 shows jitter / voltage parameters of transmitter for channel compliance.

Table 5-13: Jitter/Voltage Parameter of Behavioral Transmitter

Symbol	Description	20 GT/s	25 GT/s	Unit	Note
$T_{TX\text{-}CH\text{-}URJ}$	TX uncorrelated RJ	0.494	0.395	ps rms	
$T_{TX\text{-}CH\text{-}UDJDD}$	TX uncorrelated DjDD	3	2.4	ps p-p	
$T_{TX\text{-}CH\text{-}UPW\text{-}RJ}$	Uncorrelated PW RJ	0.607	0.486	ps rms	
$T_{TX\text{-}CH\text{-}UPW\text{-}DJ}$	PW DDJ	4	3.2	ps p-p	Include parasitic die pad capacitance

5.2.6.2 Frequency Domain Response for Channel Compliance (Informative)

These frequency domain characteristics of channel are informative specifications.

The frequency domain response for EDR-SR PHY reuses the channel compliance of CEI-28G-SR for frequency domain specification of channel. Table 5-14 defines the channel characteristics.

Table 5-14: Frequency Domain Channel Characteristic

	Symbol	Description
Measured Channel Parameters	IL(f)	Differential insertion loss, SDD21 (dB)
	RL1(f)	Differential input return loss, SDD11 (dB)
	RL2(f)	Differential output return loss, SDD22 (dB)
	NEXTm(f)	Differential near-end crosstalk loss (m-th aggressor) (dB)
	FEXTn(f)	Differential far-end crosstalk loss (n-th aggressor) (dB)
Calculated Channel Parameters	$IL_{fitted}(f)$	Fitted insertion loss (dB)
	ILD(f)	Insertion loss deviation (dB)
	ICN(f)	Integrated crosstalk noise (mV, RMS)
	ILD(rms)	RMS value of the insertion loss deviation (dB)

5.2.6.2.1 Insertion Loss Between Reference Point T_{die} and R_{die} (Informative)

EDR High Speed has informative insertion loss definition between die pads. The insertion loss between reference point T_{die} and R_{die} is up to 22dB at Nyquist frequency.

Table 5-15 shows the detail break-down of insertion loss between die pads is shown in below sub-sections.

Table 5-15: Insertion Loss Break-Down

Symbol	Description	Min	Max	Unit
$IL_{link\_die\_die}$	Differential Insertion loss of total link from T_{die} to R_{die} at Nyquist frequency		-22	dB
IL_{tx_pkg}	Differential Insertion Loss of root complex TX Package from T_{die} to T_{ball} at Nyquist frequency		-9	dB
IL_{rx_pkg}	Differential Insertion Loss of non-root complex device RX Package from R_{die} to R_{ball} at Nyquist frequency		-3.5	dB
IL_{board}	Differential Insertion Loss from T_{ball} to R_{ball} at Nyquist frequency		-9.5	dB

The boundary of insertion loss at 25.0 GT/s between reference point T_{die} and R_{die} is given in Figure 5-16 and Equation 2. The die parasitic capacitance must be included for this insertion loss.

$$IL_{max} = \begin{cases} -(0.0388 + 3.065\sqrt{\frac{f \times 25}{f_b}} + 0.89\frac{f \times 25}{f_b}), & 50MHz \le f < \frac{f_b}{2} \\ 22 - 3.52\frac{f \times 25}{f_b}\,, & \frac{f_b}{2} \le f \le f_b \end{cases}$$

$$IL_{min} = \begin{cases} 0\,, & 50MHz \le f \le 1GHz \\ -\frac{1}{3}(f-1)\,, & 1GHz < f \le 17.5GHz \\ -5.5, & 17.5GHz < f \le f_b \end{cases}$$

$$Note: f_b \ is \ 25G \ or \ 20G$$

Equation 2: Channel Insertion loss mask for EDR25-SR Link

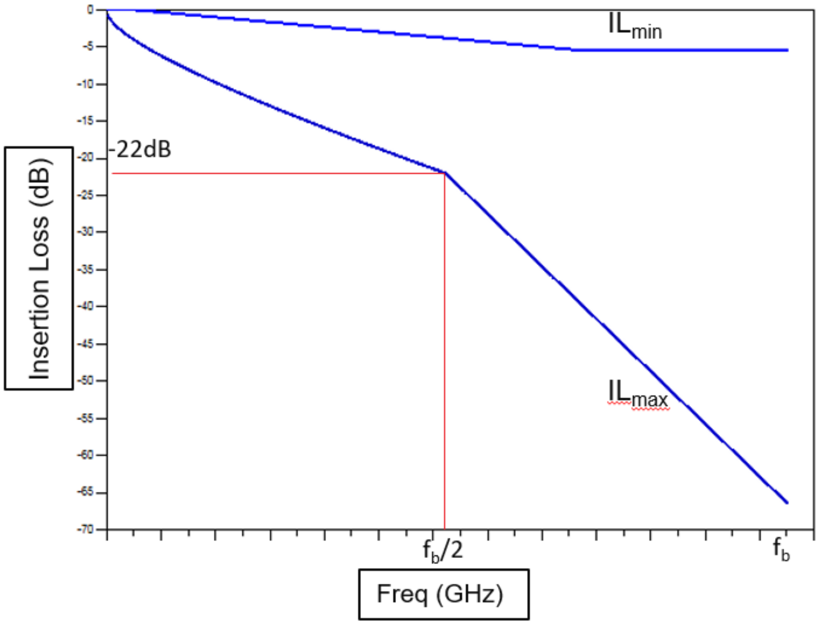


Figure 5-16: Insertion Loss Mask for EDR25-SR Link

5.2.6.2.2 Differential Return Loss of Channel (Informative)

The channel differential Return loss from T_{ball} to R_{ball} is defined by Equation 3.

$$RL_{max} = \begin{cases} -12\ dB, & 50MHz \leq f < \frac{f_b}{4} \\ -12 + 15\ log_{10}\left({}^{4f}/_{f_b}\right), & \frac{f_b}{4} \leq f \leq f_b \end{cases}$$

Equation 3: Channel Return Loss Mask for EDR25-SR Link

Figure 5-17 shows the Return Loss Plots with Limit Lines.

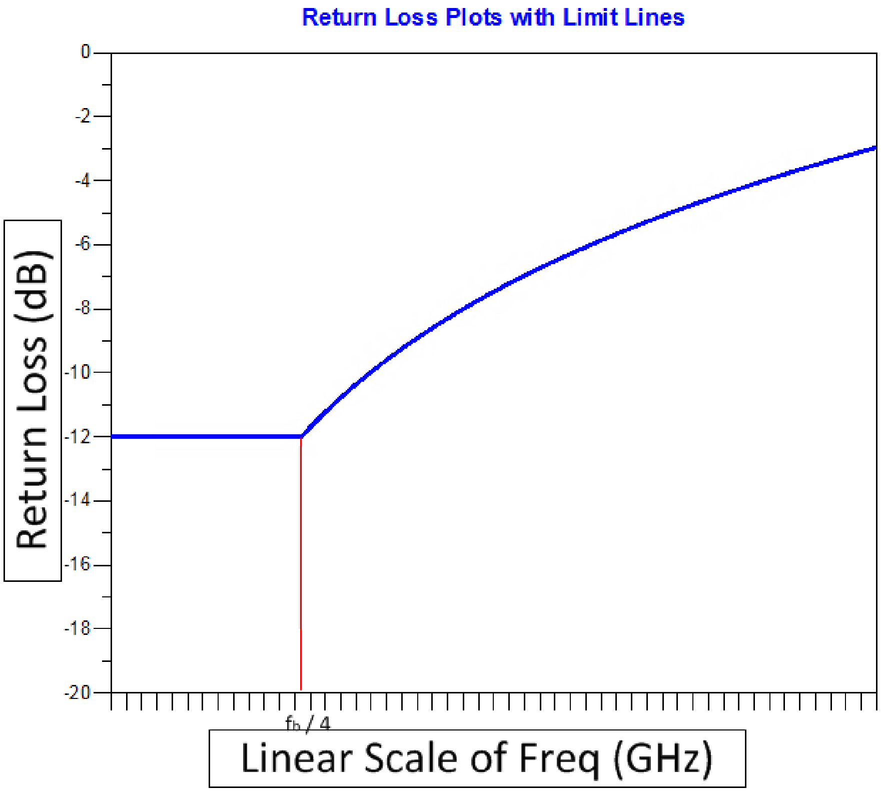


Figure 5-17: Channel Differential Return Loss Mask for EDR25-SR Link

5.2.6.2.3 Insertion Loss Deviation (Informative)

The insertion loss deviation, ILD, is the difference between the measured insertion IL and fitted insertion loss IL_{fitted}. For fitted insertion loss definitions, refer to CEI-28G-SR Section 12.2.1.1 (see Reference Documents).

$$ILD = IL - IL_{fitted}$$

The insertion loss deviation, ILD, is required to be within the ranges which is defined by the equations (10-4) and (10-5) in CEI-28G-SR (see Reference Documents).

Figure 5-18 shows the ILD range.

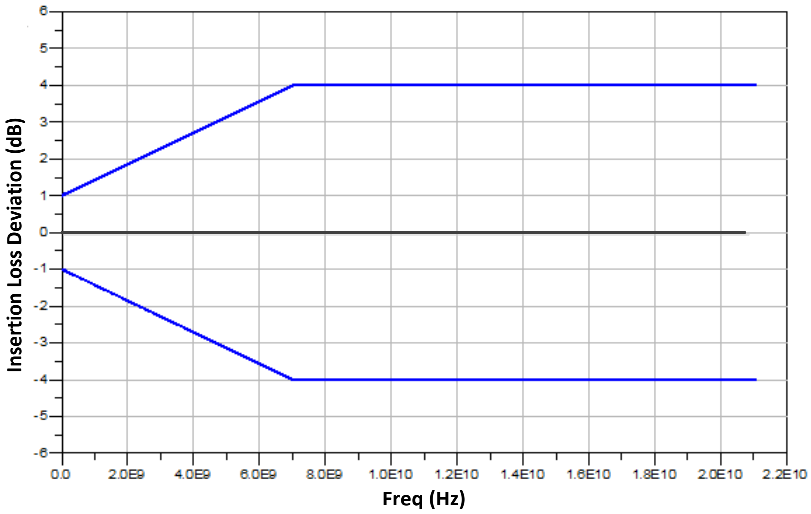


Figure 5-18: Insertion Deviation Mask

ILD_{rms} is the RMS value of the ILD curve and calculated by the equations in CEI-28G-SR Section 10.2.6.4. This is required to be less than $0.3dB_{rms}$ for valid channels.

5.2.6.2.4 Integrated Crosstalk Noise (Informative)

The integrated Crosstalk Noise, ICN, is calculated by using the integrated crosstalk noise method in CEI-28G-SR Section 10.2.6.6 (see Reference Documents).and the parameters of Table 5-16. It is required to be in the lower than max allowed ICN at given insertion loss in Figure 5-19. ICN needs to be calculated between reference point T_{die} and R_{die}.

Table 5-16: Channel Integrated Crosstalk Aggressor Parameters

Parameter	Symbol	Value	Units
Near-end aggressor peak to peak differential output amplitude	A_{nt}	1200	mVpp
Far-end aggressor peak to peak differential output amplitude	A_{ft}	1200	mVpp
Near-end aggressor 20 to 80% rise and fall times	T_{nt}	8	ps
Far-end aggressor 20 to 80% rise and fall times	T_{ft}	8	ps

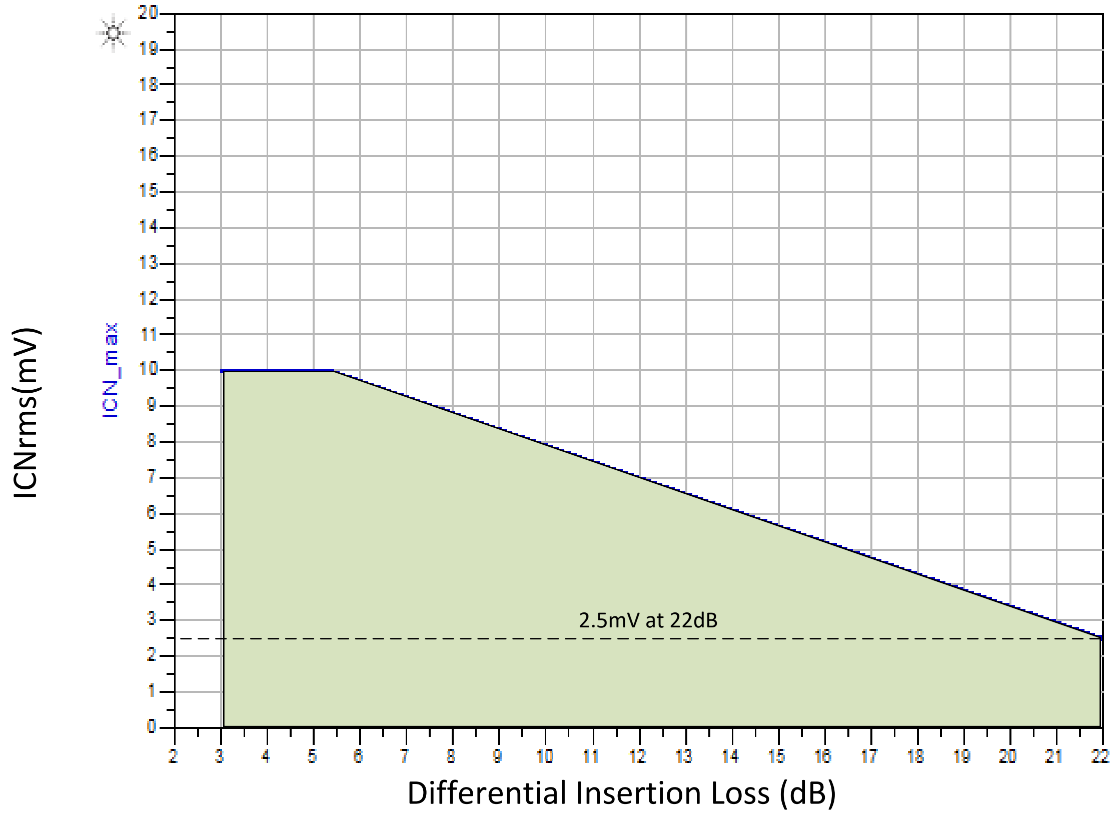


Figure 5-19: ICN$_{rms}$ Mask vs Insertion Loss

5.3 EDR25-LR Electrical Specification

5.3.1 General Specification

The EDR25-LR electrical specification applies to the long reach application of ESM Data rates in EDR PHY which are 20GT/s and 25GT/s.

5.3.1.1 Line Rate

EDR25-LR PHY must support 20GT/s and 25GT/s.

5.3.1.2 Line Coding

EDR25-LR PHY uses 128b/130b coding, same as PCIe Gen3 and Gen4.

5.3.1.3 Crosstalk

Crosstalk defined in EDR25-LR PHY specification is contributed from the elements such as connector and PCB of component-to-component channel. It should be minimized and at least less than the specified number which is defined informatively through Integrated Crosstalk Noise (ICN) from CEI-25G-LR standard.

5.3.1.4 Baud Rate Tolerance

EDR25-LR PHY is required to operate with common clock and tolerate up to +/-300 ppm from the nominal baud rate without spread spectrum clock.

Spread Spectrum Clock (SSC) is allowed for common clock operation.

Separate Reference Clocks with Independent SSC (SRIS) system will be added in future revision as an optional feature.

5.3.1.5 AC Coupling Capacitor

EDR25-LR link requires AC coupling only. The AC coupling cap value is 176nF to 265nF and its placement is determined by the form factor specification.

5.3.1.6 Target Bit Error Rate

The target Bit error rate in EDR25-LR link is 1.0E-12. All of jitter calculation and compliance test must use 1.0E-12 target BER.

5.3.1.7 Reference Model

The EDR25-LR common reference model is defined in Figure 5-20. EDR25-LR link has four reference points at both die pads and BGA balls. The socket is included in the package.

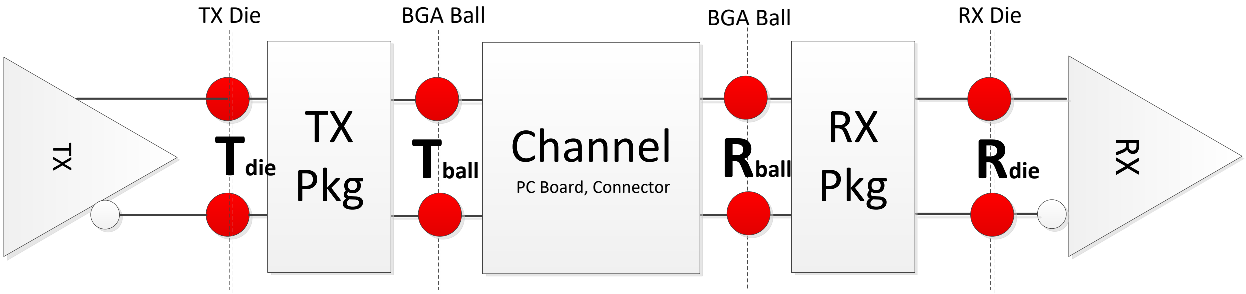


Figure 5-20: Reference Model of EDR25-LR Link

Figure 5-21 shows reference points on a downstream direction with riser of EDR25-LR link.

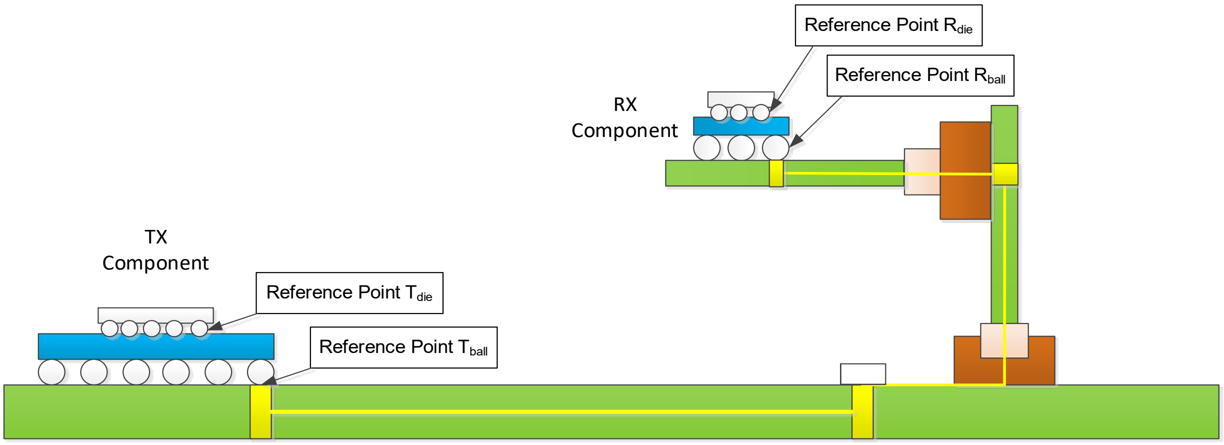

Reference Point R_{die}
Reference Point R_{ball}
RX Component
TX Component
Reference Point T_{die}
Reference Point T_{ball}

Figure 5-21: Reference Points on a downstream direction of EDR25-LR link

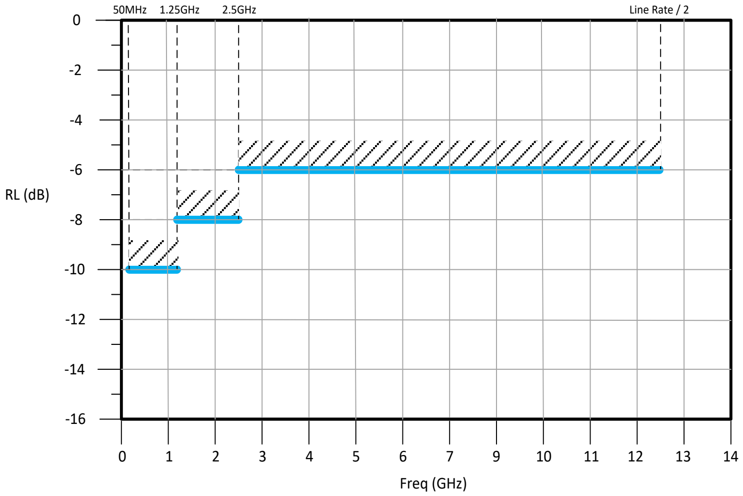
5.3.2 Transmitter Specification

5.3.2.1 Driver Termination

The EDR25-LR PHY does not specify differential source termination at the driver as long as the TX return loss specification is met.

5.3.2.2 Differential Return Loss of Transmitter

The transmitter differential return loss at T_{ball} including package and socket is given in Figure 5-22.

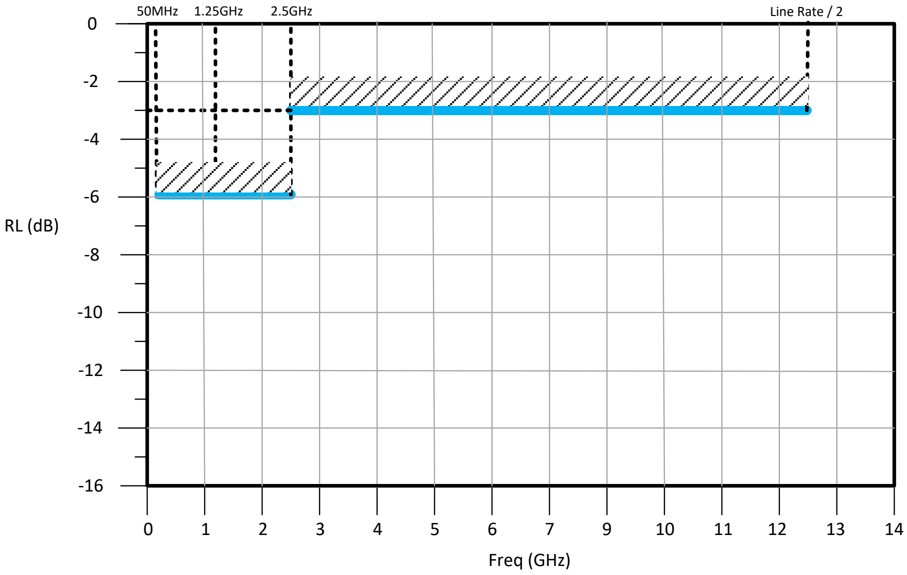


Note: The termination of measurement is 100ohm

Figure 5-22: Transmitter Differential Return Loss Mask for EDR25-LR PHY

5.3.2.3 Common Mode Output Return Loss

The common mode return loss for T_{ball} including package and socket is given in Figure 5-23. To attenuate noise and absorb even/odd mode reflections, both transmitter and receiver is required to satisfy the Common Mode Return Loss requirement.



Note: The termination of measurement is 25ohm

Figure 5-23: Transmitter Common Return Loss Mask for EDR25-LR PHY

5.3.2.4 Transmitter PLL Bandwidth and Peaking

Table 5-10 defines transmitter PLL bandwidth and peaking.

5.3.2.5 TX Voltage Parameters

5.3.2.5.1 Signal Definition

The signal definition in EDR25-LR PHY reuses voltage level definition in Section 8.3.2 of the *PCI Express Base Specification* (see Reference Documents).

5.3.2.5.2 Transmitter AC Specification

The transmitter electrical specifications are given in Table 5-17.The measurement of these parameters in EDR25-LR PHY must support the methodology in Section 8.3 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-17: The Electrical Specification at reference point T_{ball}

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Differential peak-peak Tx voltage swing for full swing operation	$V_{TX\text{-}DIFF\text{-}PP}$	Measured with compliance test load	800		1300	mVpp
Differential peak-peak Tx voltage swing for low swing operation	$V_{TX\text{-}DIFF\text{-}PP\text{-}LOW}$	Measured with compliance test load	400		1300	mVpp
Minimum voltage swing during EIEOS for full swing signaling	$V_{TX\text{-}EIEOS\text{-}FS}$		250			mVpp
Minimum voltage swing during EIEOS for reduces swing signaling	$V_{TX\text{-}EIEOS\text{-}RS}$		232			mVpp
Pseudo package loss of root complex device (Informative)	$ps21_{TX\text{-}ROOT\text{-}DEVICE}$				7.0	dB
Pseudo package loss of non-root complex device (Informative)	$ps21_{TX\text{-}NON\text{-}ROOT\text{-}DEVICE}$				2.0	dB

Note: All of measurement should be referenced to 100ohm termination

5.3.2.5.3 Transmitter Equalization

The EDR25-LR transmitter must support the equalization by using 3-tap FIR.

The transmitter equalization of EDR25-LR PHY reuses the specification of 16.0GT/s in Section 8.3.3.2 of the *PCI Express Base Specification* (see Reference Documents).

The control feature of each cursor is required for re-driver option.

5.3.2.5.4 TX Equalization Presets

The EDR25-LR transmitter must support the full range of presets in Section 8.3.3.3 of the *PCI Express Base Specification* (see Reference Documents).

The initial TX Preset must be set to achieve a minimum BER of 1.0E-4 for phase 1 of Recovery Equalization. The methodology in Section 4.7.6 of the *PCIe 4.0 CEM Specification* (see Reference Documents) is a good practice to achieve the minimum BER when the inter-changeable Add-In Card slot is implemented for a CCIX system.

5.3.2.5.5 Measuring Presets at EDR25-LR PHY

The methodology to measure transmitter Presets is defined in Section 8.3.3.5 of the *PCI Express Base Specification* (see Reference Documents), where 16.0GT/s Preset measurement is specified. The presets shall be measured at EDR25-LR PHY rates.

5.3.2.5.6 EIEOS and $V_{TX\text{-}EIEOS\text{-}FS}$ and $V_{TX\text{-}EIEOS\text{-}RS}$ Limit

The EIEOS patterns at EDR25-LR PHY consists of 32 consecutive ones followed by 32 consecutive zeros, where the pattern is repeated for a total of 128 UI.

Measuring $V_{TX\text{-}EIEOS\text{-}FS}$ and $V_{TX\text{-}EIEOS\text{-}RS}$ is defined in Section 8.3.3.9 in the *PCI Express Base Specification* (see Reference Documents) where 16.0GT/s measurement is specified.The measurement interval is UI number 9-28 at 20 GT/s and 25 GT/s. The measured voltage is averaged over this interval over more than 500 repetitions of the compliance pattern.

5.3.2.6 TX Timing Parameters

5.3.2.6.1 Jitter Parameters

The jitter parameters in EDR25-LR PHY must support the definition and methodologies in Section 8.3.5 in the *PCI Express Base Specification* (see Reference Documents).

Table 5-18 shows the jitter parameters in EDR25-LR PHY specification.

Table 5-18: The Jitter Specification at reference point T_{ball}

Characteristic	Symbol	Condition	20 GT/s	25 GT/s	Unit
			Max	Max	
TX Uncorrelated total jitter	$T_{TX\text{-}UTJ}$	At 1.0E-12	12.5	10.0	ps
TX Uncorrelated Dj for non-embedded Refclk	$T_{TX\text{-}UDJDD}$		5	4.0	ps
Total uncorrelated pulse width jitter	$T_{TX\text{-}UPW\text{-}TJ}$	At 1.0E-12	12.5	10.0	ps
Deterministic DjDD uncorrelated pulse width jitter	$T_{TX\text{-}UPW\text{-}DJDD}$		4	3.2	ps
TX Random Jitter	$T_{TX\text{-}RJ}$	Informative Parameter only	0.36 – 0.804	0.288 – 0.643	ps rms
Lane-to-Lane Output Skew	$L_{TX\text{-}SKEW}$		500ps + 8UI	500ps + 8UI	ps

5.3.2.6.2 De-Embedding the Breakout Channel for TX Jitter Measurement

When the TX jitter is measured at TP1, the replica channel can be used to de-embed the breakout channel.

The de-embedding methodology is defined in Section 8.3.5.2 of the *PCI Express Base Specification* (see Reference Documents), where 16.0GT/s de-embedding is specified.

5.3.2.6.3 Behavioral CDR Characteristics

A behavioral CDR of EDR25-LR PHY is the 1st order with 10MHz bandwidth. The transfer function of the 1st order CDR is defined in Section 8.3.5.5 in *PCI Express Base Specification* (see Reference Documents).

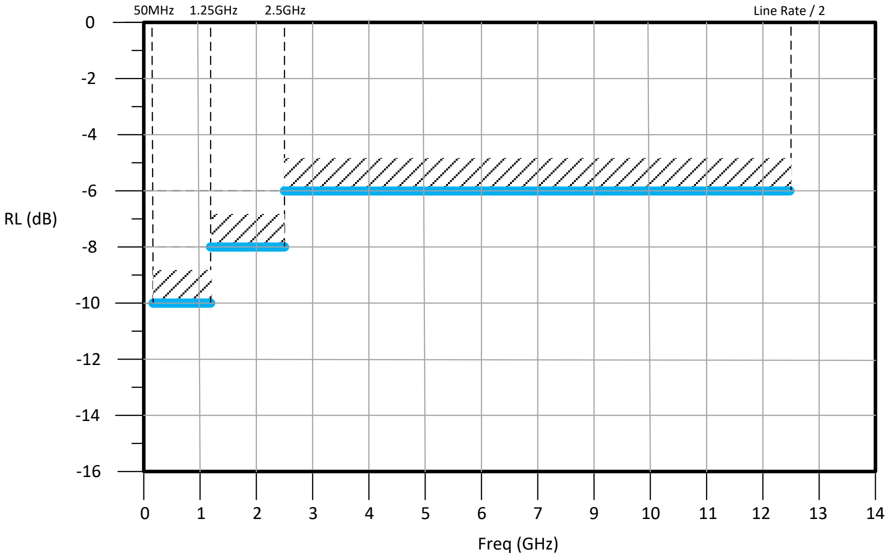
5.3.3 Receiver Specification

5.3.3.1 Receiver Termination

It specifies a nominal 100ohm differential load termination at DC at the receiver. The scheme for the termination needs to provide both differential and common mode termination to effectively absorb differential or common mode noise and reflections.

5.3.3.2 Differential Return Loss of Receiver

The receiver differential return loss at R_{ball} including package and socket is given in Figure 5-24.

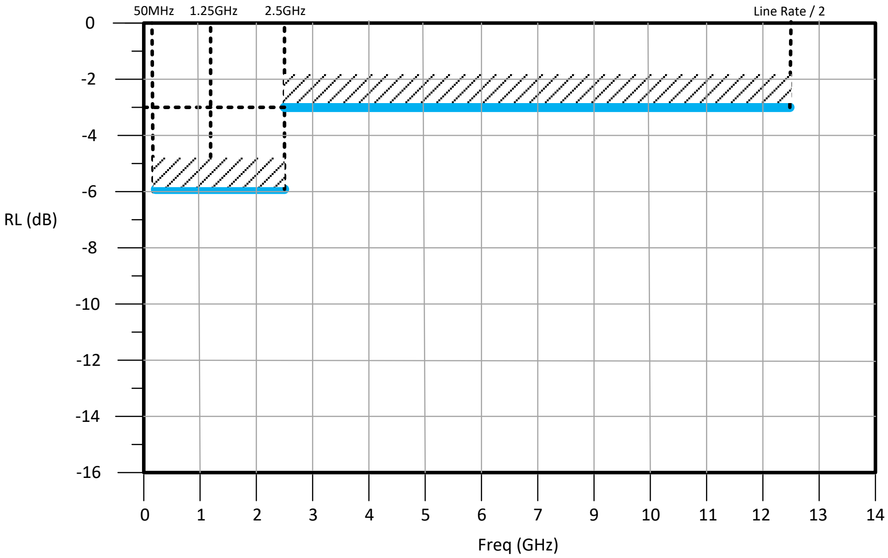


Note: The termination of the measurement is 100ohm

Figure 5-24: Receiver Differential Return Loss Mask for EDR25-LR PHY

5.3.3.3 Common Return Loss of Receiver

The common mode return loss for R_{ball} including package and socket is given in Figure 5-25. To attenuate noise and absorb even/odd mode reflections, receiver is required to satisfy the Common Mode Return Loss requirement.



Note: The termination of the measurement is 25ohm

Figure 5-25: Common Mode Return Loss Mask for EDR25-LR PHY

5.3.3.4 Common Receiver Parameters

The common receiver parameters are given in Table 5-19. EDR25-LR PHY must follow the test condition in Section 8.4.3 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-19: Common Receiver Parameters

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Rx PLL bandwidth corresponding to $PKG_{RX-PLL1}$	$BW_{RX-PKG-PLL1}$		2.0		4.0	MHz
Rx PLL bandwidth corresponding to $PKG_{RX-PLL2}$	$BW_{RX-PKG-PLL2}$		2.0		5.0	MHz
Maximum Rx PLL peaking corresponding to $BW_{RX-PKG-PLL1}$	$PKG_{RX-PLL1}$				2.0	dB
Maximum Rx PLL peaking corresponding to $BW_{RX-PKG-PLL2}$	$PKG_{RX-PLL2}$				1.0	dB
Rx termination float time	$RX_{GND-FLOAT}$				500	ns

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Rx AC common Mode Voltage	$V_{RX\text{-}CM\text{-}AC\text{-}P}$	Measured at RX pins into a pair of 50ohm terminations to GND			75 for EH <100mVPP 125 for EH >= 100mVpp	mVp
Electrical Idle Detect threshold	$V_{RX\text{-}IDLE\text{-}DET\text{-}DIFF\text{-}PP}$		65		175	mV
Unexpected Electrical Idle Enter Detect Threshold Integration Time	$T_{RX\text{-}IDLE\text{-}DET\text{-}DIFF\text{-}ENTERTIME}$				10	ms
Lane to Lane skew	$L_{RX\text{-}SKEW}$				4	ns

5.3.3.5 Jitter Tolerance

The jitter tolerance is given in the Table 5-20. The receiver shall tolerate the calibrated stress eye and sinusoidal jitter from the below table. The calibration of stressed eye must follow same methodology in Section 8.4.2 of the *PCI Express Base Specification* (see Reference Documents).

The jitter tolerance measurement in EDR25-LR PHY must follow the procedures in Section 8.4.2 of the *PCI Express Base Specification* (see Reference Documents).

Refclk in jitter tolerance measurement must include the spurs from SSC for the data-based CDR. It is typical to be applied triangular frequency modulation between 30 and 33kHz to common Refclk as defined in the *PCI Express Base Specification* (see Reference Documents).

Table 5-20: The Jitter Tolerance Specification

Characteristic	Symbol	Condition	Min.	Typ.	Max.	Unit
Sinusoidal Jitter, Maximum	R_sj_max		1			UIpp
Sinusoidal Jitter, High Frequency	R_sj_hf		0.1			UIpp

The jitter tolerance mask is given in Figure 5-26.

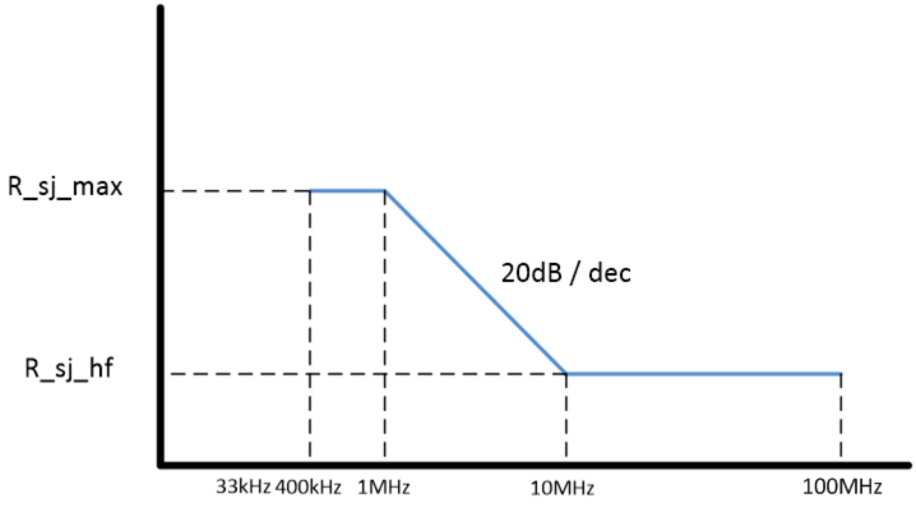


Figure 5-26: Jitter Tolerance Mask of EDR25-LR PHY

5.3.3.6 Receiver Stressed Eye Specification

All ESM Data rates of EDR25-LR PHY are required to be tested by a stressed eye including calibration channel which closely approximates the worst-case loss characteristic in the real CCIX-LR application. The recovered eye is defined at TP2P which is at the input of the receiver's analog latch.

5.3.3.6.1 Breakout and Replica Channels

Replica channel which is closely match to breakout channel of DUT board makes the measurement of the signal at RX DUT's pin possible. The impedance target of replica channel is required to match with the impedance of breakout channel which can be either 100ohm differential or 85ohm with +/-5% or better.

Figure 5-27 shows the RX Test board topology for EDR25-LR PHY. The stressed eye needs to be measured at TP2 point with external signal source. After the stressed eye has been calibrated, this signal is applied to DUT over the breakout channel.

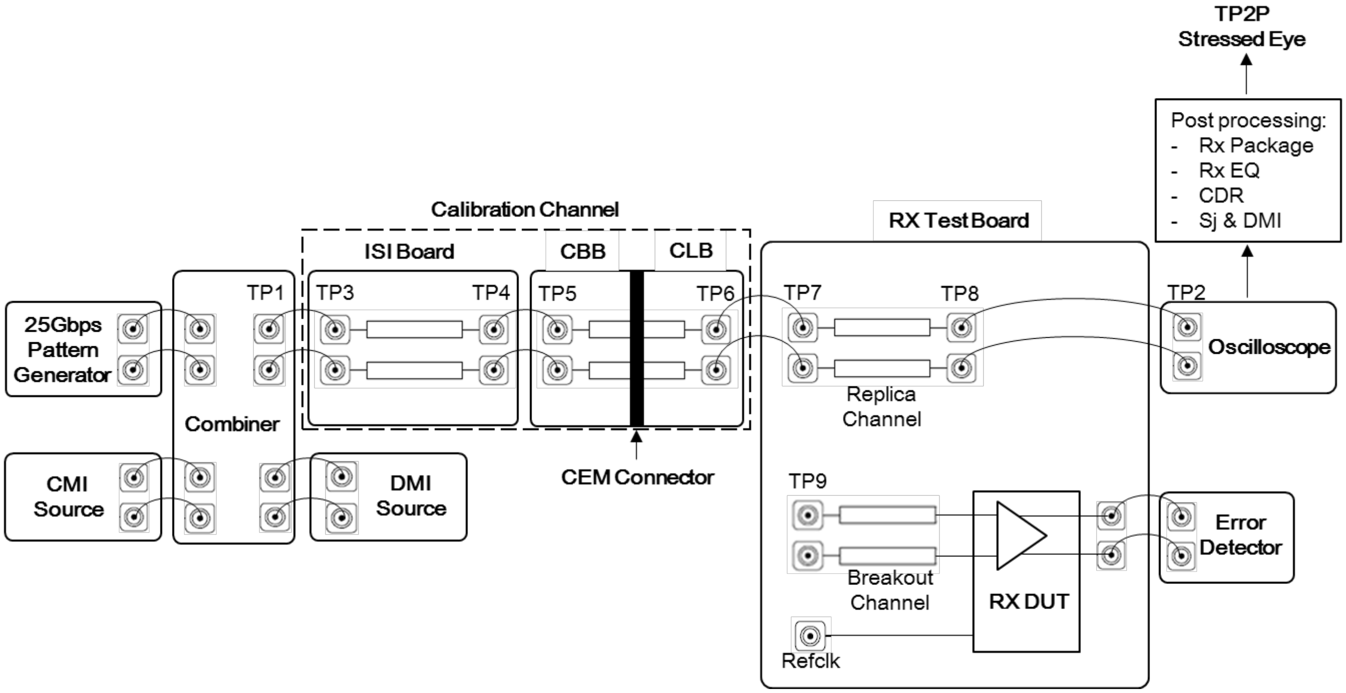


Figure 5-27: RX Test Board Topology for EDR25-LR

Note1: Reference boards shall use 2.92mm connector to connect each test points.

Note2: All cables and discrete broadband termination are 50ohm.

5.3.3.6.2 Calibration Channel Insertion Loss Characteristics

The calibration channel loss specification reuses the definition in Section 8.4.1.2 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-21 shows the loss range of the initial calibration channel for EDR25-LR PHY.

Table 5-21: Calibration Channel IL Limits for CCIX-LR

ESM Data Rate	$F_{LOW\text{-}IL\text{-}MIN}$	$F_{LOW\text{-}IL\text{-}MAX}$	$F_{HIGH\text{-}IL\text{-}MIN}$	$F_{HIGH\text{-}IL\text{-}MAX}$
20.0 GT/s Root Port	3.2 dB @ 1 GHz	4.2 dB @ 1 GHz	22.5 dB @ 10 GHz	23.5 dB @ 10 GHz
20.0 GT/s Non-Root Port	3.7 dB @ 1 GHz	4.7 dB @ 1 GHz	28.0 dB @ 10 GHz	29.0 dB @ 10 GHz
25.0 GT/s Root Port	3.2 dB @ 1 GHz	4.2 dB @ 1 GHz	22.5 dB @ 12.5 GHz	23.5 dB @ 12.5 GHz
25.0 GT/s Non-Root Port	3.7 dB @ 1 GHz	4.7 dB @ 1 GHz	28.0dB @ 12.5 GHz	29.0 dB @ 12.5 GHz

Note: Calibration channel plus RX reference package is 32dB nominally at Nyquist frequency which is informative.

The impedance target of RX Test Board topology except for Replica channel is 85ohm differential and 42.5ohm single-ended with +/-5% or better.

5.3.3.6.3 Post Processing Procedures

Section 8.4.1.3 of the *PCI Express Base Specification* (see Reference Documents), is incorporated by reference.

5.3.3.6.4 Behavioral Rx Package Models

Separate package loss models are required to be used to represent root complex device and end point device in Figure 5-27. The effective package loss reuses the definition in Section 8.3.3.11 of the *PCI Express Base Specification* (see Reference Documents).

If the actual Rx package performance is worse than that of the behavioral package, then the actual package models are permitted to be used. If the actual package models are used, the calibration channel must be adjusted such that the total Rx test channel loss described in Table 5-16, including the embedded actual package, must be same with the loss including the behavioral package model. Note that form factor overall requirements still need to be met. The Rx package performance is assessed using the methodology in Section 8.5.1.2 of the *PCI Express Base Specification* (see Reference Documents).

5.3.3.6.5 Behavioral CDR Model

Post processing should include a behavioral CDR model. EDR25-LR PHY has a first order CDR transfer function with 10MHz loop bandwidth, as described in Section 8.4.1.5 of the *PCI Express Base Specification* (see Reference Documents).

5.3.3.6.6 Behavioral CTLE

EDR25-LR PHY has a first order CTLE with variable first pole, positive peaking gain and adjustable DC gain. For details on CTLE, refer to Section 5.3.6.1.2.

5.3.3.6.7 Behavioral DFE

EDR PHY for CCIX-LR defines 8 taps DFE with the combination of a first order CTLE. For details on DFE tap size, refer to Table 5-25.

5.3.3.7 Calibration Stress Eye for Jitter Tolerance

The stress eye for jitter tolerance test needs to be calibrated to present the receiver with the worst case situation. EDR25-LR PHY must follow same methodology to calibrate the stressed eye for jitter tolerance test in Section 8.4.2 of the *PCI Express Base Specification* (see Reference Documents).

The reference receiver for the calibration procedures is defined in Section 5.3.6.

$$Calibrated\ EH\ /\ EW\ =\ 30\mathrm{mV}\ /\ 0.3\mathrm{UI}$$

Table 5-22: Calibration Parameters for Jitter Tolerance Test

Characteristic	Symbol	Condition	20 GT/s	25 GT/s	Unit
Eye Width at TP2P	$T_{RX\text{-}ST}$		0.3	0.3	UI
Eye Height at TP2P	$V_{RX\text{-}ST}$		30	30	mV
Random Jitter	$T_{RX\text{-}ST\text{-}RJ}$	*Notes: 1, 4*	0.75 (max)	0.6 (max)	ps_{rms}
Differential Noise	$V_{RX\text{-}DIFF\text{-}INT}$	Adjust to control EH Frequency = 2.1GHz *Notes: 2, 3, 4*	14	14	mVpp
Common mode Noise	$V_{RX\text{-}COMM\text{-}INT}$	Defined as a single tone at 120MHz. Measured at TP2 without post processing	150	150	mVpp
Minimum RX calibration channel loss	$ISI_{RX\text{-}ISI\text{-}LOSS\text{-}min}$	The minimum differential loss for RX calibration channel from TP1 to TP2P	26	26	dB
Maximum RX calibration channel loss range	$ISI_{RX\text{-}ISI\text{-}LOSS\text{-}max}$	The maximum differential loss for RX calibration channel from TP1 to TP2P	34	34	dB

Note1: Rj is measured at TP1. The low frequency limit of Rj may be between 1.5MHz and 10MHz, and the upper limit is 1.0GHz.

Note2: Both $T_{RX\text{-}ST\text{-}RJ}$ and $V_{RX\text{-}DIFF\text{-}INT}$ are limited to prevent the stressed eye from excessive amount of jitter or noise distortion.

Note3: The frequency of $V_{RX\text{-}DIFF\text{-}INT}$ is chosen to be slightly higher than the lowest first pole of the reference CTLE.

Note4: Common mode noise is turned off during $T_{RX\text{-}ST\text{-}rj}$ and $V_{RX\text{-}DIFF\text{-}INT}$ calibration and then turned on for the stressed eye jitter test.

5.3.3.7.1 Procedure to Calibrate Stressed Eye

Same as that described in Section 8.4.2.1 in *PCI Express Base Specification* (see Reference Documents), the goal of calibrating a stressed voltage/jitter eye is to present the receiver under test with simultaneously worst case margins whose distortion characteristics are similar to an eye produced by a real channel. Much of the distortion consists of the ISI produced by the calibration channel. Incremental changes of differential voltage from nominal values may be used to adjust the EH while the EW is post processed to evaluate the worst case of Sj that will be used onto RX DUT test, which is different from the *PCI Express Base Specification*.

The reference point where EH/EW is defined corresponds to input to the receiver latch at 20.0 GT/s and 25.0 GT/s. Since this point is not physically accessible it is necessary to construct its equivalent by means of a post-processing procedure. A step response that has been averaged 1024 times at TP2 is first post processed in order to mathematically include the additional signal distortion caused by the behavioral receiver package. As the averaged step response is used, the stresses of differential voltages are turned on, but the stresses of jitters will not be included and are turned off. Then the resulting signal is recovered by means of Rx equalization, and a behavioral CDR function, resulting in an equivalent eye. The requirements for the waveform post processing tool used for the EH/EW calibration are described further in Section 5.3.3.7.3.

As the calibration procedure of the signal generator output contains steps where the generator is connected directly to measurement instrumentation, the transition time of the output waveform can be very fast. Therefore, it is important that the bandwidth of instrumentation used to calibrate the generator be matched appropriately to the edge rate of the generator output. This specification requires the use of a generator whose outputs have less than 19ps rise time (20% / 80%) which also requires a minimum oscilloscope bandwidth of 32GHz. This oscilloscope bandwidth is also the minimum required bandwidth for transmitter measurements. The oscilloscope must have enough low noise floor up to the bandwidth of it and the measured step response needs to show reasonable response in the frequency domain.

The loss of combiner at 12.5GHz including cables from the output of generator to TP1 should be less than 2dB. The 3dB cutoff frequency of combiner is required to be at least 20GHz.

Note that for the eye calibration process, the Tx equalization is fixed to the preset that gives the optimal eye area with the post processing tool being used for calibration. Once the testing procedure is under way the Tx preset may be adjusted to yield the best eye margins with the DUT. For calibration at ESM rates the following process is used to calibrate the eye:

1. *Calibrate the stress values as shown below at TP1:*
    a. *Tx Differential Voltage Swing to 800 mV PP (+ 10 mV / - 0 mV).*
    b. *TX presets that specified in Table 8-1 in PCI Express Base Specification 4.0 within +/- 0.1 dB.*
    c. *Sj by 0.1 UI (4 ps) at 100 MHz. As the step response is used, this Sj is included to post-processing tool.*
    d. *Rj to 0.6 ps rms. As the step response is used, this Rj is included to post-processing tool.*
    e. *DMI to 14mV at TP2P. As the step response is used, this DMI is included to post-processing tool.*
2. *Connect an initial test channel with selecting the ISI length that give a loss of TP1 to TP2P*
    a. *The loss of an initial test channel for 20GT/s: 26dB +/-0.5dB at 10GHz*
    b. *The loss of an initial test channel for 25GT/s: 26dB +/-0.5dB at 12.5GHz*
3. *Measure the eye diagram for each TX EQ preset using the calibrated TX EQ and select the TX EQ preset that gives the largest eye area.*

IMPLEMENTATION NOTE: REQUIREMENT OF COMBINER

To combine the signal from pattern generator and DMI noise source either RF combiner or directional coupler can be used. Figure 5-28 is the recommended frequency mask for the combiner block.

“Nominal loss” means DC or low frequency loss at less than couple hundreds kHz. Usually RF combiner has 6dB and directional coupler has 0dB.

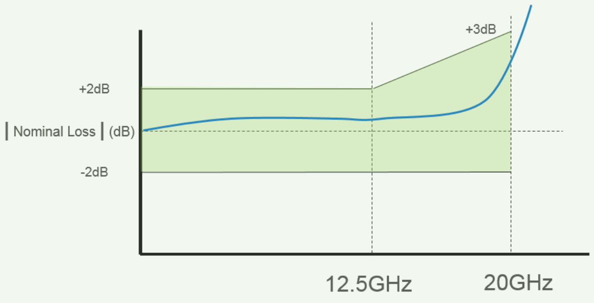


Figure 5-28: The frequency mask of combiner block

For all EH, EW and eye area measurements performed in receiver calibration the DC Gain in the reference receiver CTLE is varied over its minimum to maximum range in 0.25 dB steps. This is done to improve repeatability and accuracy in automated Rx calibration software and is only done for stressed eye calibration (not for channel compliance, etc.)

4. *Increase the calibration channel loss to the next available length/loss and measure the new eye diagram at the selected preset. Continue to increase the length/loss until either the height or width have fallen below the targets (EH – 30 mV +/- 3 mV , EW – 0.3 UI +/- 0.0125UI) then the previous calibration channel length/loss is selected. If neither the height or width have fallen below the targets and the TP1 to TP2P loss at 10 GHz for 20 GT/s and 12.5 GHz for 25 GT/s has reached 34.0 dB then advance to the next step.*
5. *For the selected calibration channel length/loss, measure the eye diagram for each TX EQ preset and select the preset that gives the largest eye area. Note that this may be a different preset than step 3 due to the length/loss change. Also the noises in differential and common modes are calibrated as following:*
6. *For the fixed calibration channel length/loss and Tx EQ in step 5, adjust DMI_{TP2P} and Voltage Swing to make final adjustments to the eye by sweeping them through the following ranges:*
    a. *Sj in post-processing tool – 4 to 8ps*
    b. *DMI_{TP2P} in post-processing tool – Equivalent range with (10 mV to 25 mV) DMI at TP2*
    c. *Differential Voltage Swing at generator – 720 mV to 800 mV at TP1*

    The range of DMI_{TP2P} in post-processing tool is calculated by DMI range at TP2 (10mV ~ 25mV) and the gain of adapted CTLE. The following table can be used to map DMI and DMI_{TP2P}

$$DMI_{TP2P} = DMI * 10^{((Gain\_dB + Rxpkg\_dB) / 20)}$$

Table 5-23: The gain of adapted CTLE at 2.1GHz (dB)

Adc	fp1 = 2 GHz	fp1 = 3 GHz	fp1 = 4 GHz	fp1 = 5 GHz	fp1 = 6 GHz	fp1 = 7 GHz
-12 dB	3.227877015	1.277809476	-0.441769238	-1.897447015	-3.122641603	-4.157936392
-11 dB	3.243783934	1.312899404	-0.381050909	-1.805719296	-2.995695202	-3.992745244
-10 dB	3.263727069	1.356675649	-0.305798676	-1.692929302	-2.840981112	-3.793336228
-9 dB	3.288704467	1.411166568	-0.212878946	-1.554977483	-2.653731465	-3.554652053
-8 dB	3.319946255	1.478808493	-0.098657761	-1.387314200	-2.428927746	-3.271695704
-7 dB	3.358960443	1.562493428	0.040994128	-1.185045318	-2.161520706	-2.939839475
-6 dB	3.407583237	1.665604194	0.210651390	-0.943108544	-1.846719804	-2.555170287
-5 dB	3.468031678	1.792026076	0.415224431	-0.656523732	-1.480330533	-2.114825712
-4 dB	3.542954593	1.946121219	0.659778460	-0.320705984	-1.059099557	-1.617265183
-3 dB	3.635475450	2.132650548	0.949278798	0.068187943	-0.581014204	-1.062424810
-2 dB	3.749217823	2.356629378	1.288274679	0.512926595	-0.045501588	-0.451723930
-1 dB	3.888301081	2.623109029	1.680552158	1.014939940	0.546512788	0.212077983
0 dB	4.057291596	2.936888665	2.128803702	1.574175246	1.192699283	0.925144475

7. *If the final calibrated Sj is less than 0.1 UI then the Rj level in post processing tool is reduced so the eye width meets the target eye width with 0.1 UI of Sj in post processing tool*

8. *If there are multiple combinations of DMI and Voltage Swing that give valid solutions first pick the combination that is closest to the target eye width (0.3 UI). The selected values must give a mean eye height and width (over at least 5 measurements – exact number of 15 measurements needed for stable values will depend on lab set-up and tools) within the following ranges at BER 1E-12:*
    a. *Eye height – 30 mV +/- 3 mV*
    b. *Eye width – 0.3 UI +/- 0.0125 UI*

Figure 5-29 shows a block diagram for calibrating the stressed eye for EDR25-LR PHY. If the calibration set up with the specific pattern generator and the oscilloscope cannot achieve calibration at the specific loss range, this set up is not valid and cannot be used for RX calibration procedures. Table 5-28 shows the loss range of RX calibration for EDR25-LR PHY.

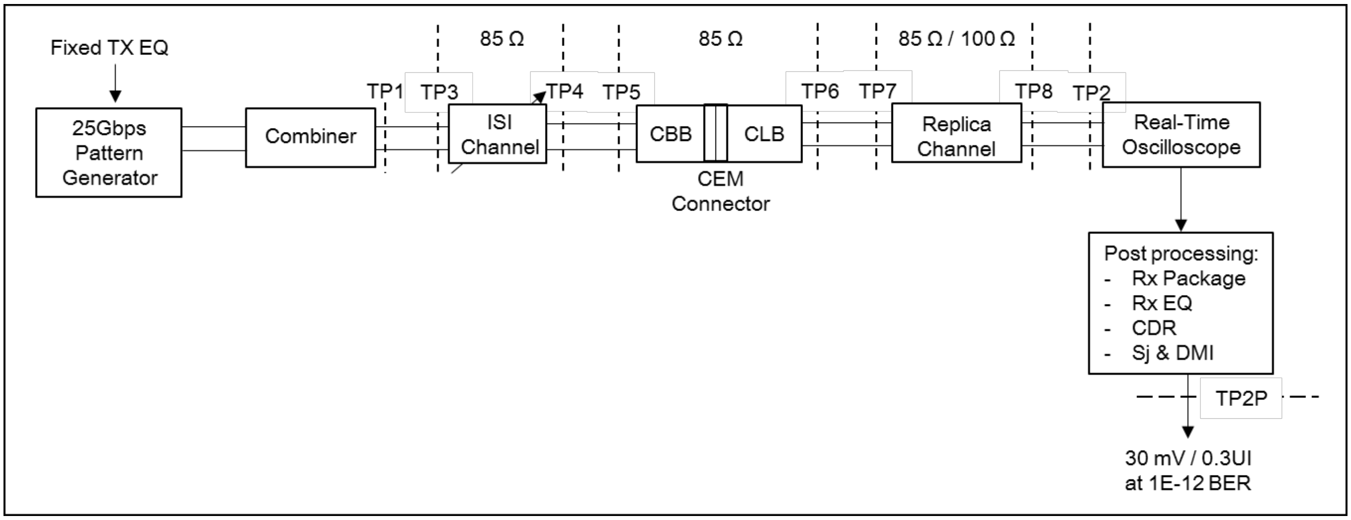


Figure 5-29: Layout for Calibrating the Stressed Jitter Eye at ESM Rates

5.3.3.7.2 Procedure for Testing Rx DUT

Once a calibrated EH and EW have been obtained, the cable connected at TP7 in Figure 5-29 is moved to TP9 so that the Rx DUT is connected to the far end of calibration channel as shown in Figure 5-30. The cable from TP8 to TP2 during the calibration is not included to the Rx DUT test. So, the cable from TP8 to TP2 should be picked carefully to have enough low loss at 10 GHz for 20 GT/s and 12.5 GHz for 25 GT/s to be ignored or deembedded during the calibration process.

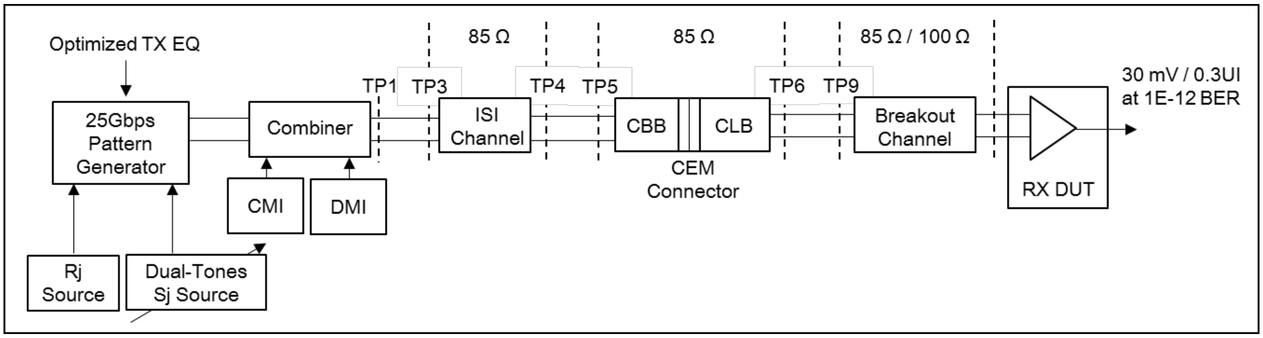


Figure 5-30: Layout for Testing Rx DUT

The Tx equalization may then be optimized with the assumption that the DUT Rx will also optimize its equalization. Sj is set to an initial value of 0.1 UI at 100 MHz and the receiver CDR must achieve lock. The 100 MHz Sj initial tone is removed and then the appropriate swept Sj profile is tested. Also an additional Sj tone at 210 MHz is present for all testing. The amplitude of this additional tone is equal to the amplitude of the 100 MHz Sj required to achieve the target eye width minus 0.1 UI. If the calibration Sj level was less than 0.1 UI then no additional tone at 210 MHz is used. Receivers operating the CC Refclk mode shall utilize the Sj profile shown in Figure 8-36 in *PCI Express Base Specification* (see Reference Documents).

5.3.3.7.3 Waveform Post Processing Tool Requirements

The waveform post processing tool to calibrate EH/EW for RX stressed eye test must be consistent with the channel compliance methodology in Section 8.5.1 of the *PCI Express Base Specification* (see Reference Documents).

This tool needs to be used during RX stressed eye calibration and there are few notes for using the tool as follows:

- ✓ *The reference package model of Rx need to be embedded by either the oscilloscope or the post processing tool.*
- ✓ *All step measurements are done at TP2.*
- ✓ *A step pattern with 512 ones and zeroes is captured through the test channel by averaging 1024 times on a real time oscilloscope. The step is saved with an x-axis resolution of 1 ps or less to be used as the transmit waveform for the channel compliance methodology. The step pattern is captured with the calibrated presets of TX EQ enabled at Generator. When the measured step is converted to the frequency domain by using appropriate mathematical function, it is required to show reasonable frequency domain response*
- ✓ *A captured step is simulated in the post processing tool with TX EQ of the tool itself disabled. The equivalent jitters and DMI are set in the tool at the same time.*
- ✓ *For the preset that gives the largest eye area with the waveform post processing tool the EH and EW @ 1E-12 BER must match 30 mV +/- 10% and 0.3 UI +/- 0.0125 UI.*

A general flow for using post processing tool is shown in Figure 5-31.

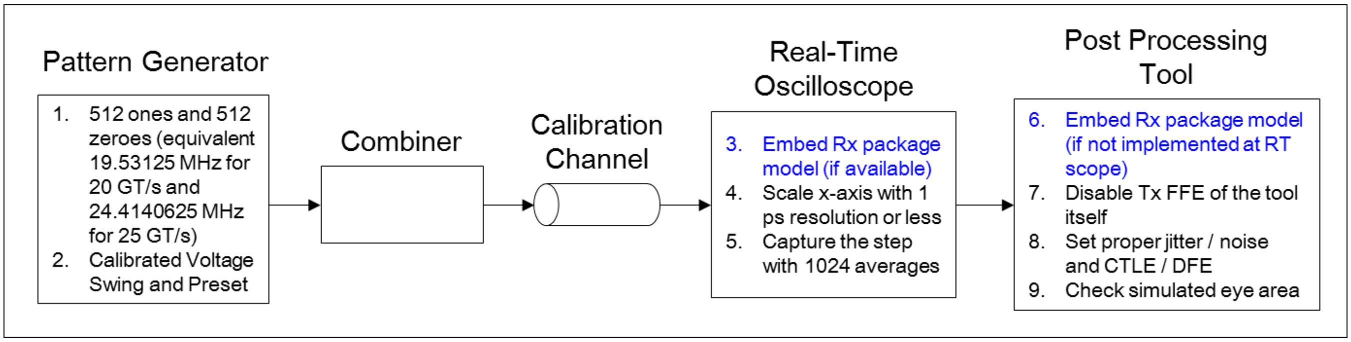

Pattern Generator
1. 512 ones and 512 zeroes (equivalent 19.53125 MHz for 20 GT/s and 24.4140625 MHz for 25 GT/s)
2. Calibrated Voltage Swing and Preset
Combiner
Calibration Channel
Real-Time Oscilloscope
3. Embed Rx package model (if available)
4. Scale x-axis with 1 ps resolution or less
5. Capture the step with 1024 averages
Post Processing Tool
6. Embed Rx package model (if not implemented at RT scope)
7. Disable Tx FFE of the tool itself
8. Set proper jitter / noise and CTLE / DFE
9. Check simulated eye area

Figure 5-31: General Flow for Using Post Processing Tool

5.3.3.8 Receiver Refclk Modes

CCIX supports Common Clock (CC) mode only. Figure 5-32 shows the Refclk connection for a receiver. A single Refclk source drives both the external signal source and DUT.

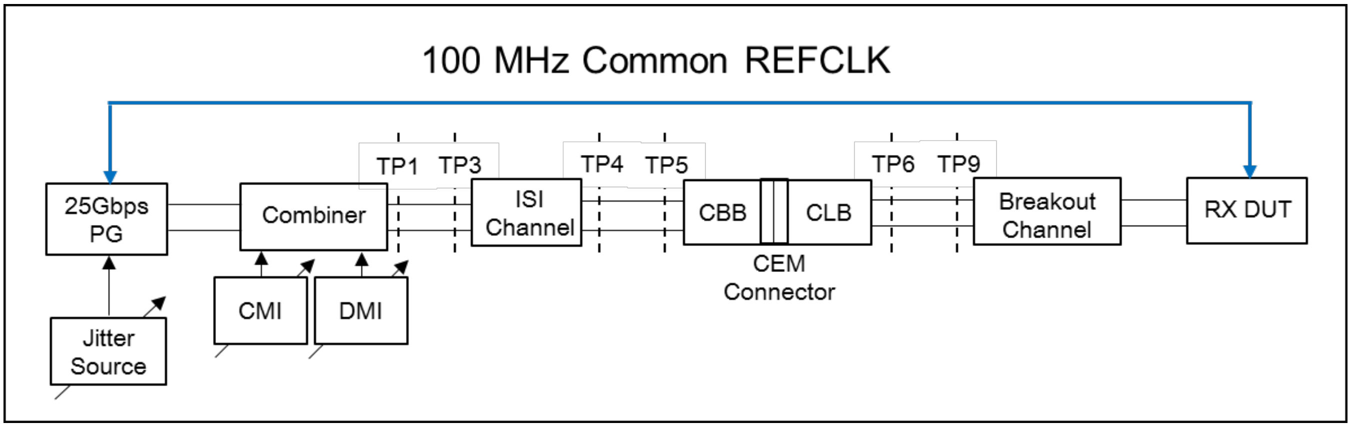


Figure 5-32: Layout for Jitter Testing for Common Refclk Rx

5.3.4 PCIe-Specific Specifications

5.3.4.1 Electrical Idle

Section 8.4.5.5 of the *PCI Express Base Specification* (see Reference Documents) is incorporated by reference.

5.3.4.2 Receiver Detection

Section 8.4.5.7 of the *PCI Express Base Specification* (see Reference Documents) is incorporated by reference.

5.3.4.3 Receiver Margining

Section 8.4.4 of the *PCI Express Base Specification* (see Reference Documents) is incorporated by reference.

5.3.4.4 Link Training for TX EQ with back channel

Section 4.2.3 of the *PCI Express Base Specification* (see Reference Documents), is incorporated by reference.

5.3.5 Reference Clock Specification

The reference clock specification in EDR25-LR specification reuses the requirements for common refclk (CC) in Section 8.6 of the *PCI Express Base Specification* (see Reference Documents).

5.3.5.1 Reference Clock Electrical Specification

Table 5-24 shows the Reference Clock Electrical Specification.

Table 5-24: Refclk specification

Symbol	Description	Min.	Max.	Unit
F_{refclk}	Reference Clock frequency	99.97	100.03	MHz
F_{SSC}	SSC Frequency Range	30	33	kHz
$T_{SSC\text{-}FREQ\text{-}DEVIATION}$	SSC Deviation	-0.5	0	%
T_{REFCLK}	Residual Jitter Limit for CC using 2 port without system noise		0.35	ps $_{rms}$
T_{REFCLK_noise}	Residual Jitter Limit which can be used channel simulations to account for additional noise in a real system		0.5	ps $_{rms}$
$T_{TRANSPORT\_DELAY}$	TX-RX Transport delay		12	ns

5.3.5.2 Reference Clock Compliance Measurement

EDR25-LR PHY must use same methodology to calculate the residual jitter of Refclk for the compliance in Section 8.6 of the *PCI Express Base Specification* (see Reference Documents).

The jitter measurement of Refclk shall be made with a capture at least 100,000 clock cycles by a real-time oscilloscope with a sample rate of 20GS/s or greater. Broadband oscilloscope noise must be minimized in the measurement. The PLL and CDR filters in Table 5-10 should be applied to time domain raw data measurement.

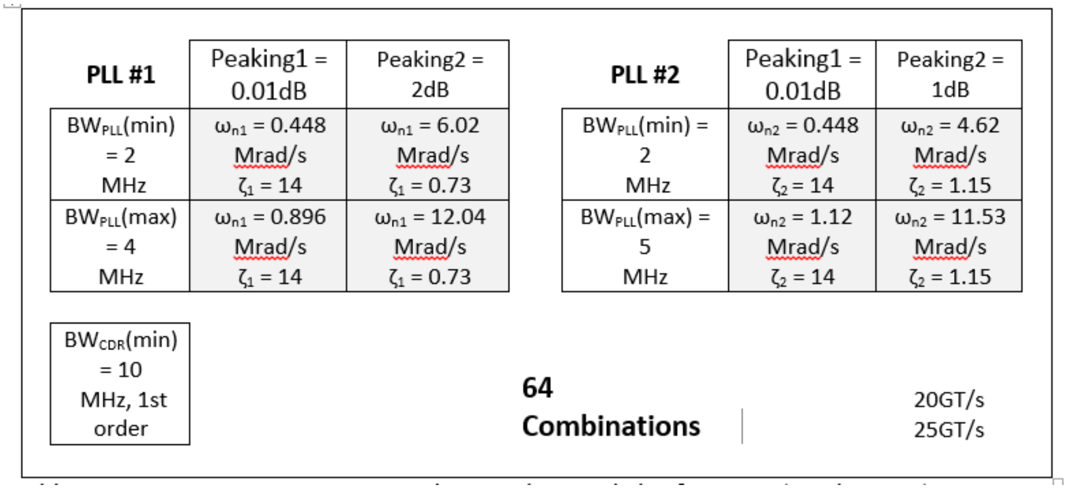
SSC spurs from the fundamental and harmonics are removed up to a cutoff frequency of 2MHz taking care to minimize removal of any non-SSC content.

5.3.5.3 CDR and PLL Bandwidth and Peaking for Reference Clock Compliance

CDR and PLL BW and Peaking for reference clock measurement need to be defined and EDR25-LR PHY reuses the methodology in Sections 8.6.5.1 and 8.6.5.2 of the *PCI Express Base Specification* (see Reference Documents).

Table 5-25 shows CDR and PLL BW and peaking of reference TX PLL, RX PLL and CDR for EDR link.

Table 5-25: Common Refclk PLL and CDR Characteristics for 20 GT/s and 25 GT/s



PLL #1	Peaking1 = 0.01dB	Peaking2 = 2dB
$BW_{PLL}(min)$ = 2 MHz	ω_{n1} = 0.448 Mrad/s ζ_1 = 14	ω_{n1} = 6.02 Mrad/s ζ_1 = 0.73
$BW_{PLL}(max)$ = 4 MHz	ω_{n1} = 0.896 Mrad/s ζ_1 = 14	ω_{n1} = 12.04 Mrad/s ζ_1 = 0.73

PLL #2	Peaking1 = 0.01dB	Peaking2 = 1dB
$BW_{PLL}(min)$ = 2 MHz	ω_{n2} = 0.448 Mrad/s ζ_2 = 14	ω_{n2} = 4.62 Mrad/s ζ_2 = 1.15
$BW_{PLL}(max)$ = 5 MHz	ω_{n2} = 1.12 Mrad/s ζ_2 = 14	ω_{n2} = 11.53 Mrad/s ζ_2 = 1.15

$BW_{CDR}(min)$ = 10 MHz, 1st order

64 Combinations

20GT/s
25GT/s

5.3.6 Channel Compliance

5.3.6.1 Channel Compliance Methodology

5.3.6.1.1 Using Internal EW/EH with Reference Receiver

EDR25-LR PHY must follow the same methodology to check the channel compliance in Section 8.5.1 of the *PCI Express Base Specification* (see Reference Documents).

Figure 5-33 shows the key components and processes of channel tolerancing at ESM rates. This flow must be implemented in channel tolerancing tools.

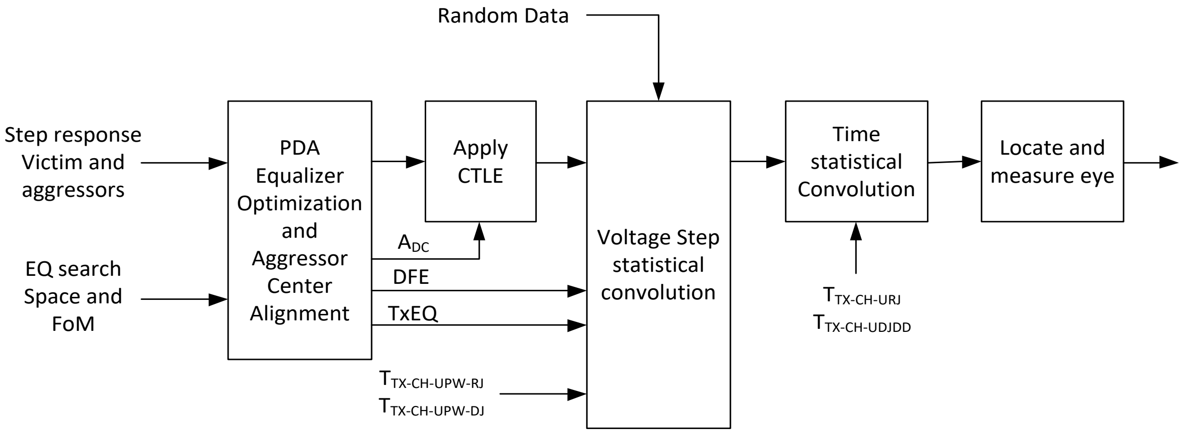

Random Data
Step response Victim and aggressors
EQ search Space and FoM
PDA Equalizer Optimization and Aggressor Center Alignment
Apply CTLE
A_{DC}
DFE
TxEQ
$T_{TX\text{-}CH\text{-}UPW\text{-}RJ}$
$T_{TX\text{-}CH\text{-}UPW\text{-}DJ}$
Voltage Step statistical convolution
Time statistical Convolution
$T_{TX\text{-}CH\text{-}URJ}$
$T_{TX\text{-}CH\text{-}UDJDD}$
Locate and measure eye

Figure 5-33: Flow Diagram for Channel Tolerancing at ESM Rates

Table 5-26 shows reference receiver parameters for EDR25-LR channel compliance. $V_{RX\text{-}CH\text{-}EH}$ is eye height for the nominal channel.

Table 5-26: Reference Receiver Parameters for Channel Compliance

Symbol	Description	Value	Unit	Note
$V_{RX\text{-}CH\text{-}EH}$	Eye height with nominal channel	30 (min)	mVpp	Eye height at BER=10-12 Note1
$T_{RX\text{-}CH\text{-}EW}$	Eye width at zero crossing	0.3 (min)	UI	Eye width at BER=10-12
R_{H1_H0}	H1 / H0 Ratio	0.7 (max)		The ratio between data cursor and the 1st post cursor
$T_{RX\text{-}DS\text{-}OFFSET}$	Peak EH offset from UI center	+/- 0.1	UI	
$V_{RX\text{-}DFE\text{-}D1}$	Range for DFE d1 coefficient	+/-30	mV	
$V_{RX\text{-}DFE\text{-}D2}$	Range for DFE d2 coefficient	+/-20	mV	
$V_{RX\text{-}DFE\text{-}D3}$	Range for DFE d3 coefficient	+/-20	mV	
$V_{RX\text{-}DFE\text{-}D4}$	Range for DFE d4 coefficient	+/-20	mV	
$V_{RX\text{-}DFE\text{-}D5}$	Range for DFE d5 coefficient	+/-20	mV	
$V_{RX\text{-}DFE\text{-}D6}$	Range for DFE d6 coefficient	+/-20	mV	
$V_{RX\text{-}DFE\text{-}D7}$	Range for DFE d7 coefficient	+/-20	mV	
$V_{RX\text{-}DFE\text{-}D8}$	Range for DFE d8 coefficient	+/-20	mV	

Note1: The variation of channel must be considered in the simulation for channel compliance

5.3.6.1.2 Behavioral CTLE

CCIX behavioral Rx equalization defines a 1st order CTLE with fixed LF and HF poles, and an adjustable DC gain (A_{DC}) specified to the family of curves shown in Figure 5-34. A_{DC} is adjustable over a minimum range f 0 to -12dB in steps of 1.0dB. Transfer function of CTLE is shown in Equation 4.

$$\mathbf{H(s)} = \boldsymbol{g_{dc}} \frac{(\frac{\boldsymbol{s}}{\boldsymbol{\omega_Z}} + \mathbf{1})}{\left(\frac{\boldsymbol{s}}{\boldsymbol{\omega_{P1}}} + \mathbf{1}\right) * (\frac{\boldsymbol{s}}{\boldsymbol{\omega_{P2}}} + \mathbf{1})}$$

$$\boldsymbol{\omega_{P1} = pole1 = 2\pi * f_{P1}}$$

$$\boldsymbol{(f_{P1} = pole1 = from\ 2GHz\ to\ 7GHz\ in\ steps\ of\ 1GHz)}$$

$$\boldsymbol{\omega_{P2} = pole2 = 2\pi * 25\ GHz}$$

$$\boldsymbol{\omega_Z = Zero\ Frequency = \omega_{P1} * (\frac{g_{dc}}{g_{ac}})}$$

$$\boldsymbol{A_{DC} = from - 12\ to\ 0\ dB\ in\ steps\ of\ 1dB}$$

$$\boldsymbol{A_{AC} = 6\ dB}$$

$$\boldsymbol{g_{dc} = \ 10^{A_{DC}/20}}$$

$$\boldsymbol{g_{ac} = \ 10^{A_{AC}/20}}$$

Equation 4: Transfer Function for CCIX Behavioral CTLE

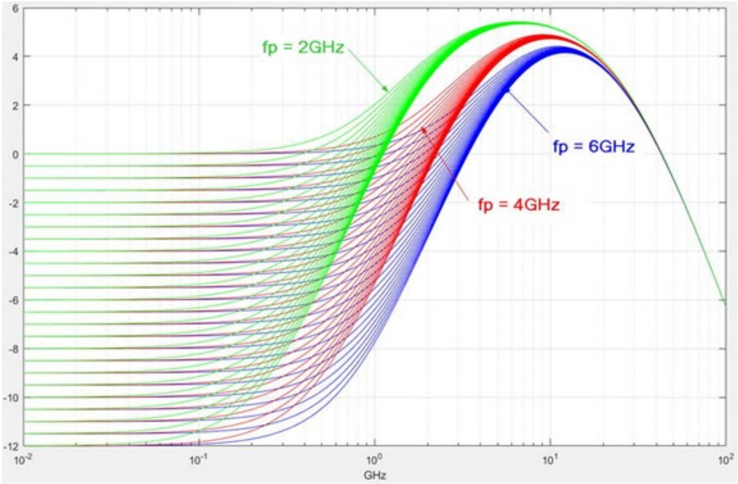


Figure 5-34: Loss Curves for CCIX Behavioral CTLE

5.3.6.1.3 Behavioral Transmitter and Receiver Package Models For LR (Informative)

The behavioral package models are defined to represent the combined die and package loss. Table 5-27 has the parasitic capacitances at die pin and BGA ball.

Table 5-27: Package Model Capacitances

Symbol	Description	Value	Unit	Note
C_{pin}	BGA Ball	0.25	pF	
C_{pad}	Die-pad	0.25	pF	

The package model which is used during LR channel compliance procedures must include the parasitic capacitances in Table 5-23.

5.3.6.1.4 Behavior Transmitter Parameters

Table 5-28 shows jitter / voltage parameters of transmitter for channel compliance.

Table 5-28: Jitter/Voltage Parameter of behavioral transmitter

Symbol	Description	20 GT/s	25 GT/s	Unit	Note
$T_{TX\text{-}CH\text{-}URJ}$	TX uncorrelated RJ	0.494	0.395	ps_{rms}	
$T_{TX\text{-}CH\text{-}UDJDD}$	TX uncorrelated DjDD	3	2.4	ps_{pp}	
$T_{TX\text{-}CH\text{-}UPW\text{-}RJ}$	Uncorrelated PW RJ	0.607	0.486	ps_{rms}	
$T_{TX\text{-}CH\text{-}UPW\text{-}DJ}$	PW DDJ	4	3.2	ps_{pp}	Include parasitic die pad capacitance

5.3.6.2 Frequency Domain Response for Channel Compliance (Informative)

The frequency domain characteristics of channel in this section are informative specifications.

The frequency domain responses for EDR25-LR PHY channel reuses the channel compliance of CEI-25G-LR for frequency domain specification of channel as an informative specification. The channel characteristics are defined in Table 5-29.

Table 5-29: Channel Characteristic

	Symbol	Description
Measured Channel Parameters	IL(f)	Differential insertion loss, SDD21 (dB)
	RL1(f)	Differential input return loss, SDD11 (dB)
	RL2(f)	Differential output return loss, SDD22 (dB)
	NEXTm(f)	Differential near-end crosstalk loss (m-th aggressor) (dB)
	FEXTn(f)	Differential far-end crosstalk loss (n-th aggressor) (dB)
Calculated Channel Parameters	$IL_{fitted}(f)$	Fitted insertion loss (dB)
	ILD(f)	Insertion loss deviation (dB)
	ICN(f)	Integrated crosstalk noise (mV, RMS)
	ILD(rms)	RMS value of the insertion loss deviation (dB)

5.3.6.2.1 Insertion Loss between reference point T_{die} and R_{die} (Informative)

EDR High Speed has informative insertion loss definition between die pads. The insertion loss between reference point T_{die} and R_{die} is up to -32dB at Nyquist frequency.

Table 5-30 shows the detail break-down of insertion loss between die pads is shown in below sub-sections.

Table 5-30: Insertion Loss Break-Down

Symbol	Description	Min	Max	Unit
$IL_{link\_die\_die}$	Differential Insertion loss of total link from T_{die} to R_{die} at Nyquist frequency		-32	dB
IL_{tx_pkg}	Differential Insertion Loss of root complex TX Package from Tdie to Tball at Nyquist frequency		-9	dB
IL_{rx_pkg}	Differential Insertion Loss of non-root complex device RX Package from R_{die} to R_{ball} at Nyquist frequency		-3.5	dB
IL_{board}	Differential Insertion Loss from T_{ball} to R_{ball} at Nyquist frequency		-19.5	dB

The boundary of insertion loss for EDR25-LR between reference point T_{die} and R_{die} is illustrated in Figure 5-35 and Equation 5. The die parasitic capacitance should be included for this insertion loss.

$$IL_{max} = \begin{cases} -(0.91 + 4.2\sqrt{\frac{f \times 25}{f_b}} + 1.3\frac{f \times 25}{f_b}), & 50MHz \le f < \frac{f_b}{2} \\ 11.75 - 3.5\frac{f \times 25}{f_b}\,, & \frac{f_b}{2} \le f \le f_b \end{cases}$$

$$IL_{min} = \begin{cases} 0\,, & 50MHz \le f \le\ 1GHz \\ -\frac{1}{3}(f-1)\,, & 1GHz < f \le\ 17.5GHz \\ -5.5, & 17.5GHz < f \le\ f_b \end{cases}$$

$$Note: f_b\ is\ 25G\ or\ 20G$$

Equation 5: Channel Insertion Loss Mask for EDR25-LR Link

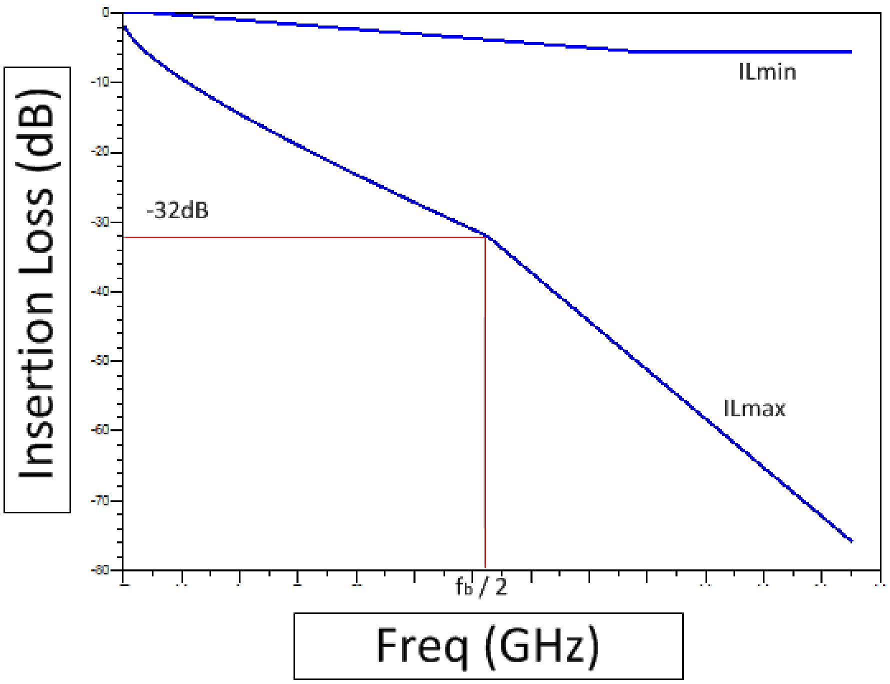


Figure 5-35: Insertion Loss Mask for EDR25-LR

5.3.6.2.2 Differential Return Loss of Channel (Informative)

The receiver differential Return loss from T_{ball} to R_{ball} is defined by these equations and Figure 5-36 and Equation 6.

$$RL_{max} = \begin{cases} -12\ dB, & 50MHz \le f < \frac{f_b}{4} \\ -12 + 15\ log_{10}({}^{4f}/_{f_b})\,, & \frac{f_b}{4} \le f \le\ f_b \end{cases}$$

Equation 6: Channel Return Loss Mask for EDR25-LR Link

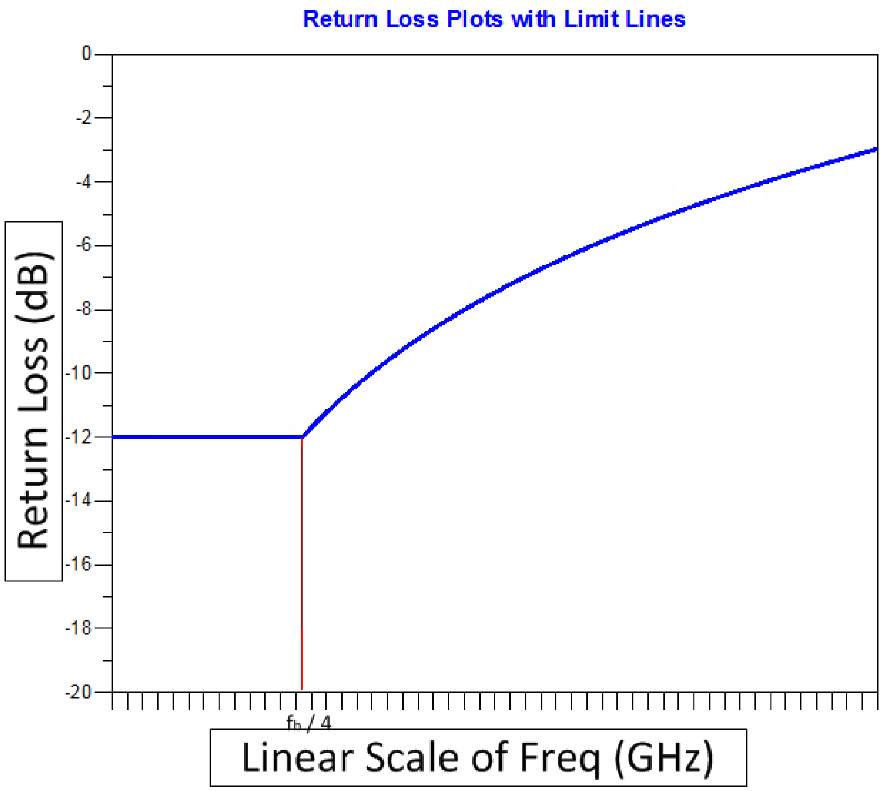


Figure 5-36: Channel Differential Return Loss Mask for EDR25-LR Link

5.3.6.2.3 Insertion Loss Deviation (Informative)

The insertion loss deviation, ILD, is the difference between the measured insertion IL and fitted insertion loss IL_{fitted}. For fitted insertion loss definitions, please refer to CEI-25G-LR.

$$ILD = IL - IL_{fitted}$$

The insertion loss deviation, LD, is required to be within the ranges which is defined by the equations (11-4) and (11-5) in CEI-25G-LR (see Reference Documents).

Figure 5-37 shows the compliance mask for ILD.

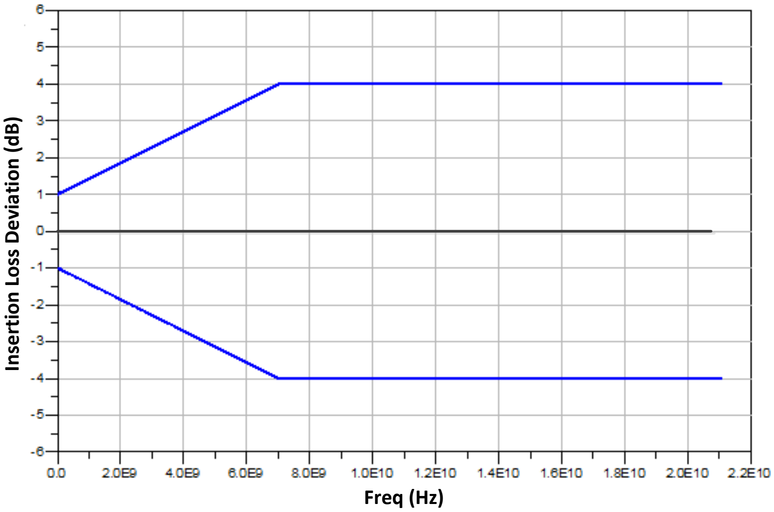


Figure 5-37: Insertion Deviation Mask for EDR25-LR PHY

ILD_{rms} is the RMS value of the ILD curve and calculated by the equations in CEI-25G-SR Section 11.2.6.4 (see Reference Documents). This is required to be less than $0.3dB_{rms}$ for valid channels.

5.3.6.2.4 Integrated Crosstalk Noise (Informative)

The integrated Crosstalk Noise, ICN, is calculated by using the integrated crosstalk noise method in CEI-25G-LR Section 11.2.6.6 (see Reference Documents) and the parameters of Table 5-31. It is required to be lower than max allowed ICN at given insertion loss in Figure 5-38. ICN needs to be calculated between reference point T_{die} and R_{die}.

Table 5-31: Channel Integrated Crosstalk Aggressor Parameters

Parameter	Symbol	Value	Units
Near-end aggressor peak to peak differential output amplitude	A_{nt}	1200 (max)	mV_{ppd}
Far-end aggressor peak to peak differential output amplitude	A_{ft}	1200 (max)	mV_{ppd}
Near-end aggressor 20 to 80% rise and fall times	T_{nt}	8 (min)	ps
Far-end aggressor 20 to 80% rise and fall times	T_{ft}	8 (min)	ps

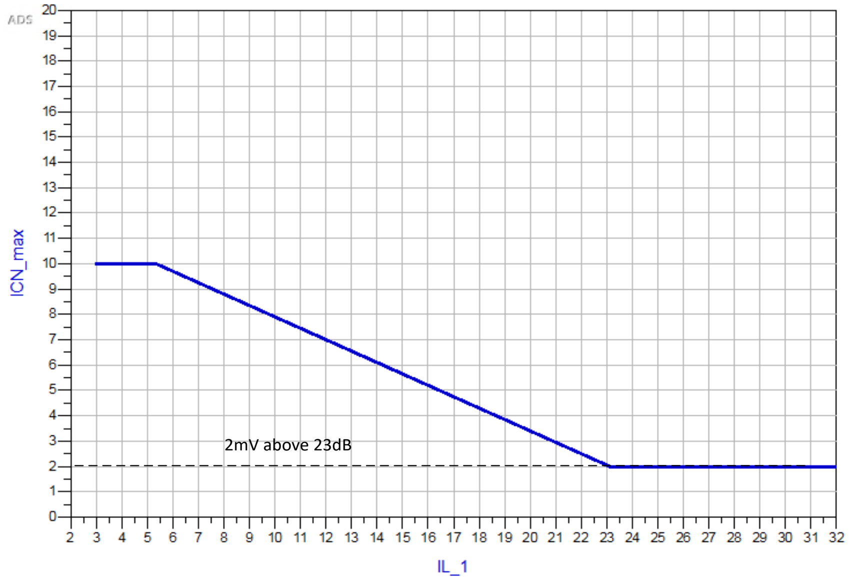


Figure 5-38: ICN$_{rms}$ mask vs Insertion Loss

5.4 EDR32 Electrical Specification

The EDR32 electrical specification applies to 2.5 GT/s, 5 GT/s, 8 GT/s, 16 GT/s and 32 GT/s. Table 5-1 shows how these rates are supported in *CCIX Base Specification 1.1.*

The EDR32 PHY must be compliant to the electrical specification of *PCI Express Base Specification 5.0* (see Reference Documents).

Chapter 6. Protocol Layer and Transport Layer DVSEC

6.1 Overview

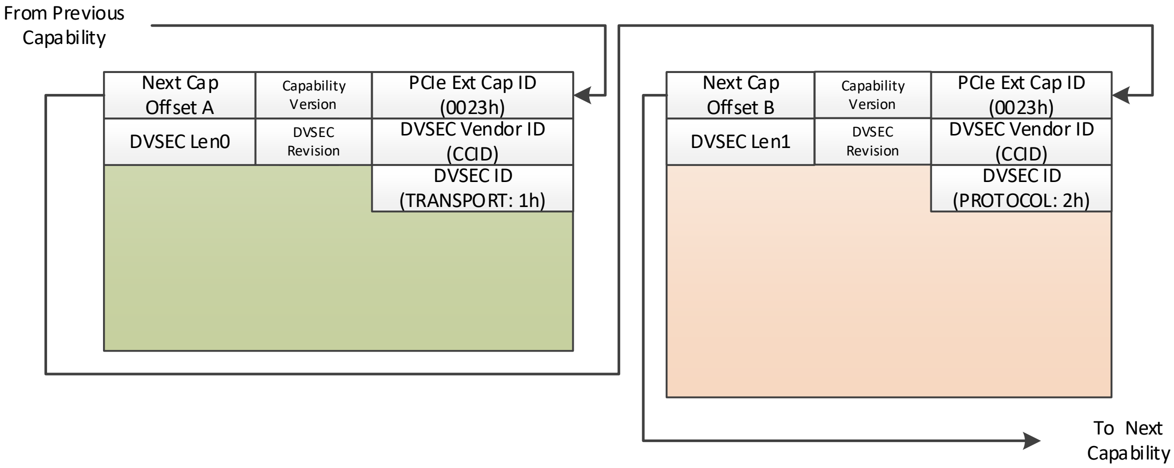


Figure 6-1: DVSECs Supported by CCIX Devices

Figure 6-1 shows an example of DVSECs supported by CCIX® devices. There is no requirement that the DVSECs be linked in any specific order, or that they be directly linked to each other. The Capability Version, illustrated in Figure 6-1 and Figure 6-5, must be 1h, consistent with its definition in the *PCI Express Base Specification*. The DVSEC Revision, illustrated in Figure 6-1 and Figure 6-5, must be 1h for this version of the CCIX Specification. The DVSEC Vendor ID, illustrated in Figure 6-1 and Figure 6-5, must be the CCID. The DVSEC ID, illustrated in Figure 6-1, must be consistent with the encodings described in Table 6-8.

Transport DVSEC contains Control and Status Registers (CSRs) for CCIX Physical, Data Link and Transaction Layers.

Protocol DVSEC contains CSRs for the CCIX Protocol Layer.

Both Protocol and Transport DVSECs can be implemented only in Root Ports, RCRBs, Switch Upstream Ports, Switch Downstream Ports, and all Endpoints (including Root Complex Integrated Endpoints). They are not applicable to PCI Express® PCI/PCI-X Bridges and Root Complex Event Collectors.

For Upstream Ports that are associated with a Multi-Function Device, the Transport DVSEC is implemented in Function 0 only. All other Functions, including VFs, use the settings from the Function 0 implementation.

All Register Attributes, and Next Capability and Control Offsets, in PCIe DVSEC structures described in Chapter 6 and 6.6.3, are consistent with their definitions in the *PCI Express Base Specification* (see Reference Documents). Next Capability and Control Offsets are relative to the beginning of the PCI-Compatible

Configuration Space, described in Table 7-35 of the *PCI Express Base Specification*; for example, the Next Capability Offset A and Next Capability Offset B illustrated in Figure 6-1. Register attributes are consistent with their definitions in Table 7-2 of the *PCI Express Base Specification*, for example HwInit, RsvdZ, and RsvdP illustrated in Figure 6-3.

6.2 Protocol Layer DVSEC

Figure 6-2 shows an example location of the CCIX Protocol Layer DVSEC within the PCI Express® (PCIe) Configuration space. As illustrated in Figure 6-2, the recommended CCIX Protocol Layer DVSEC location to speed up the discovery process of all CCIX capabilities, is as the Next Capability Offset of CCIX Transport DVSEC, when both DVSECs are present in Function0.

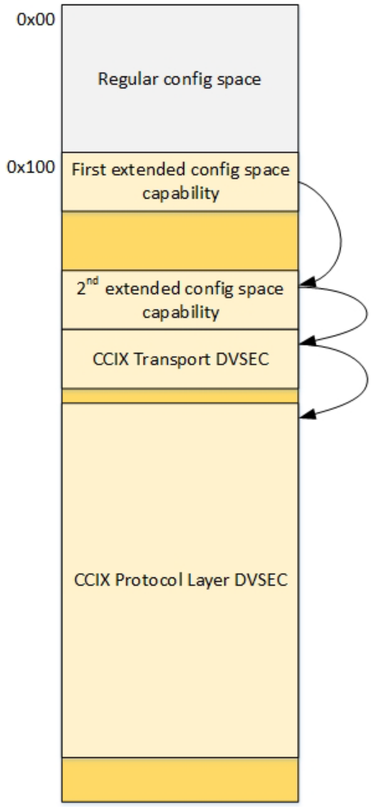

0x00
Regular config space
0x100
First extended config space capability
2nd extended config space capability
CCIX Transport DVSEC
CCIX Protocol Layer DVSEC

Figure 6-2: CCIX Protocol Layer DVSEC located in PCIe Configuration Space

6.2.1 Introduction to CCIX Protocol Layer DVSEC

Within the CCIX Protocol Layer DVSEC, Capabilities, Status, and Control fields are each separated into distinct DWords. Each DW type has further classification in terms of the allowed Register Attributes for each DW type:

- Capabilities & RO Status DW: HwInit, RO, ROS, RsvdZ.
- RW Status DW: RW1C, RW1CS, RsvdP.
- Control: RW, RWS, RsvdZ.

A CCIX Device provides separate offsets for the location of the Control data structures vs. Capabilities & Status data structures to independently expand the number of DWords available for each. Capabilities & RO Status DWords can only be interleaved with other DWords of the same two classifications and similarly, Control and RW Status DWords can only be interleaved with other DWords of the same two classifications. Figure 6-3 illustrates the distinction in DW types and the allowed Register attributes for the overall CCIX Protocol Layer DVSEC structure.

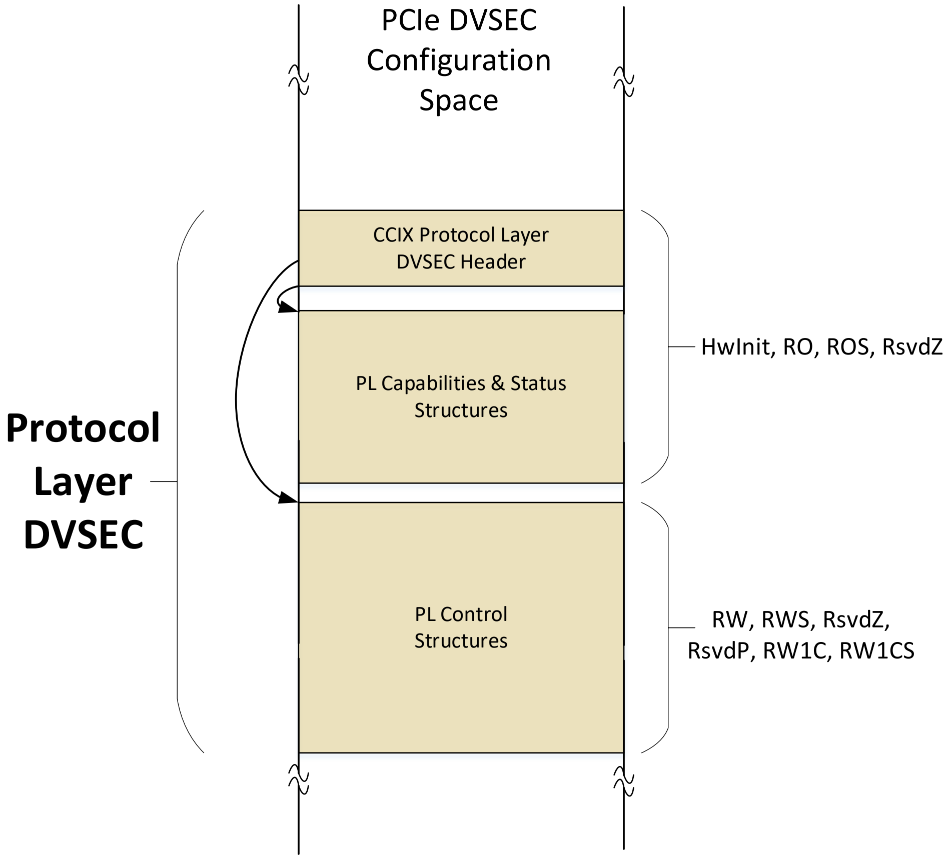

PCIe DVSEC Configuration Space
CCIX Protocol Layer DVSEC Header
PL Capabilities & Status Structures
PL Control Structures
Protocol Layer DVSEC
HwInit, RO, ROS, RsvdZ
RW, RWS, RsvdZ, RsvdP, RW1C, RW1CS

Figure 6-3: CCIX Protocol Layer DVSEC structure types

6.2.1.1 Location of CCIX Protocol Layer DVSEC

A CCIX Protocol Layer DVSEC header of a CCIX capable physical device must be placed in at least Function0 of all CCIX-capable PCIe Ports of that device. The Common, CCIX Port and CCIX Link Capabilities, Status and the CCIX Port and CCIX Link Control data structures must also be placed in Function0 of all CCIX capable PCIe Ports of that device. The Common Control and AF Properties data structures must only be placed in Function0 of all CCIX Primary Capable Ports of that device.

CCIX Protocol Layer DVSEC contains certain data structures that are chip-level, such as System Address Map (SAM) and ID Map (IDM) data structures, and these chip-level data structures must be placed in Function0. A physical device with a single PCIe Port must therefore have chip-level data structures accessible from Function0. A physical device with multiple CCIX capable PCIe Ports must declare at least one CCIX Port where chip-level data structures can be accessed (see Table 6-12).

These CCIX Ports are said to have Primary CCIX Port capability, with an indicator of that capability in CCIX Protocol Layer DVSEC.

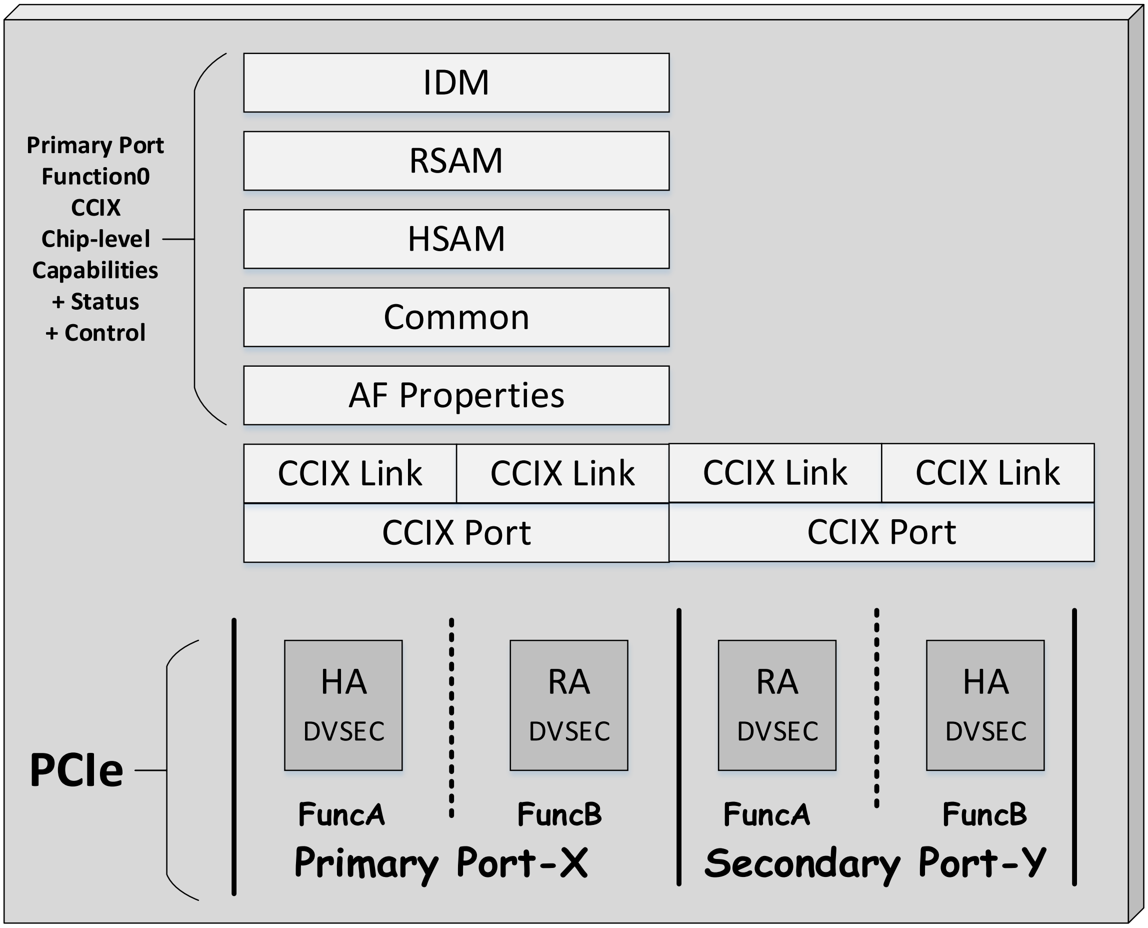


Figure 6-4: CCIX Protocol Layer DVSEC structures over various PCIe Ports and Functions

CCIX Configuration Software must enable only one CCIX Port as the Primary CCIX Port and discover and configure chip-level data structures only through that CCIX Port (see Table 6-14). All ports that are not enabled as Primary CCIX Ports, and all ports that do not have Primary CCIX Port capability, are designated as Secondary

CCIX Ports. CCIX Agent DVSEC Structures can be located in any CCIX Port, Primary or Secondary, and on any Function Number within that CCIX Port, including Function0.

As illustrated in Figure 6-4, chip-level data structures are only configured in Primary CCIX Port-X Function0. A CCIX Device may declare additional DVSEC data-structures for specific CCIX Agents, illustrated as HA DVSEC in CCIX Port-X FunctionA, RA DVSEC in CCIX Port-X FunctionB, RA DVSEC in CCIX Port-Y FunctionA, and HA DVSEC in CCIX Port-Y FunctionB in Figure 6-4.

Figure 6-4 illustrates an example CCIX Device, however, devices are permitted to have CCIX Port counts other than 2, per-Port CCIX Link Counts other than 2, and RA, HA, and SA Agents and Agent counts other than what is illustrated.

Even though Figure 6-4 only illustrates a CCIX Device with one CCIX Agent DVSEC structure per function, CCIX Devices are also permitted to have more than one CCIX Agent DVSEC structure per function.

6.2.1.2 PCIe Function Level Reset (FLR)

CCIX Protocol Layer DVSEC structures must retain their programmed values during, and following a FLR. FLR shall not affect the CCIX Protocol Layer, which must continue to operate consistent with its DVSEC configuration.

6.2.1.3 PCIe ARI and SR-IOV

For CCIX Devices that support PCIe ARI, CCIX Agent DVSEC structures can be located in any of the 256 Functions.

For CCIX Devices that support PCIe SR-IOV, CCIX Protocol Layer DVSEC structures must only be located in a Physical Function (PF). A PF may also contain Capabilities, Control, and Status, for Acceleration Functions (AFs). There is no architected binding defined in CCIX, between the AF's Register space and PCIe SR-IOV Virtual Functions (VFs). VFs also do not have an architected binding to Protocol Layer DVSEC Space. Any such binding between PCIe VFs and the AF's Register space, or Protocol Layer DVSEC Space, is implementation specific. However, the mechanisms for how such a binding is to be achieved is defined in the CCIX Software Architecture Guide.

6.2.1.4 CCIX Protocol Layer DVSEC Header

Figure 6-5 shows the layout of the CCIX Protocol Layer (PL) DVSEC Header. Register fields at Byte Offset 00h conform to the definitions in Section 7.9.6 of the *PCI Express Base Specification* (see Reference Documents).

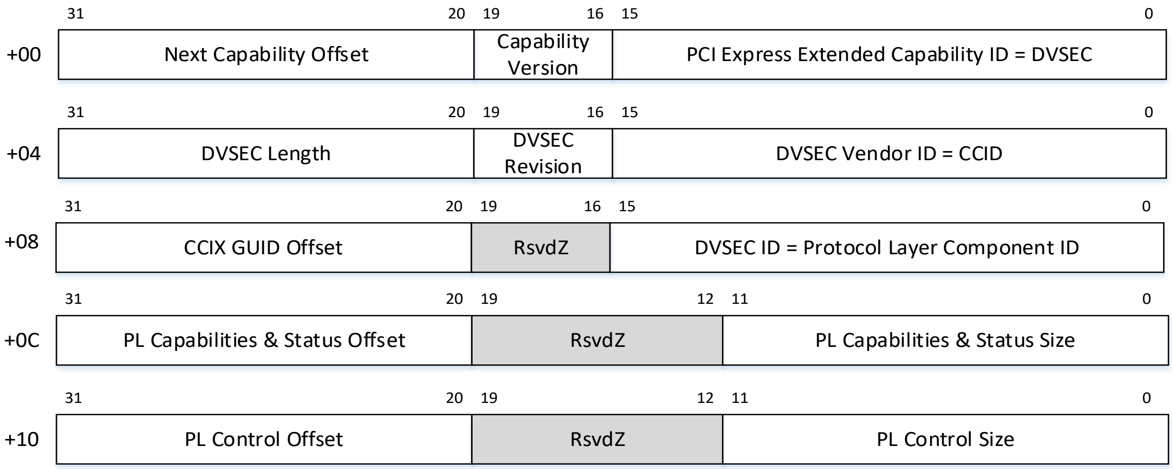


Figure 6-5: CCIX Protocol Layer DVSEC Header

The CCIX PL DVSEC Header includes a CCIX GUID Offset field at Byte Offset 08h for discovering the location to override the hardware default value of CCID. Apart from advertising the CCID, Revision, and CCIX Protocol Layer DVSEC Length, the CCIX Protocol Layer DVSEC Header also has registers indicating the offset and size for the CCIX Protocol Layer's Capabilities & Status structures and Control structures, also illustrated as pointers from the DVSEC Header in Figure 6-5.

Table 6-1 describes the CCIX PL DVSEC Header Register fields at Byte Offset 04h.

Table 6-1: CCIX PL DVSEC Header Register fields at Byte Offset 04h

Bit Location	Register Description	Attributes
15:0	**CCID** This field indicates the CCIX Consortium ID value. The value in the CCID field is either CCUV or is overridden with COV (see Table 6-3). CCIX Device initializes CCID to CCUV after reset (except FLR).	RO
19:16	**DVSECRevID** This field indicates the DVSEC Revision ID value of 1h consistent with its definition in the *PCI Express Base Specification*.	RO
31:20	**DVSECLength** This field indicates size of the DVSEC structure. The definition of the DVSEC Length field conforms to the definition in Section 7.9.6 of the *PCI Express Base Specification*.	RO

Table 6-2 describes the CCIX PL DVSEC Header Register fields at Byte Offset 08h.

Table 6-2: CCIX PL DVSEC Header Register fields at Byte Offset 08h

Bit Location	Register Description	Attributes
15:0	**PLCompID** This field indicates the CCIX Protocol Layer ComponentID value of 0002h.	RO
19:16	Reserved and Zero	RsvdZ
31:20	**CCIXGUIDOffset** This field indicates the CCID Override Structure Offset in number of bytes. The offset must be in integer multiples of DW. The CCID Override Structure must remain within the size of CCIX Protocol Layer DVSEC.	RO

Figure 6-6 illustrates the CCID Override Structure located at the CCIX GUID Offset. The CCID Override Structure contains the CCIX GUID and the programmable field to override the CCID.

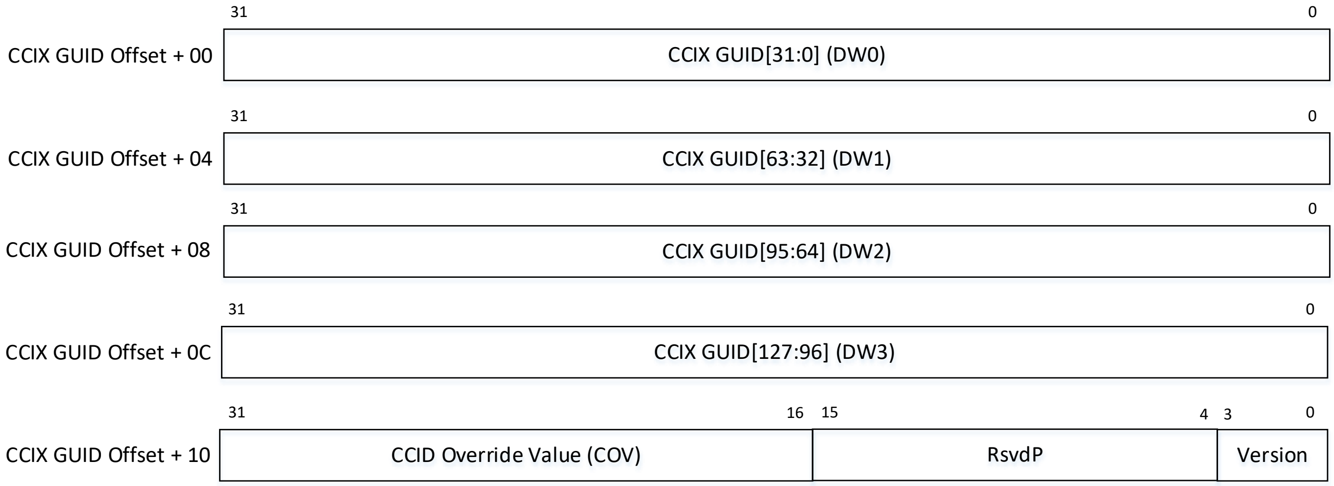


Figure 6-6: CCID Override Structure

Table 6-3 describes the CCIX Globally Unique Identifier (GUID) located at CCID Override Structure's fields from Byte Offset 00h through Byte Offset 0Ch.

Table 6-3: CCID Override Structure Register fields from Byte Offset 00h through Byte Offset 0Ch

Bit Location	Register Description	Attributes
127:0 or DW3:DW0	**CCIXGUID** This field indicates the Globally Unique Identifier (GUID) value for CCIX: C3CB993B-02C4436F-9B68D271-F2E8CA31 Discovery by CCIX Configuration Software of the CCIXGUID value at the CCIX GUID Offset confirms that the PCIe DVSEC structure is the CCIX Protocol Layer DVSEC structure. Discovery by CCIX Configuration Software of the CCIXGUID value at the CCIX GUID Offset also provides software the location to override the existing CCID value, i.e. the COV in DW4 of the CCID Override Structure (see Table 6-4).	RO

Table 6-4 describes the CCID Override Structure fields at Byte Offset 10h.

Table 6-4: CCID Override Structure Register fields at Byte Offset 10h

Bit Location	Register Description	Attributes
3:0	**CCIDOverrideStructureVersion** This field indicates the CCID Override Structure's Version Number. All CCIX Devices based on this specification must report a CCIDOverrideStructureVersion value of 1h.	RO
15:4	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
31:16	**COV** This field controls the CCID override value to be subsequently reflected in PCIe Compatible Header Protocol Layer Messages as well as Protocol Layer DVSEC and Transport Layer DVSEC headers. CCIX configuration software must ensure COV is the same across the CCIX system and doesn't conflict with a Vendor ID on the PCIe side of that same system. 0000h: CCID has not been overrridden. 0001h: FFFFh: Encodings for overriding CCID with values 1 through 65535. A non-zero COV value written by Software must be replicated by the CCIX Device in the CCID field of all Protocol Layer DVSEC and Transport Layer DVSEC headers on the same CCIX Device. This must result in the same non-zero COV value being returned when Software subsequently reads the CCID field of any of the CCIX DVSEC headers on that CCIX Device. All PCIe Compatible Header CCIX Messages sent by that CCIX Device must also reflect the same non-zero COV value in the CCID field of those messages.	RW

Table 6-5 describes the CCIX Protocol Layer Capabilities & Status Pointer (PLCapStatPtr) Register at Byte Offset-0Ch.

Table 6-5: CCIX PLCapStatPtr Register at Byte Offset-0Ch

Bit Location	Register Description	Attributes
11:0	**PLCapStatSize** This field indicates the PL Capabilities & Status structure size in number of DW.	RO
19:12	Reserved and Zero	RsvdZ
31:20	**PLCapStatOffset** This field indicates the PL Capabilities & Status structure Offset in number of bytes. The offset must be in integer multiples of DW. The offset + size must remain within the size of CCIX Protocol Layer DVSEC.	RO

Table 6-6 describes the CCIX Protocol Layer Control Pointer (PLCntlPtr) Register at Byte Offset-10h.

Table 6-6: CCIX PLCntlPtr Register at Byte Offset-10h

Bit Location	Register Description	Attributes
11:0	**PLCntlSize** This field indicates the PL Control structure size in number of DW.	RO
19:12	Reserved and Zero	RsvdZ
31:20	**PLCntlOffset** This field indicates the PL Control structure Offset in number of bytes. The offset must be in integer multiples of DW. The offset + size must remain within the size of CCIX Protocol Layer DVSEC.	RO

6.2.1.5 Sequence for Capabilities & Status and Control Structures

Figure 6-7 illustrates the sequence for CCIX Protocol Layer Components' Capabilities & Status structures and Control structures located in Primary CCIX Port Function0.

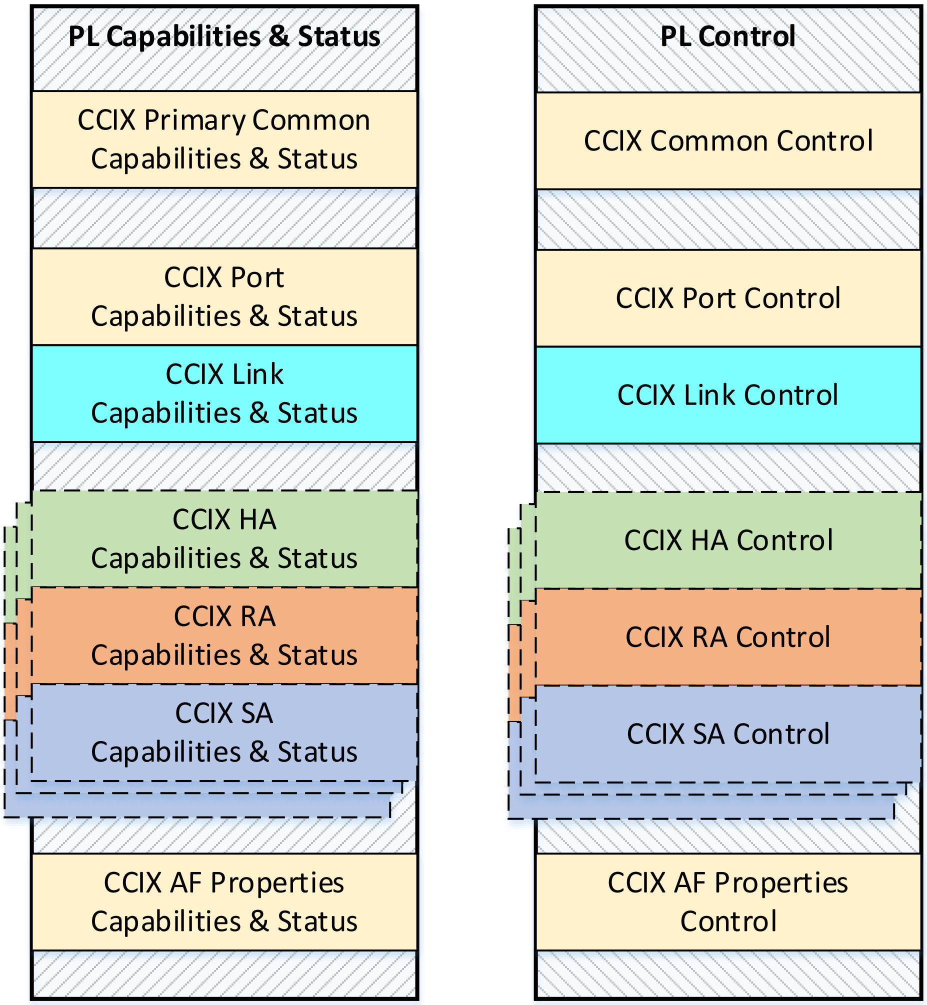
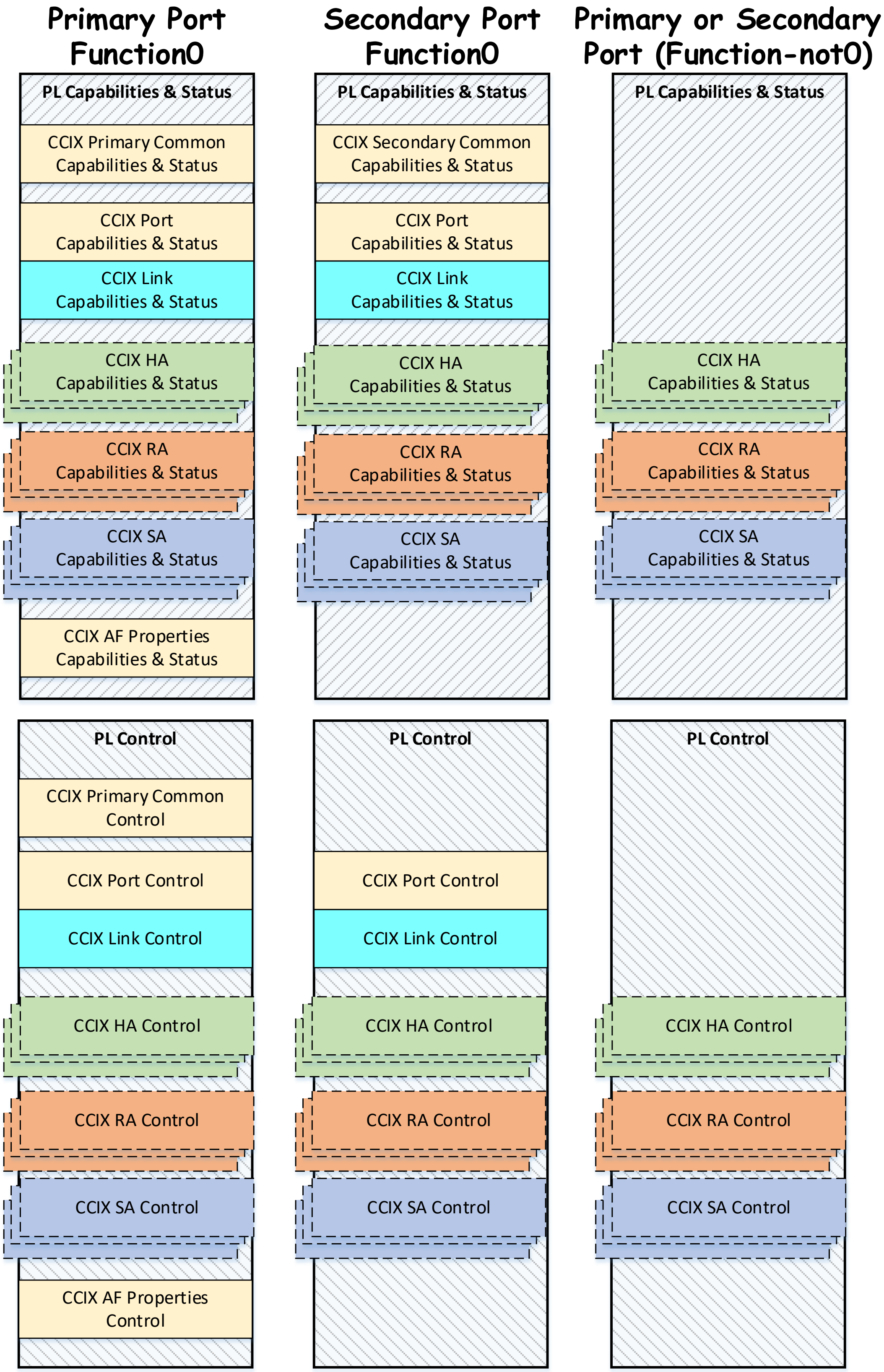

PL Capabilities & Status
CCIX Primary Common Capabilities & Status
CCIX Port Capabilities & Status
CCIX Link Capabilities & Status
CCIX HA Capabilities & Status
CCIX RA Capabilities & Status
CCIX SA Capabilities & Status
CCIX AF Properties Capabilities & Status
PL Control
CCIX Common Control
CCIX Port Control
CCIX Link Control
CCIX HA Control
CCIX RA Control
CCIX SA Control
CCIX AF Properties Control

Figure 6-7: Sequence of CCIX Protocol Layer Component Structures

6.2.1.5.1 Sequence for Multiport CCIX Devices

While Common, CCIX Port and CCIX Link structures are unique per CCIX capable PCIe port, any Function on a Primary or Secondary CCIX Port is permitted to have CCIX Agent structures as shown in Figure 6-8. Thus, CCIX Device Error Control and Status are communicated via the Primary CCIX Port. Primary CCIX Ports also communicate Error Control and Status specific to CCIX Components on that CCIX Port. Secondary CCIX Ports communicate Error Control and Status specific to CCIX Components on that CCIX Port.


Primary Port Function0
Secondary Port Function0
Primary or Secondary Port (Function-not0)
PL Capabilities & Status
CCIX Primary Common Capabilities & Status
CCIX Port Capabilities & Status
CCIX Link Capabilities & Status
CCIX HA Capabilities & Status
CCIX RA Capabilities & Status
CCIX SA Capabilities & Status
CCIX AF Properties Capabilities & Status
PL Capabilities & Status
CCIX Secondary Common Capabilities & Status
CCIX Port Capabilities & Status
CCIX Link Capabilities & Status
CCIX HA Capabilities & Status
CCIX RA Capabilities & Status
CCIX SA Capabilities & Status
PL Capabilities & Status
CCIX HA Capabilities & Status
CCIX RA Capabilities & Status
CCIX SA Capabilities & Status
PL Control
CCIX Primary Common Control
CCIX Port Control
CCIX Link Control
CCIX HA Control
CCIX RA Control
CCIX SA Control
CCIX AF Properties Control
PL Control
CCIX Port Control
CCIX Link Control
CCIX HA Control
CCIX RA Control
CCIX SA Control
PL Control
CCIX HA Control
CCIX RA Control
CCIX SA Control

5

Figure 6-8: Structures for multi-Port CCIX Devices

6.2.1.6 Capabilities & Status Structure and Control Structure

Figure 6-9 shows the overall layout of a CCIX Component's Capabilities & Status Registers with Byte Offset-00h containing the CCIX Component's Capabilities & Status Register Header (<ComponentName>CapStatHdr). Similarly, Figure 6-10 shows the overall layout of a CCIX Component's Control Registers with Byte Offset-00h containing the CCIX Component's Control Header (<ComponentName>CntlHdr). Each Capabilities & Status header and Control header contains a field indicating the offset of the next Capabilities & Status header (<ComponentName>NextCapStatHdrPtr) and Control header (<ComponentName>NextCntlHdrPtr) respectively.

Only the CCIX Protocol Layer DVSEC Header follows the PCIe DVSEC format. The CCIX Protocol Layer Components' Capabilities & Status Header, as shown in Figure 6-9, and Control Header, as shown in Figure 6-10, do not follow the PCIe DVSEC format. The Component Header size of 4-bytes is an efficient alternative to the first 12-bytes of the CCIX Protocol Layer DVSEC Header.

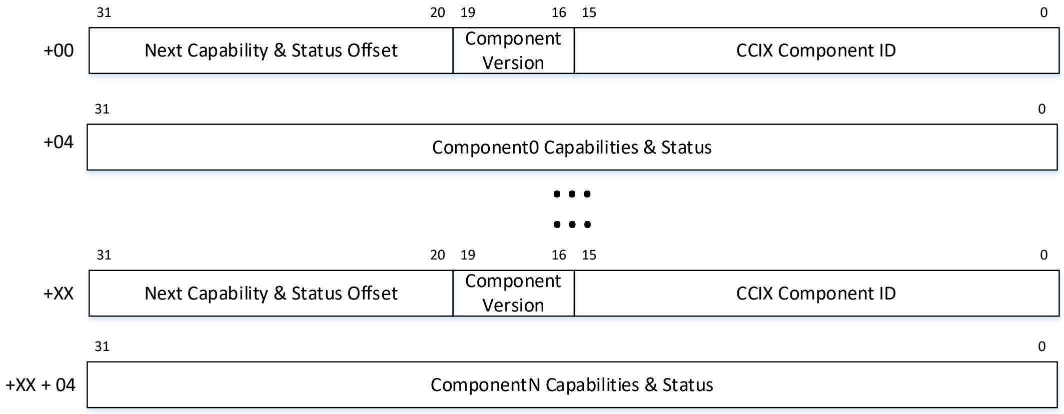

31
20
19
16
15
0
+00
Next Capability & Status Offset
Component Version
CCIX Component ID
31
0
+04
Component0 Capabilities & Status
31
20
19
16
15
0
+XX
Next Capability & Status Offset
Component Version
CCIX Component ID
31
0
+XX + 04
ComponentN Capabilities & Status

Figure 6-9: CCIX Component's Capabilities & Status Registers

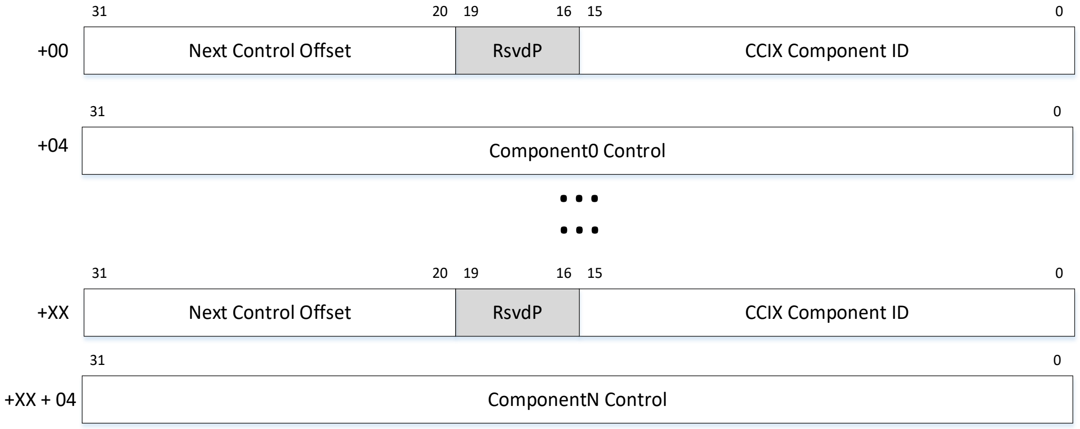


Figure 6-10: CCIX Component's Control Registers

Table 6-7 describes the Capabilities & Status Version field.

Table 6-7: Capabilities & Status Version field

Bit Location	Register Description	Attribute
19:16	**ComponentVersionSupported** 0h: CCIX Version 1.0 Supported. All other encodings: Reserved. All CCIX Components based on this specification must report 0h CCIX Version Capability.	RO

Table 6-8 describes the encoding for the Capabilities & Status and Control structure's CCIX Component ID field contained in Figure 6-9 and Figure 6-10 respectively. Table 6-8 also describes the encoding for the CCIX Protocol Layer DVSEC ID field contained in Figure 6-5, at Byte Offset-08h.

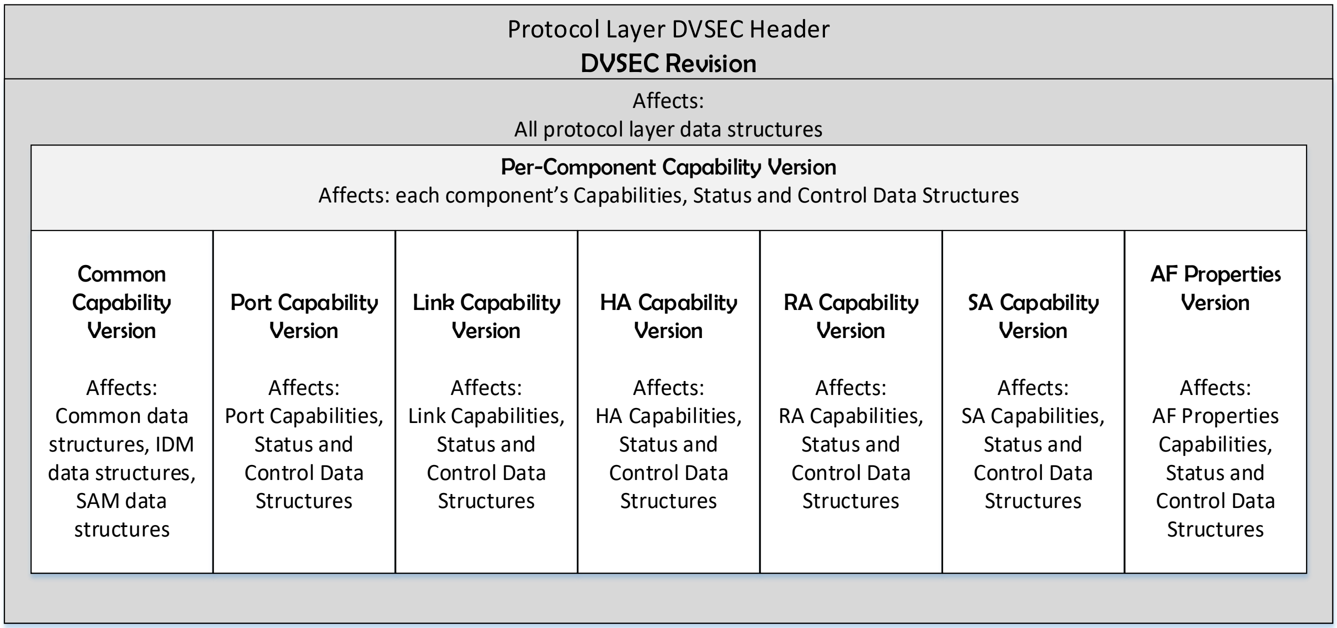
Table 6-8: CCIX Component ID Encodings

Component ID	CCIX Component
0000h	CCIX General This allows for System Wide CCIX Configuration
0001h	CCIX Transport DVSEC
0002h	CCIX Protocol Layer DVSEC
0003h	CCIX Protocol Layer Common structure
0004h	CCIX Protocol Layer Port structure
0005h	CCIX Protocol Layer Link structure
0006h	CCIX Home Agent structure
0008h	CCIX Request Agent structure
000Ah	CCIX Subordinate Agent structure
000Bh	CCIX DVM Agent structure
000Ch	CCIX Acceleration Function Properties structure
	All other encodings are reserved

6.2.1.7 Version Numbers and their impact on data structure definition

A version number of a particular CCIX Protocol Layer DVSEC data structure indicates the format and size of the data structure as well as the definition of the fields within the data structure. The version number also indicates the data structure's mandatory and optional fields and applies equally to the Capabilities & Status, and Control data structures in CCIX Protocol Layer DVSEC.

As shown in Figure 6-11, the CCIX DVSEC Revision in the CCIX Protocol Layer DVSEC Header governs the definition of all protocol layer data structures. Each CCIX Component also has a Capability version in the per-component data structure header as shown in Figure 6-9. The CCIX Component's Capability version governs the format and size of that CCIX Component's data structure as well as the definition of the fields within the data structure.



Protocol Layer DVSEC Header
DVSEC Revision

Affects:
All protocol layer data structures

Per-Component Capability Version
Affects: each component's Capabilities, Status and Control Data Structures

Common Capability Version	Port Capability Version	Link Capability Version	HA Capability Version	RA Capability Version	SA Capability Version	AF Properties Version
Affects: Common data structures, IDM data structures, SAM data structures	Affects: Port Capabilities, Status and Control Data Structures	Affects: Link Capabilities, Status and Control Data Structures	Affects: HA Capabilities, Status and Control Data Structures	Affects: RA Capabilities, Status and Control Data Structures	Affects: SA Capabilities, Status and Control Data Structures	Affects: AF Properties Capabilities, Status and Control Data Structures

Figure 6-11: Version Numbers and their impact on data structure definition

6.2.1.8 CCIX Topology creation via Address Space and CCIX AgentID Name Space Allocation

This section provides an overview of the addressing and routing data structures resulting in a topology of CCIX Devices, defined by the addressing and routing attributes between them.

Sections 6.2.1.8.1 through 6.2.1.8.3 describe the Global ID Map (G-IDM), Global RA-to-HA System Address Map (G-RSAM), and Global HA-to-SA System Address Map (G-HSAM), respectively.

Section 6.2.1.8.4 describes the CCIX Device's local view of the G-IDM, G-RSAM, and G-HSAM structures. The DVSEC specification requires, and therefore describes, the CCIX Device's local view of G-IDM, G-RSAM, or G-HSAM DVSEC data structures from Section 6.2.2 onwards.

G-IDM, G-RSAM, and G-HSAM are data structures maintained by software in memory and do not require their DVSEC equivalent. As such, other than for explanation purposes, the DVSEC specification does not require, and therefore does not contain detailed descriptions of G-IDM, G-RSAM, or G-HSAM data structures from Section 6.2.2 onwards.

6.2.1.8.1 Global ID Map (G-IDM)

A connected graph is created of the discovered CCIX Devices, with the connection attributes based on a combination of CCIX Ports on those CCIX Devices, and the transport connections between them. A CCIX topology is then defined, where the topology nodes are the enumerated CCIX Devices. Based on the location of the enumerated CCIX Agents in the CCIX topology, a Global ID Map (G-IDM) data structure describes the ID routing attributes between the enumerated CCIX AgentIDs across the CCIX topology.

6.2.1.8.2 Global RA-to-HA System Address Map (G-RSAM)

Figure 6-12 shows the Global RA-to-HA System Address Map (G-RSAM) generated within the System Address Space. The G-RSAM is typically generated by CCIX Configuration Software after identifying the capabilities of all CCIX Memory Devices as well as the addressing capabilities of all CCIX Requesting Devices.

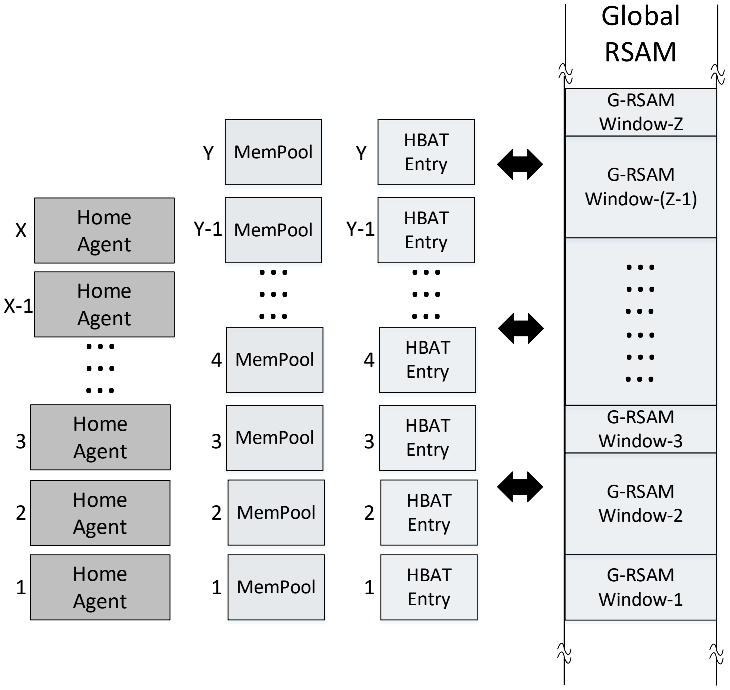


Figure 6-12: G-RSAM and its relation to HAs, MemPools, and HBAT Entries

G-RSAM and its relation to HAs, MemPools, and HA Base Address Table (HBAT) Entries; as illustrated in Figure 6-12:

- For a given CCIX topology with 1-to-X number of Home Agents discovered, a corresponding 1-to-Y number of Memory Pool Capabilities & Status data structures are declared across all the Home Agents. Each Home Agent must declare at least one Memory Pool Entry, thus Y ≥ X.
- The 1-to-Y number of Memory Pool Entries declare the memory attributes hosted by their corresponding Home Agents. Memory attributes declared include the Memory Size, Memory Type, and addressing capability of the Memory Pool.

- The 1-to-Y number of Memory Pool Capabilities & Status data structures must have their corresponding 1-to-Y number of Home Agent Base Address Table Entry, or HBAT Entry control structures.
- A Global RA-to-HA System Address Map (G-RSAM) is generated with 1-to-Z number of G-RSAM Windows. Each G-RSAM Window is defined by a 4GB aligned Start and End Address.
- The Memory in the 1-to-Y Memory Pools are all mapped to the G-RSAM by programming the HBAT Entry control structures. Depending on the attributes declared in the 1-to-Y number of Memory Pool Entries, an HBAT Entry can be programmed with the addresses contained in one G-RSAM Window, or multiple HBAT Entries can be programmed with the addresses contained in one G-RSAM Window, thus Z ≤ Y.

6.2.1.8.3 Global HA-to-SA System Address Map (G-HSAM)

Figure 6-13 shows the Global HA-to-SA System Address Map (G-HSAM) generated within the System Address Space. The G-HSAM is typically generated by CCIX Configuration Software after identifying the capabilities of all CCIX Subordinate Agents, and all CCIX Home Agents with Memory Expansion Capabilities.

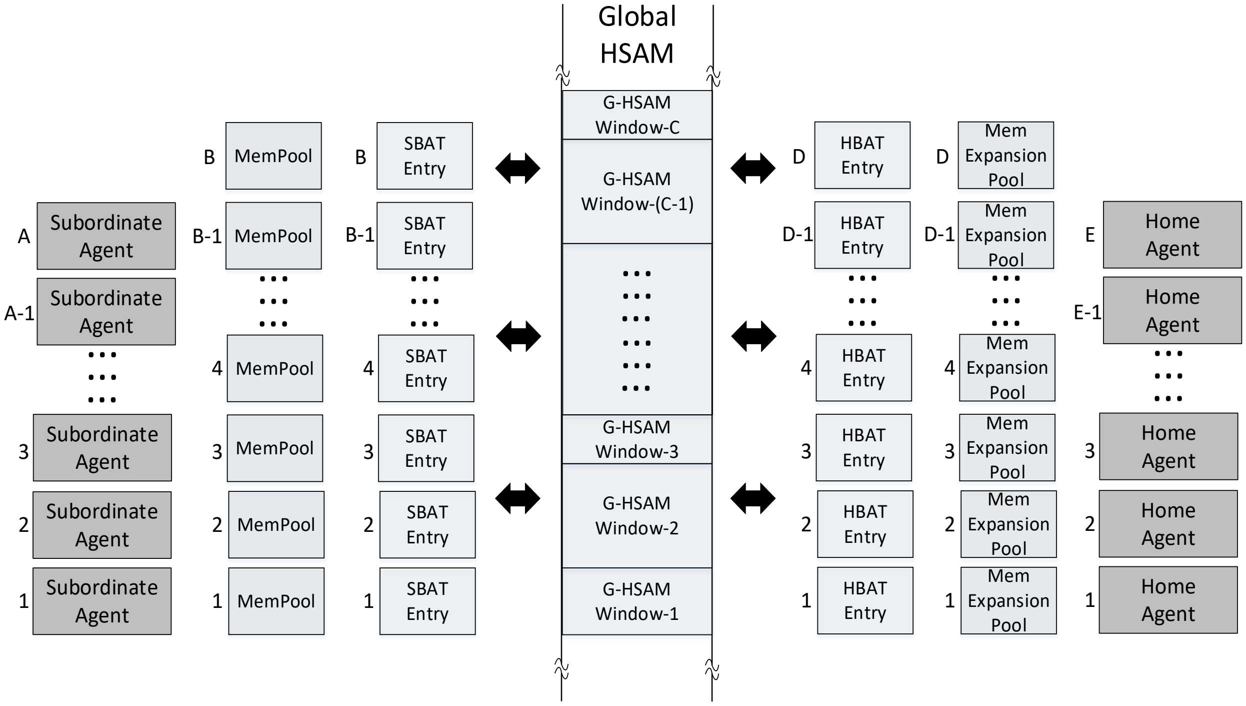


Figure 6-13: G-HSAM and its relation to HAs with Memory Expansion Pools, and SAs

G-HSAM and its relation to SAs, SA MemPools, and SA Base Address Table (SBAT) Entries; as illustrated in Figure 6-13:

- For a given CCIX topology with 1-to-A number of Subordinate Agents discovered, a corresponding 1-to-B number of Memory Pool Capabilities & Status data structures are declared across all the Subordinate Agents. Each Subordinate Agent must declare at least one Memory Pool Entry, thus B ≥ A.
- The 1-to-B number of Memory Pool Entries declare the memory attributes hosted by their corresponding Subordinate Agents. Memory attributes declared include the Memory Size and Memory Type of the Memory Pool.

- The 1-to-B number of Memory Pool Capabilities & Status data structures must have their corresponding 1-to-B number of Subordinate Agent Base Address Table Entry, or SBAT Entry control structures.
- A Global HA-to-SA System Address Map (G-HSAM) is generated with 1-to-C number of G-HSAM Windows. Each G-HSAM Window is defined by a 4GB aligned Start and End Address.
- The Memory in the 1-to-B Memory Pools are all mapped to the G-HSAM by programming the SBAT Entry control structures. An SBAT Entry can be programmed with the addresses contained in one G-HSAM Window, or multiple SBAT Entries can be programmed with the addresses contained in one G-HSAM Window, thus C ≤ B

G-HSAM and its relation to HAs with Memory Expansion Pools, and SAs; as illustrated in Figure 6-13:

- For a given CCIX topology with 1-to-E number of Home Agents discovered with Memory Expansion Capability, a corresponding 1-to-D number of Expansion Memory Pool Capabilities & Status data structures are declared across all the Home Agents. Each Home Agent that declares Memory Expansion Capability must declare at least one Memory Expansion Pool Entry, thus D ≥ E.
- The 1-to-D number of Memory Pool Entries declare the memory attributes hosted by their corresponding Home Agents. Memory attributes declared include the Memory Expansion Size.
- The 1-to-D number of Memory Pool Capabilities & Status data structures must have their corresponding 1-to-D number of Home Agent Base Address Table Entry, or HBAT Entry control structures.
- The Global HA-to-SA System Address Map (G-HSAM) generated with 1-to-C number of G-HSAM Windows must be mapped to the 1-to-D Memory Expansion Pools by programming the HBAT Memory Expansion Entry control structures. An HBAT Memory Expansion Entry can be programmed with the addresses contained in one G-HSAM Window, or multiple HBAT Memory Expansion Entries can be programmed with the addresses contained in one G-HSAM Window, thus C ≤ D

Multiple BAT Entries mapped to a single G-HSAM Window:

- The Expansion Memory of multiple Home Agents can come from a single Subordinate Agent because multiple HBAT Memory Expansion Entries can be programmed with the addresses contained in one G-HSAM Window, while at the same time, a single SBAT Entry can be programmed with the addresses contained in the same G-HSAM Window.
- The Expansion Memory of a single Home Agent can come from multiple Subordinate Agents because multiple SBAT Entries can be programmed with the addresses contained in one G-HSAM Window, while at the same time, a single HBAT Memory Expansion Entry can be programmed with the addresses contained in the same G-HSAM Window.

6.2.1.8.4 CCIX Device view of G-IDM, G-RSAM, and G-HSAM

The CCIX Device’s IDM and SAM data structures are set up based on the CCIX Device’s location in the CCIX topology, its routing attributes, and the types of Agents and their enumerated AgentIDs.

The CCIX Device’s IDM Table contains the CCIX Device’s view of the G-IDM, i.e. it contains the ID routing attributes to all enumerated AgentIDs in the topology, based on the CCIX Device’s location in that topology relative to the location of all the enumerated AgentIDs in that topology.

A CCIX Device's optional Snoop Response IDM, or SR-IDM, Table also contains the CCIX Device's view of the G-IDM, the data structure being referenced for the routing of Snoop Response ID routed packets from that CCIX Device in a Mesh Topology.

The CCIX Device's RSAM Table contains the CCIX Device's view of the G-RSAM, i.e. it contains the address routing attributes to all enumerated Home Agent IDs (HAIDs) in the topology, based on the CCIX Device's location in that topology relative to the location of all the Home Agents in that topology.

A CCIX Device references an RSAM Table if there is a Request Agent on that device, or the CCIX Device can perform RA-to-HA CCIX Packet Port-to-Port forwarding.

The CCIX Device's HSAM Table contains the CCIX Device's view of the G-HSAM, i.e. it contains the address routing attributes to all enumerated Subordinate Agent IDs (SAIDs) in the topology, based on the CCIX Device's location in that topology relative to the location of all the Subordinate Agents in that topology. A CCIX Device references an HSAM Table if there is a Home Agent with Memory Expansion enabled on that device, or the CCIX Device can perform HA-to-SA CCIX Packet Port-to-Port forwarding.

The host is not required to implement the CCIX Protocol Layer DVSEC data structure. The host may use equivalent host-specific methods to support features such as CCIX capability discovery, enumeration, routing and error reporting.

6.2.1.9 CCIX Component Address and ID based routing tables

CCIX Components within a CCIX Device require Address and ID routing attributes in order to determine the CCIX Port or local (CCIX Agent) destination of CCIX packets. These Address and ID routing attributes are described via control structures called SAM and IDM tables respectively.

The Request Agent Address Map across the CCIX System is described in RSAM Tables which must be present in CCIX Devices that contain Request Agents, and intermediate points that are capable of forwarding traffic from Request Agents. Further details of RSAM Table usage is described in Section 6.2.2.3.4.

The Home Agent Address Map across the CCIX System is described in the HSAM Tables which must be present in CCIX Devices that contain Home Agents that have Memory Expansion enabled, and intermediate points that are capable of forwarding traffic from Home Agents. Further details of HSAM Table usage is described in Section 6.2.2.3.4.

For CCIX Software to program the tables for the entire CCIX Device, Primary-capable CCIX Ports have the relevant SAM and IDM pointer registers as Primary CCIX Port Extended Capabilities located in the Common Capabilities & Status structure as shown in Figure 6-14.

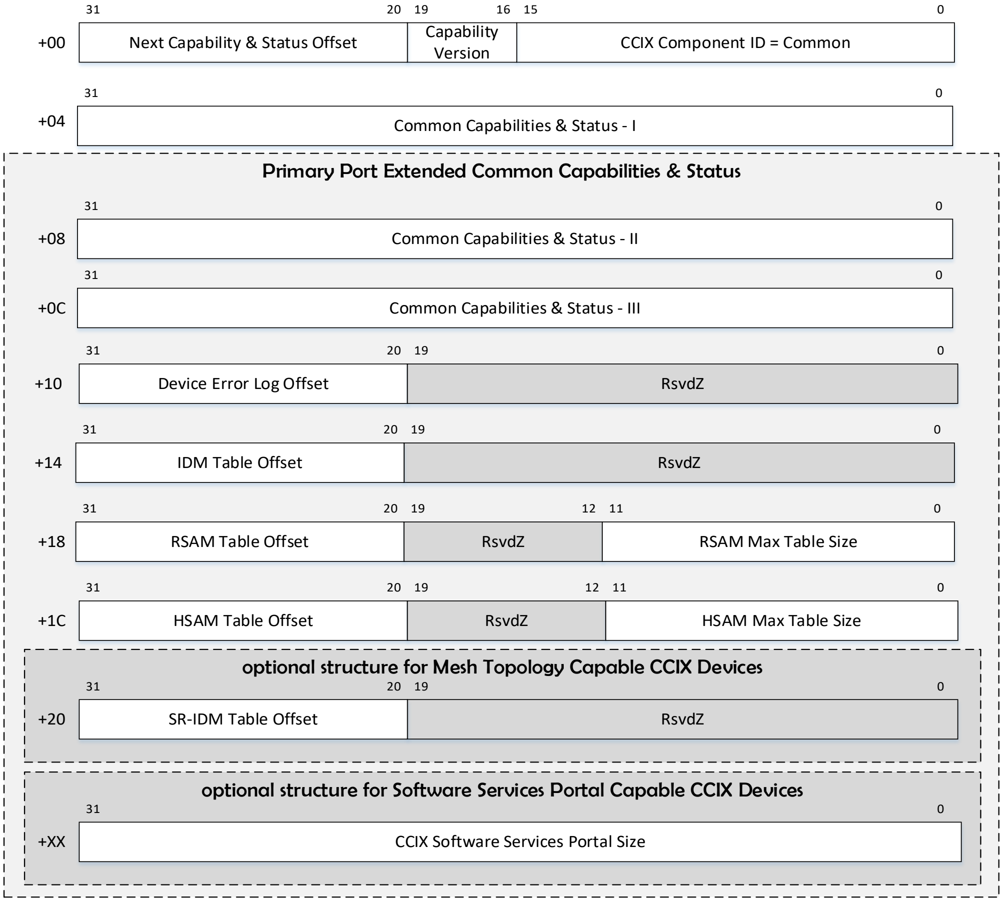

31
20 19
16 15
0
+00
Next Capability & Status Offset
Capability Version
CCIX Component ID = Common
+04
Common Capabilities & Status - I
Primary Port Extended Common Capabilities & Status
+08
Common Capabilities & Status - II
+0C
Common Capabilities & Status - III
+10
Device Error Log Offset
RsvdZ
+14
IDM Table Offset
RsvdZ
+18
RSAM Table Offset
12 11
RsvdZ
RSAM Max Table Size
+1C
HSAM Table Offset
RsvdZ
HSAM Max Table Size
optional structure for Mesh Topology Capable CCIX Devices
+20
SR-IDM Table Offset
RsvdZ
optional structure for Software Services Portal Capable CCIX Devices
+XX
CCIX Software Services Portal Size

Figure 6-14: Common Capabilities & Status structure

The Primary CCIX Port's Extended Common Capabilities & Status Register at Byte Offset-10h contains the CCIX Device Error Log Offset which is described in the CCIX RAS Overview. The remaining bits in this register are Reserved and Zero.

Table 6-9 describes the IDM Pointer Register fields at Byte Offset-14h and Byte Offset-20h shown in Figure 6-14.

Table 6-9: IDMPtr Register fields

Bit Location	Register Description	Attributes
19:0	Reserved and Zero	RsvdZ
31:20	**IDMTblOffset** This IDM Table offset field indicates the byte offset to the start of this IDM Table in the Primary CCIX Port. The offset must be DW-aligned. The IDMTblOffset + fixed 64 DW size must remain within the size of CCIX Protocol Layer DVSEC.	RO

Table 6-10 describes the RSAM Pointer (RSAMPtr) Register at Byte Offset-18h and HSAM Pointer (HSAMPtr) Register at Byte Offset-1Ch shown in Figure 6-14.

Table 6-10: SAMPtr Register /Fields

Bit Location	Register Description	Attributes
11:0	**SAMMaxTblSize** This field indicates the maximum number of bytes in this type of SAM Table. The size must be DW-aligned. • When the CCIX Device has CCIX Port Aggregation Capability, as indicated by the ComnCapStat2.PortAggCap field (see Table 6-13), the indicated SAM Table Size must include two DW of Hash Mask Registers as shown in Figure 6-23. • A SAMMaxTblSize value of 000h indicates the CCIX Device does not have that particular SAM Table Type. For example, a CCIX Device that is not an intermediate point and only has Subordinate Agents may indicate an RSAM SAMMaxTblSize value of 000h. • A CCIX Device, at a minimum, must be capable of being a node in a CCIX tree topology. Therefore, the minimum non-zero SAMMaxTblSize value must reflect (P + 1 + Local) SAM Entries. P is the number of CCIX Ports on the CCIX Device. Local is a minimum value of 1 when the CCIX Device has a Local SAM Destination (see SAMEntryAttr.DestType in Table 6-21) and supports Port-to-Port Forwarding (see PortCapStat1.PortToPortFwdingCap in Figure 6-29). ○ Minimum non-zero SAMMaxTblSize is 018h when P=1, ComnCapStat2.PortAggCap=0, L=0. ○ The minimum, non-zero, (P + 1 + Local) SAM Entry requirement only applies to the worst-case routing requirement in a tree topology, which applies when an intermediate node in the tree must provide routing to two distinct ranges (higher and lower) on the same port. For a particular CCIX tree topology, it is possible for certain nodes in that topology to require all (P + 1 + Local) SAM Entries, while other nodes achieve their necessary SAM Window routing attributes by utilizing less than (P + 1 + Local) SAM Entries. This is further illustrated by an example in Section 6.6.2.1.	RO
19:12	Reserved and Zero	RsvdZ
31:20	**SAMTblOffset** This field indicates the byte offset to the start of this SAM Table Offset in the Primary CCIX Port. The offset must be DW-aligned. The SAMTblOffset + size must remain within the size of CCIX Protocol Layer DVSEC.	RO

The CCIX Software Services Portal Size register is at Byte Offset-20h, i.e., XX in Figure 6-14 is 20h, if the SR-IDM Pointer Register is not part of the Primary Port Extended Capabilities & Status structure. The CCIX Software Services Portal registers is at Byte Offset-24h, i.e., XX in Figure 6-14 is 24h, if the SR-IDM Pointer Register is part of the Primary Port Extended Capabilities & Status structure.

Table 6-11 describes the fields of the CCIX Software Services Portal (SoftwareServicesPortal) register.

Table 6-11: SoftwareServicesPortal Register

Bit Location	Register Description	Attributes
31:0	**PortalSize** This field indicates the encodings for the Software Services Portal Size Supported by this CCIX Device. The PortalSize is encoded as integer multiples of 64KB, starting with the smallest Portal Size of 64KB. 0000h: 64K Portal Size. 0001h: 128K Portal Size. …. FFFFh: 4GB Portal Size.	RO

Figure 6-15 illustrates how the IDM and SAM Pointer Registers, while located in the Capabilities & Status structures, point to data structures located in the Control structures of the Primary CCIX Port located in Function0. The location of IDM and SAM tables within the Control structures are determined by the IDMTblOffset and SAMTblOffset values respectively.

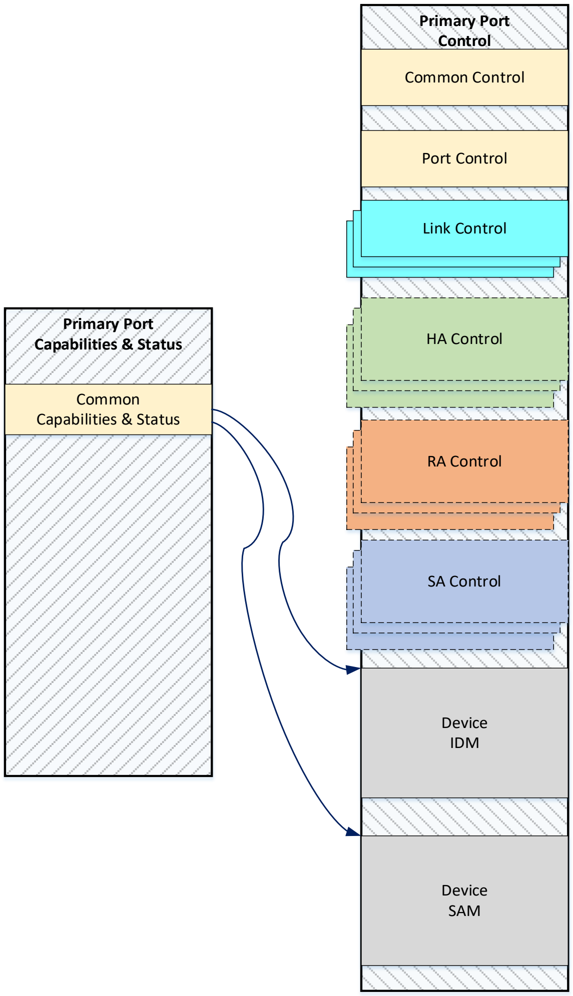


Figure 6-15: CCIX Device SAM/IDM Tables

6.2.2 CCIX Component Structures

This section describes the layout of the Capabilities & Status Registers and Control Registers. Tables describe fields within these registers, the field's position and width, and field names. Field names are the same for fields with the same definition and encodings across different tables.

6.2.2.1 Common Structures

The Common Capabilities & Status Registers and Common Control Registers contain attributes that are common to all CCIX Components on a CCIX Device. An example of a capabilities attribute is the Address Width Capability attribute, which is a capability common to all CCIX Components on that CCIX Device. Similarly, the Address Width Enable attribute is a control applied to all CCIX Components on that CCIX Device.

Status bits in the Common Capabilities & Status Registers communicate cross-component status and follow a hierarchical model where an individual CCIX Component's status only reflects the status of that CCIX Component and the Common status indicates the overall status of the CCIX Device.

Similarly, the Control bits allow for cross-component control and follows a hierarchical model where a Common Enable control bit when cleared, disables all CCIX Components regardless of the CCIX Component's individual Enable control bit being set.

6.2.2.1.1 Common Capabilities & Status Data Structures

Figure 6-14 shows the overall layout of the Common Capabilities & Status structure.

Figure 6-16 shows the layout of the Common Capabilities & Status 1 (ComnCapStat1) Register at Byte Offset-04h.

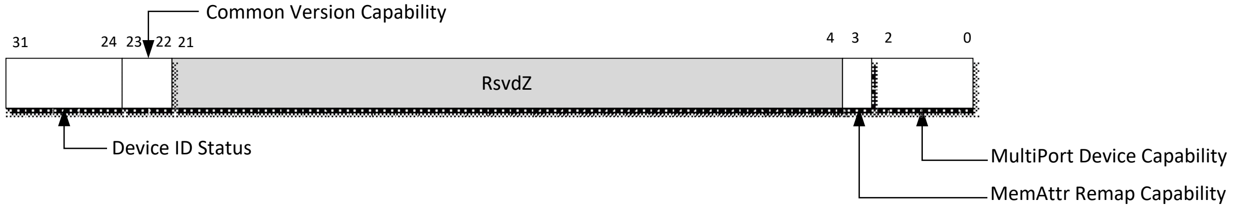


Figure 6-16: ComnCapStat1 Register at Byte Offset-04h

Table 6-12 describes the ComnCapStat1 Register fields at Byte Offset-04h.

Table 6-12: CommCapStat1 Register fields at Byte Offset-04h

Bit Location	Register Description	Attributes
2:0	**MultiPortDevCap** This field describes the CCIX Device's multi-port capability. Bit-0: • 0b: CCIX Device can be accessed by a single CCIX Port. • 1b: CCIX Device can be accessed by at least two CCIX Ports. This attribute is repeated in each CCIX Port's DVSEC Common capability and gives an indication to CCIX Software during the CCIX Device Discovery phase that CCIX DeviceID Replication will be performed by the CCIX Device during the configuration phase. See the DevIDCntl field description in Table 6-14. Bit-1: • 0b: Indicates this is a Secondary CCIX Port. • 1b: Indicates this port has Primary CCIX Port Capability and Function0 contains the chip-level Capabilities & Status and Control structures. Bit-2: • 0b: Indicates this CCIX Device does not support being a Node in a CCIX Mesh Topology. • 1b: Indicates this CCIX Device supports being a Node in a CCIX Mesh Topology. Additional Description: A physical device with only one CCIX capable PCIe Port must indicate an MultiPortDevCap[2:0] value of 010b. A physical device with multiple CCIX capable PCIe Ports must indicate an MultiPortDevCap[2:0] value of x11b on at least one of its ports. MultiPortDevCap[1] value of 1b indicates the existence of Primary CCIX Port Extended Common Capabilities & Status structure. MultiPortDevCap[2:0] value of 111b indicates the existence of the SR-IDM Offset Register in the Primary CCIX Port Extended Common Capabilities & Status structure (see Figure 6-14) and the SR-IDM table, see Section 6.2.2.1.2.3	RO
3	**MemAttrRemapCap** This field describes the CCIX Device's capability to remap the MEM[2:0] encodings received from the PCIe ATS Completion Message to the CCIX Memory Attribute Encodings 0b: CCIX Device does not have the capability to remap MEM[2:0] from the current default mapping described in Table 8-1. 1b: CCIX Device has the capability to remap MEM[2:0] from the current default mapping described in Table 8-1.	RO
21:4	Reserved and Zero	RsvdZ

Bit Location	Register Description	Attributes
23:22	**ComnVersionCap** 00b: Common data structures Version 1 Capability. All other encodings: Reserved.	RO
31:24	**DevIDStat** This field contains the CCIX DeviceID replicated from the ComnCntl1.DevIDCntl field described in Table 6-14. 00h: ComnCntl1.DevIDCntl has not been enumerated or ComnCntl1.DevIDCntl has not been replicated. 01h – FFh: ComnCntl1.DevIDCntl has been replicated. CCIX Device initializes to 00h after reset (except FLR).	RO

Figure 6-17 shows the layout of the Primary CCIX Port's Extended Common Capabilities & Status 2 (ComnCapStat2) Register at Byte Offset-08h.

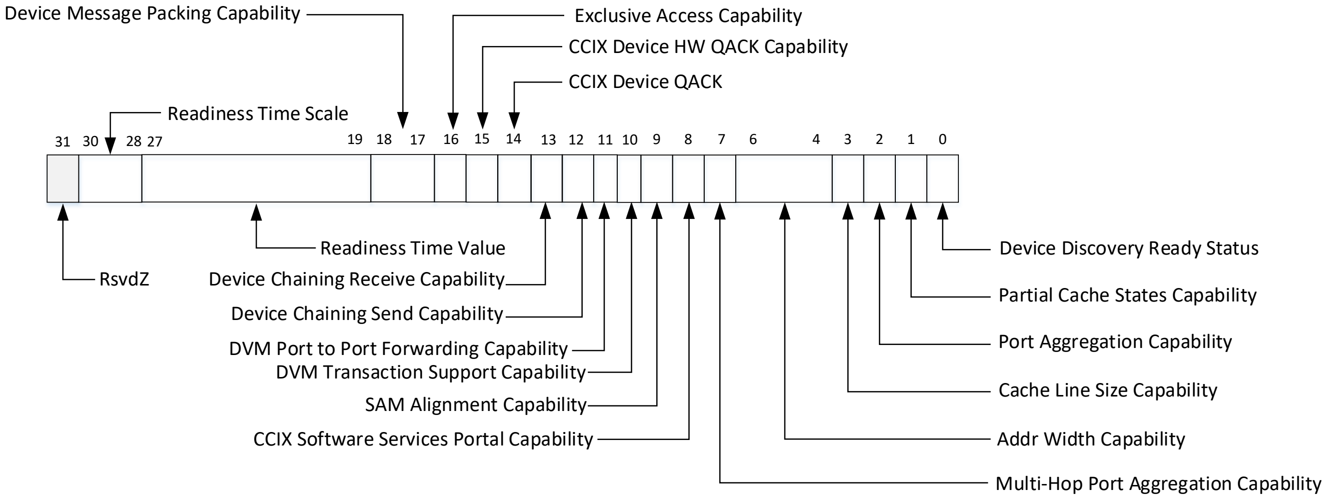

Device Message Packing Capability
Exclusive Access Capability
CCIX Device HW QACK Capability
CCIX Device QACK
Readiness Time Scale
31
30
28
27
19
18
17
16
15
14
13
12
11
10
9
8
7
6
4
3
2
1
0
RsvdZ
Readiness Time Value
Device Chaining Receive Capability
Device Chaining Send Capability
DVM Port to Port Forwarding Capability
DVM Transaction Support Capability
SAM Alignment Capability
CCIX Software Services Portal Capability
Device Discovery Ready Status
Partial Cache States Capability
Port Aggregation Capability
Cache Line Size Capability
Addr Width Capability
Multi-Hop Port Aggregation Capability

Figure 6-17: ComnCapStat2 Register at Byte Offset-08h

Table 6-13 describes the ComnCapStat2 Register fields at Byte Offset-08h.

5

Table 6-13: ComnCapStat2 Register fields at Byte Offset-08h

Bit Location	Register Description	Attributes
0	**DevDiscRdyStat** This field describes the CCIX Device's Discovery Readiness Status. 0b: Indicates the CCIX Device's DVSEC data structures are not ready to be discovered and configured. 1b: Indicates the CCIX Device's DVSEC data structures are ready to be discovered and configured. A CCIX Device must take no longer than <CCIX Device Readiness Time Reported> to set the DevDiscRdyStat field. After <CCIX Device Readiness Time Reported> has elapsed since a reset, software detecting a zero returned for the DevDiscRdyStat field is permitted to conclude that the CCIX Device cannot be configured/enumerated. <CCIX Device Readiness Time Reported> is determined from the DevRdyTimeScale and DevRdyTimeValue fields. Software reading the data structures when the DevDiscRdyStat field is clear, must consider the values in those data structures to be transient capabilities & status, unless stated otherwise in the individual field definitions.	RO
1	**PartialCacheStatesCap** This field describes whether the CCIX Device supports transactions that operate on partial cache lines. 0b: Partial Cache States not supported. 1b: Partial Cache States supported.	RO
2	**PortAggCap** This field describes whether the CCIX Device supports CCIX Port Aggregation. This field must be 0b for a CCIX Device without Multi-Port Capability, i.e. when ComnCapStat1.MultiPortDevCap[0] is 0b (see Table 6-12). A CCIX Device is permitted to support CCIX Port Aggregation on some CCIX Ports but not all CCIX Ports. 0b: No CCIX Ports on this CCIX Device support CCIX Port Aggregation. 1b: At least 2 CCIX Ports on this CCIX Device support CCIX Port Aggregation. The CCIX Ports which are aggregation capable are described by the PortCapStat1.PortAggVctr, described further in Table 6-32.	RO
3	**CachelineSizeCap** This field describes the Cacheline size supported on the CCIX Device. 0b: Indicates only 64B Cacheline size supported. 1b: Indicates both 64B and 128B Cacheline size supported.	RO

Bit Location	Register Description	Attributes
6:4	**AddrWidthCap** This field describes the maximum addressing capability supported by CCIX Components on the CCIX Device. As a result, this field also describes the maximum address decode capabilities of the SAM and/or BAT entries by CCIX Components on the CCIX Device. 000b: 48-bit and lower address width supported. 001b: 52-bit and lower address width supported. 010b: 56-bit and lower address width supported. 011b: 60-bit and lower address width supported. 100b: 64-bit and lower address width supported. All other encodings: Reserved.	RO
7	**MultiHopPortAggCap** This field describes whether the CCIX Ports on the CCIX Device that support CCIX Port-Aggregation also support sending/receiving aggregated traffic from one set of aggregated CCIX Ports to another set of aggregated CCIX Ports on the CCIX Device. This field must be 0b for a CCIX Device without CCIX Port aggregation capability, i.e. when ComnCapStat2.PortAggCap is 0b. 0b: • No CCIX Ports on this CCIX Device support Multi-Hop CCIX Port Aggregation. 1b: • At least 2 groups of CCIX Ports on this CCIX Device support Multi-Hop CCIX Port Aggregation.	RO
8	**SoftwareServicePortalCap** This field indicates whether the CCIX Device's Common Control structure includes a programmable Portal window for CCIX Software Services. 0b: • CCIX Device does not have a Software Services Portal. Software Services must be performed by the CCIX Device driver. 1b: • CCIX Device has a Software Services Portal for CCIX Software Services as an alternative, CCIX standardized mechanism, for software services typically performed by the CCIX Device driver. • This setting also indicates the presence of the Software Services Portal Size register, illustrated in Figure 6-14 and described in Table 6-11.	RO
9	**SAMAlignCap** This field describes the alignment requirement for SAM entries in the CCIX Device's SAM Tables: 0b: • Indicates that SAM entries must be programmed with 4GB aligned Start Address and End Address combinations, i.e. the SAM Window described by the SAM entry is 4GB aligned.	RO

Bit Location	Register Description	Attributes
	1b: • Indicates that SAM entries must be programmed with a Start Address that is 2^n size aligned, and End Address is Start Address + 2^n Size. • If the total size of the SAM window is <4GB, the SAM window is rounded up to 4GB size. • If the total size of the SAM window is not 2^n sized, then the SAM window is rounded up to the next 2^n size, implicitly creating a Memory Hole for address space above the total size. This field also describes the alignment requirement for BAT entries on this CCIX Device for Memory Pools that advertise Base Address Capability (as opposed to Fixed Offset Capability), as described by the MemPoolEntryCapStat0.MemPoolAddrCap field: 0b: • Indicates that BAT entries must be programmed with a 4GB aligned Base Address. 1b: • Indicates that BAT entries must be programmed with a Base Address that is 2^n size aligned to the total size of a Memory Pool or Memory Pool Group. • A Memory Pool Group is the contiguous set of Memory Pools where the first Memory Pool has Base Address Capability and subsequent Memory Pools all have Fixed Offset Capability. • If the total size of a Memory Pool or Memory Pool Group is not 2^n sized, then the alignment of the Base Address is rounded up to the next 2^n size, implicitly creating a Memory Hole for address space above the total size. A 2^n size alignment requirement for SAM and BAT Entries can lead to a sparse Global System Address Map for a particular CCIX topology, to the extent that there is insufficient Physical Address Space to map all the discovered Memory Pools, and also achieve the 2^n size alignment requirement. CCIX Configuration Software may therefore, choose not to enable a CCIX Device that declares a SAMAlignCap value of 1b, if a combination of the Address Width Capability discovered across all the CCIX Devices, the sizes of all the Memory Pools, and the number of CCIX Devices with a 2^n size alignment requirement, results in insufficient Physical Address Space. Implementations are therefore encouraged to declare a SAMAlignCap value of 0b.	
10	**DVMTransactionSupportCap** Defines if there are RAs on this chip which are DVM issuers/receivers. Value 0: The chip does not contain RAs which support DVM transactions. Value 1: The chip contains RAs which support DVM transactions.	RW
11	**DVMForwardingCap** Defines if the chip is capable of broadcasting incoming DVM transactions to egress ports.	RW

Bit Location	Register Description	Attributes
	Value 0: The chip is not capable of broadcasting incoming DVM transactions. Value 1: The chip is capable of broadcasting incoming DVM transactions.	
12	**DevChainSendCap** This field describes whether the HAs or SAs on this CCIX Device that are SA Device Aggregation Capable are also capable of sending Device Chained Message Requests.	RO
13	**DevChainRcvCap** This field describes whether the HAs or SAs on this CCIX Device that are SA Device Aggregation Capable are also capable of receiving Device Chained Message Requests.	RO

Bit Location	Register Description	Attributes
14	**DeviceQACK** This field describes the CCIX Device's Quiesce Acknowledgement status. 0b: CCIX Device not quiesced. 1b: CCIX Device quiesced. If the CCIX Device has Hardware Quiesce Acknowledgement (HW QACK) capability, as indicated by a ComnCapStat2.DeviceHWQACKCap value of 1b, then ComnCapStat2.DeviceQACK is set by implementation specific methods. Software can choose to poll ComnCapStat2.DeviceQACK instead of waiting for the <Device Quiesce Time> value to check DeviceQACK if ComnCapStat2.DeviceHWQACKCap has a value of 1b. If the CCIX Device does not have HW QACK capability as indicated in ComnCapStat2.DeviceHWQACKCap value of 0b, the following sequence is followed: The CCIX Device sets the DeviceQACK bit to a value 1b after completing or detecting the following actions in this order: 1. The CCIX Device detects that the CCIX Device Quiesce Request (ComnCntl2.DeviceQREQ) Control bit is set. 2. The CCIX Device has not issued any Requests and sent and received all relevant outstanding Responses, for the duration of <Device Quiesce Time>. <Device Quiesce Time> is based on the value of ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue, described further in Table 6-12. 3. Following the transition of the CCIX Device Quiesce Request (ComnCntl2.DeviceQREQ in Table 6-12) control bit from 0b to 1b, the CCIX Device must take no longer than 2 * <Device Quiesce Time> to set DeviceQACK. After 2 * <Device Quiesce Time>, CCIX Software detecting a zero returned for the DeviceQACK field indicates an error condition such that the CCIX Device was unable to reach a quiescent state. Following the transition of the ComnCntl2.DeviceQREQ control bit from 1b to 0b, the CCIX Device must transition the DeviceQACK bit from 1b to 0b.	RO

Bit Location	Register Description	Attributes
15	**DeviceHWQACKCap** This field describes the CCIX Device's Hardware Quiesce Acknowledgement Capability. 0b: • CCIX Device does not have a hardware mechanism to achieve a quiescent state across all CCIX Components on that Device. 1b: • CCIX Device has a hardware mechanism to achieve a quiescent state across all CCIX Components on that Device.	RO
16	**ExAccCap** This field describes the Exclusive Access capability of the CCIX Device. 0b: • Indicates Exclusive Accesses are not supported. 1b: • Indicates Exclusive Accesses are supported.	RO
18:17	**MsgPackCap** This field describes the Message Packing capability of the CCIX Device. 00b: • Indicates the CCIX Device only supports the legacy variable packet size message packing capability. 01b: • Indicates the CCIX Device only supports Container Mode. 10b: • Reserved. 11b: Indicates the CCIX Device supports both Container Mode as well as the legacy variable packet size message packing capability.	RO

Bit Location	Register Description	Attributes
27:19	**DevRdyTimeValue** This field describes the CCIX Device's Readiness Time Value Encoding where the overall <CCIX Device Readiness Time Reported> is DevRdyTimeValue *<CCIX Device Readiness Time Multiplier>. <CCIX Device Readiness Time Multiplier> is $32^{DevRdyTimeScale}$ns. 000h – 1FFh: Encodings for DevRdyTimeValue of 0 through 511. A DevRdyTimeValue value of 000h indicates the CCIX Device is always ready to be discovered and configured, i.e. ComnCapStat2.DevDiscRdyStat is always 1b. The <CCIX Device Readiness Time Reported> must be the longer of the following two readiness times: 1. The readiness time following a Conventional Reset and 2. The readiness time following a Function Level Reset. The validity of DevRdyTimeValue is not subject to the ComnCapStat2.DevDiscRdyStat indicator. DevRdyTimeValue is valid at the same time the CCIX Protocol Layer DVSEC Header is valid.	RO
30:28	**DevRdyTimeScale** This field describes the CCIX Device's Readiness Time Scale Encoding in order to generate the <Readiness Time Multiplier> where: <CCIX Device Readiness Time Multiplier> is $32^{DevRdyTimeScale}$ ns. 0h – 7h: Encodings for DevRdyTimeScale of 0 through 7 which allows for <CCIX Device Readiness Time Multiplier> values of 1ns through 34359738368ns. DevRdyTimeScale allows encodings 6h and 7h to accommodate CCIX Devices with PCIe as the Transport Layer, where the <CCIX Device Readiness Time Reported> is larger than the PCIe Readiness Time Reporting optional capability, where a PCIe Device can declare a maximum readiness time of approximately 8.5s. The validity of DevRdyTimeScale is not subject to the ComnCapStat2.DevDiscRdyStat indicator. DevRdyTimeScale is valid at the same time the CCIX Protocol Layer DVSEC Header is valid. During initialization, should a CCIX Device determine that it's capable of a lower readiness time, the CCIX Device is permitted to reduce its <CCIX Device Readiness Time Reported> by either reducing its DevRdyTimeValue, DevRdyTimeScale, or both. However, CCIX Configuration Software may use either the original or reduced <CCIX Device Readiness Time Reported>. A CCIX Device is not permitted to increase its <CCIX Device Readiness Time Reported>.	RO
31	Reserved and Zero	RsvdZ

5

The Primary CCIX Port's Extended Common Capabilities & Status 3 (ComnCapStat3) Register at Byte Offset-0Ch has all bits in the register as Reserved and Zero (RsvdZ).

6.2.2.1.2 Common Control Data Structures

shows the overall layout of the Primary CCIX Port's Common Control data structure. Secondary CCIX Ports do not have a Common Control data structure.

The CCIX Device Error Control & Status (DevErrCntlStat) Register at Byte Offset-0Ch is described in the CCIX RAS Overview. The requirements for, and usage of, the optional Snoop Request Hash Mask Register are described in Section 6.2.2.1.2.1. The requirements for, and usage of, the optional CCIX Software Services Portal structure are described in Section 6.2.2.1.2.2.

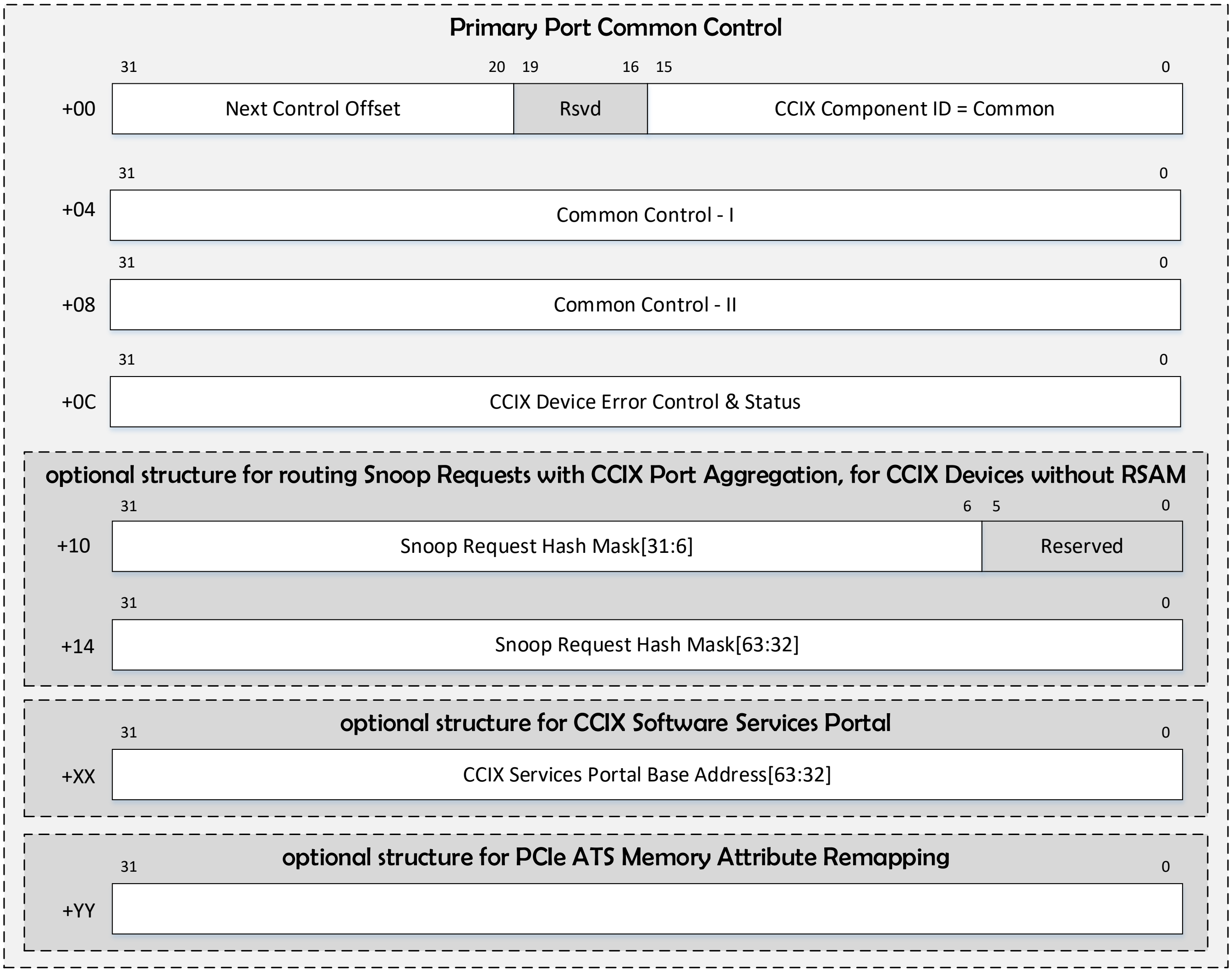

Primary Port Common Control
31
20
19
16
15
0
+00
Next Control Offset
Rsvd
CCIX Component ID = Common
31
0
+04
Common Control - I
31
0
+08
Common Control - II
31
0
+0C
CCIX Device Error Control & Status
optional structure for routing Snoop Requests with CCIX Port Aggregation, for CCIX Devices without RSAM
31
6
5
0
+10
Snoop Request Hash Mask[31:6]
Reserved
31
0
+14
Snoop Request Hash Mask[63:32]
optional structure for CCIX Software Services Portal
31
0
+XX
CCIX Services Portal Base Address[63:32]
optional structure for PCIe ATS Memory Attribute Remapping
31
0
+YY

Figure 6-18: Primary CCIX Port Common Control Structure

Figure 6-19 shows the layout of the Common Control 1 (ComnCntl1) Register at Byte Offset-04h.

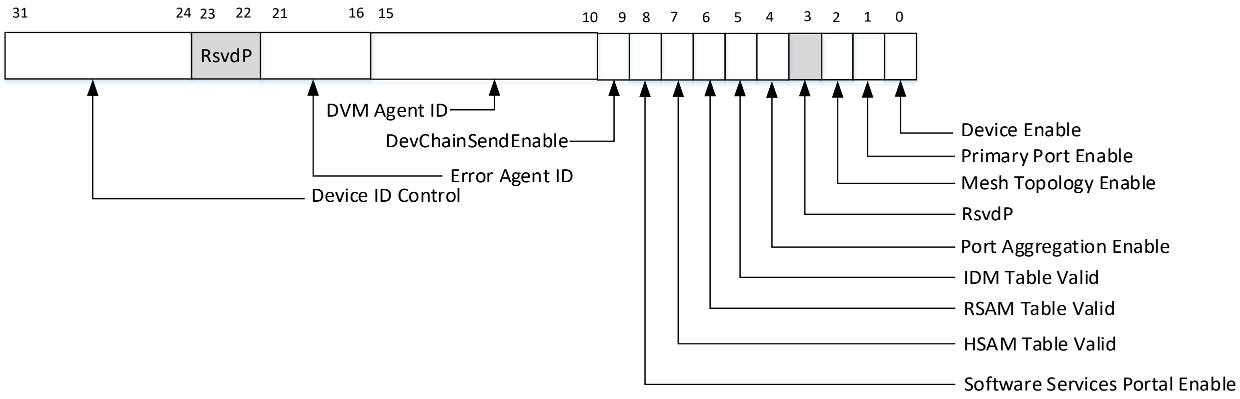


Figure 6-19: ComnCntl1 Register at Byte Offset-04h

Table 6-14 describes the Common Control Register (ComnCntl1) fields at Byte Offset-04h.

Table 6-14: ComnCntl1 Register fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**DevEnable** This field controls whether the CCIX Device is configured to send/receive CCIX traffic, except for CCIX Protocol Error Messages. 0b: Indicates that either the CCIX Device has not been configured, or the previously configured CCIX Device has been taken offline. 1b: Indicates the CCIX Device is configured and enabled. CCIX Device initializes to 0b after reset (except FLR). CCIX Device Enable control takes precedence over CCIX Component Enable control, i.e. CCIX traffic from all enabled CCIX Components on this CCIX Device must be disabled if the CCIX Device is not enabled. DevEnable must be 0b if the ComnCapStat2.DevDiscRdyStat field, described in Table 6-13 is 0b.	RW
1	**PrimaryPortEnable** This field controls enabling the Primary CCIX Port, and the chip-level CCIX Protocol Layer control structures in this Primary CCIX Port. 0b: Primary CCIX Port disabled. 1b: Primary CCIX Port enabled. CCIX Device initializes to 0b after reset (except FLR). PrimaryPortEnable must be set to 0b for a CCIX Device when Common Capability ComnCapStat1.MultiPortDevCap[1] value is 0b (see Table 6-12).	RW

Bit Location	Register Description	Attributes
2	**MeshTopologyEnable** If the CCIX Device is Mesh Topology Capable, i.e. the ComnCapStat1.MultiPortDevCap[2] value is 1b (see Table 6-12), the MeshTopologyEnable field indicates whether this CCIX Device is enabled as part of a Mesh Topology. 0b: CCIX Device is not enabled as part of a Mesh Topology. 1b: CCIX Device is enabled as part of a Mesh Topology. CCIX Device initializes to 0b after reset (except FLR). MeshTopologyEnable must be set to 0b for a CCIX Device when Common Capability ComnCapStat1.MultiPortDevCap[2] value is 0b. A CCIX Device must reference the Common Control SR-IDM Table, instead of the Common Control IDM Table, for the routing of Snoop Responses when MeshTopologyEnable value is 1b. A CCIX Device declaring Common Capability ComnCapStat1.MultiPortDevCap[2] value of 1b can choose to redeploy resources dedicated for Mesh Topology for other purposes when MeshTopologyEnable value is 0b and ComnCntl1.DevEnable value is 1b.	RW
3	Reserved and Preserved	RsvdP
4	**PortAggEnable** This field controls CCIX Device Port Aggregation. PortAggEnable is the single cross-port aggregation enable after individual CCIX Ports on this CCIX Device have been enabled (PortCntl.PortEnable field in Table 6-31). 0b: CCIX Port Aggregation Disabled. 1b: CCIX Port Aggregation Enabled. CCIX Device initializes to 0b after reset (except FLR). PortAggEnable must be 0b if the ComnCapStat2.PortAggCap field is 0b (see Table 6-13).	RW

Bit Location	Register Description	Attributes
5	**IDMTblVal** This field controls the validity of the entire IDM Table. This field also controls the validity of the SR-IDM Table when ComnCapStat1.MultiPortDevCap[2:0] has value of 111b (see Table 6-12). While IDM Table entries have per-entry Valid bits that control mappings on a per-AgentID level, CCIX Software can use a 0b-to-1b transition of IDMTblVal to communicate to the CCIX Device that it is done creating the IDM Table, or IDM and SR-IDM Table, for the CCIX Device. A CCIX Device is permitted to de-assert its ComnCapStat2.DevDiscRdyStat indicator following a 1b-to-0b transition of IDMTblVal. A CCIX Device is permitted to delay assertion of its ComnCapStat2.DevDiscRdyStat indicator following a 0b-to-1b transition of IDMTblVal. For Memory Requests, an implementation may choose to statically select a TgtID from the TgtID pool. The static TgtID selection must be consistent with the updated TgtID pool generated after a 0b-to-1b transition of IDMTblVal.	RW
6	**RSAMTblVal** This field controls the validity of the entire RSAM Table. RSAM Tables must exist in CCIX Devices that contain at least one RA. RSAM Tables must also exist in CCIX Devices that have port-to-port forwarding capability in order to resolve whether packets are destined to a local HA or an HA on another CCIX Device connected via an egress CCIX Port. While RSAM Table entries have Valid bits that control address mappings on a per-entry level, CCIX Software can use a 0b-to-1b transition of RSAMTblVal to communicate to the CCIX Device that it is done creating the RSAM Table for the CCIX Device. A CCIX Device is permitted to de-assert its ComnCapStat2.DevDiscRdyStat indicator following a 1b-to-0b transition of RSAMTblVal. A CCIX Device is permitted to delay assertion of its ComnCapStat2.DevDiscRdyStat indicator following a 0b-to-1b transition of RSAMTblVal.	RW

Bit Location	Register Description	Attributes
7	**HSAMTblVal** This field controls the validity of the entire HSAM Table. HSAM Tables must exist in CCIX Devices that contain at least one HA with Memory Expansion Enabled. HSAM Tables must also exist in CCIX Devices that have port-to-port forwarding capability in order to resolve whether packets are destined to a local SA or an SA on another CCIX Device connected via an egress CCIX Port. While HSAM Table entries have Valid bits that control address mappings on a per-entry level, CCIX Software can use a 0b-to-1b transition of HSAMTblVal to communicate to the CCIX Device that it is done creating the HSAM Table for the CCIX Device. A CCIX Device is permitted to de-assert its ComnCapStat2.DevDiscRdyStat indicator following a 1b-to-0b transition of HSAMTblVal. A CCIX Device is permitted to delay assertion of its ComnCapStat2.DevDiscRdyStat indicator following a 0b-to-1b transition of HSAMTblVal.	RW
8	**SoftwareServicesPortalEnable** If the CCIX Device has Software Services Portal Capability, i.e. the ComnCapStat2.SoftwareServicePortalCap value is 1b (see Table 6-13), the SoftwareServicesPortalEnable field indicates whether the CCIX Software Services Portal is enabled. 0b: The CCIX Software Services Portal is disabled. 1b: The CCIX Software Services Portal is enabled. CCIX Device initializes to 0b after reset (except FLR). SoftwareServicesPortalEnable must be set to 0b for a CCIX Device when the ComnCapStat2.SoftwareServicePortalCap value is 0b.	RW
9	**DevChainSendEnable** This field indicates whether the HAs or SAs on this CCIX Device that are SA Device Aggregation Capable are enabled to send Device Chained Message Requests.	RW
15:10	**DVMAgentID** A six bit field for holding on-chip DVM Agent ID. The field value is applicable only when DVMTransactionSupportCap is set to one, else the field value is inapplicable and must be Reserved and set to zero.	RW

Bit Location	Register Description	Attributes
21:16	**ErrAgentID** This field designates the AgentID of an Error Agent (EA) that resides in the CCIX Host and receives CCIX Protocol Error (PER) Messages. An EA resides in the host. It is mandatory for the Host to have at least one EA. For the list of CCIX Agents reachable on the host, a minimum one EA must also be reachable. It is recommended that CCIX Configuration Software re-use an AgentID for an EA so that ID routed PER Messages follow established routes for those from one of the Host's enumerated RAIDs/HAIDs/SAIDs. The Host-provided information with respect to the RAs/SAs/HAs/EAs present in the host, may also provide proximity information of an EA to the RA/HA/SA Agents. Proximity information may be communicated in the same manner as proximity information between a Hosts's RA and HA are communicated. For the case where the Host has declared multiple EAs, CCIX Configuration Software may then choose to use the Host's CCIX Agent proximity to this CCIX Device in selecting the ErrAgentID for this CCIX Device. Unlike the discovery of the CCIX connection graph, and the subsequent creation of the CCIX topology, CCIX Configuration Software is not required to include EAs in the discovery and topology creation process. For IDM Tables created for all enumerated HAID/SAID/RAID, where an EA shares one of those CCIX AgentIDs, CCIX Configuration Software does not need to comprehend IDM routing explicitly for EAs.	RW
23:22	Reserved and Preserved	RsvdP
31:24	**DevIDCntl** This field controls the enumerated CCIX DeviceID programmed by CCIX Software. CCIX software must ensure the CCIX DeviceID programmed is unique across the CCIX system. 00h: CCIX Device has not been enumerated. 01h: FFh: Encodings for enumerating CCIX DeviceID 1 through 255. A non-zero DevIDCntl value written by Software to the Primary CCIX Port's Common control structure must be replicated by the CCIX Device to the ComnCapStat1.DevIDStat field in all CCIX Common status structures on the same CCIX Device. The same non-zero DevIDCntl value must be returned when Software reads the ComnCapStat1.DevIDStat field of any of the CCIX Common status structures on the CCIX Device.	RW

Figure 6-20 shows the layout of the Common Control 2 (ComnCntl2) Register at Byte Offset-08h.

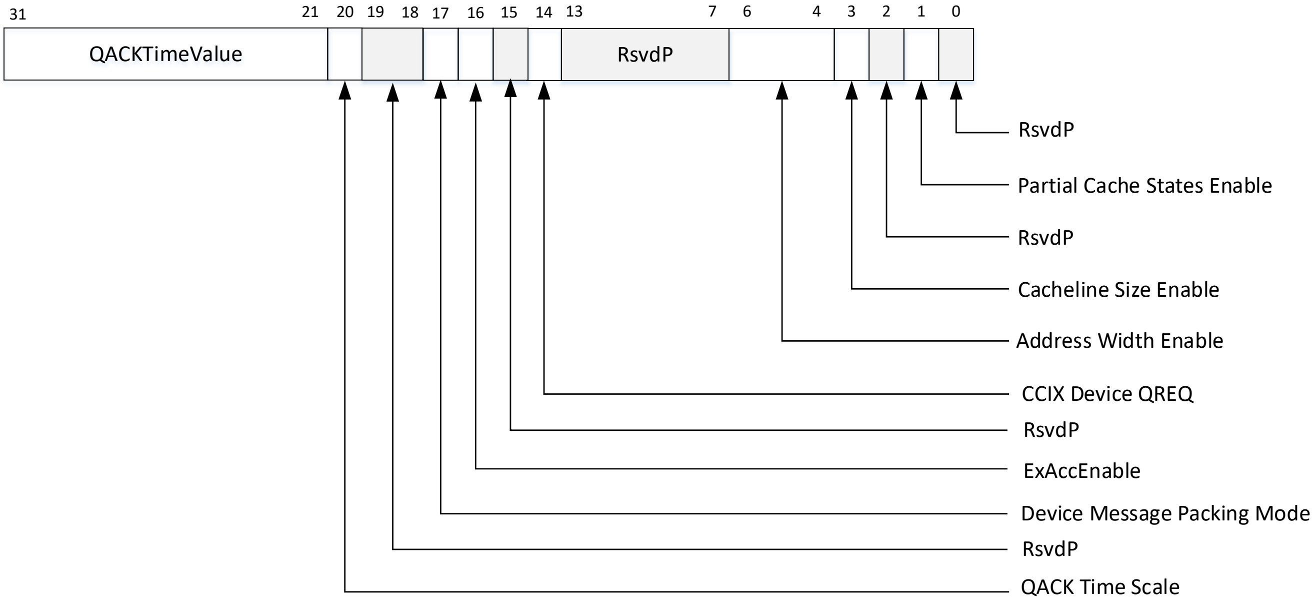


Figure 6-20: ComnCntl2 Register at Byte Offset-08h

Table 6-15 describes the ComnCntl2 Register fields at Byte Offset-08h.

Table 6-15: ComnCntl2 Register fields at Byte Offset-08h

Bit Location	Register Description	Attributes
0	Reserved and Preserved	RsvdP
1	**PartialCacheStatesEnable** This field enables the CCIX Device to handle CCIX transactions that operate on partial cachelines. 0b: Disable. 1b: Enable. CCIX Device initializes to 0b after reset (except FLR).	RW
2	Reserved and Preserved	RsvdP
3	**CachelineSizeEnable** This field controls the Cacheline size selected on the CCIX Device. 0b: Indicates only 64B Cacheline size selected. 1b: Indicates only 128B Cacheline size selected. CachelineSizeEnable must be 0b when ComnCapStat2.CachelineSizeCap is 0b. CCIX Device initializes to 0b after reset (except FLR).	RW

Bit Location	Register Description	Attributes
6:4	**AddrWidthEnable** 000b: Max 48-bit address width enabled. 001b: Max 52-bit address width enabled. 010b: Max 56-bit address width enabled. 011b: Max 60-bit address width enabled. 100b: Max 64-bit address width enabled. All other encodings: Reserved. CCIX Device initializes to 000b after reset (except FLR).	RW
13:7	Reserved and Preserved	RsvdP
14	**DeviceQREQ** Device Quiesce Request controls when the CCIX Device will act to achieve a quiesced state. There can only be a change in the value, 0b or 1b, of the corresponding ComnCapStat2.DeviceQACK bit after CCIX configuration software changes the value, 0b or 1b, of this DeviceQREQ control bit. 0b: CCIX Device is not required to be quiesced. 1b: CCIX Device must be quiesced. CCIX Device initializes to 0b after reset (except FLR).	RW
15	Reserved and Preserved	RsvdP
16	**ExAccEnable** This field enables the use of Exclusive Accesses on the CCIX Device. 0b: Indicates use of Exclusive Accesses is not enabled. 1b: Indicates use of Exclusive Accesses is enabled. Default value is 0b.	RW
17	**MsgPackMode** This field controls the Message Packing Mode of the CCIX Device. 0b: • Indicates the CCIX Device is enabled for legacy variable packet size message packing. This MsgPackMode can only be selected if the ComnCapStat2.MsgPackCap field has values of 00b or 11b. 1b: Indicates the CCIX Device is enabled for Container Mode. This MsgPackMode can only be selected if the ComnCapStat2.MsgPackCap field has values of 01b or 11b. When set, the LinkAttrCntl.MsgPackingEnable and LinkAttrCntl.MaxPktSize fields don’t have any effect.	RW
19:18	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
20	**QACKTimeScale** This field controls the timescale of the Quiesce Acknowledgement Time Value, i.e. QACKTimeValue, and the combination of the settings in QACKTimeScale and QACKTimeValue determine the Queisce Time of a CCIX Component. 0b: Timescale is in µs. 1b: Timescale is in ms. CCIX Device initializes to 0b after reset (except FLR).	RW
31:21	**QACKTimeValue** This field controls the value of the Queisce Time for a CCIX Component. <Component Queisce Time> for a CCIX Component is based on ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue when that Component's Quiesce Request bit transitions from 0b to 1b. For example, <Device Quiesce Time> is based on ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue when DeviceQREQ transitions from 0b to 1b. 000h: reserved. 001h: 1 µs if QACKTimeScale is 0b, 1ms if QACKTimeScale is 1b. 002h: 2 µs if QACKTimeScale is 0b, 2ms if QACKTimeScale is 1b. … 3E8h: 1 ms if QACKTimeScale is 0b, 1s if QACKTimeScale is 1b. All values above 3E8h: Reserved. CCIX Device initializes to 001h after reset (except FLR).	RW

6.2.2.1.2.1 Snoop Request Hash Mask

The Snoop Request Hash Mask data structure must be provided by:

- CCIX Devices that are not intermediate CCIX Components (i.e. CCIX Components that are not CCIX Port-to-Port Forwarding capable) and support CCIX Port Aggregation, and contain Home Agents but no Request Agents
- CCIX Devices that are intermediate CCIX Components (i.e. CCIX Components are CCIX Port-to-Port Forwarding capable) and support CCIX Port Aggregation, but do not contain any Home Agents or Request Agents.

The use of the Snoop Request Hash Mask is described in Section 6.2.2.2.2. The Snoop Request Hash Mask is an optional data structure and need not be provided by CCIX Devices that support CCIX Port Aggregation and contain the RSAM Table. This is because the RSAM data structure contains a Hash Mask that can be used for Snoop Request routing as well.

Table 6-16 describes the Common Control Snoop Request Hash Mask 0 (SnpReqHashMask0) Register field at Byte Offset-10h.

Table 6-16: SnpReqHashMask0 Register field at Byte Offset-10h

Bit Location	Register Description	Attributes
5:0	Reserved and Preserved	RsvdP
31:6	**SnpReqHashMaskLo** This field indicates which lower address bits are part of the Snoop Request Hash Mask function for the aggregated CCIX Ports. This field is only valid when the IDMEntry.NumAggPorts field in Table 6-20 is a valid non-zero value. For each bit: 0b: indicates the address bit is not part of the hash mask result. 1b: indicates the address bit is part of the hash mask result SnpReqHashMaskLo[0] must be 0b if the ComnCntl2.CachelineSizeEnable field indicates 128B Cacheline size is enabled.	RW

Table 6-17 describes the Common Control Snoop Request Hash Mask 1 (SnpReqHashMask1) Register field at Byte Offset-14h.

Table 6-17: Common Control Register field at Byte Offset-14h

Bit Location	Register Description	Attributes
31:0	**SnpReqHashMaskHi** This field indicates which upper address bits are part of the Snoop Request Hash Mask function for the aggregated CCIX Ports. This field is only valid when the IDMEntry.NumAggPorts field in Table 6-20 is a valid non-zero value. For each bit: 0b: indicates the address bit is not part of the hash mask result. 1b: indicates the address bit is part of the hash mask result. The Snoop Request Hash Mask, i.e. {SnpReqHashMaskHi[31:0], SnpReqHashMaskLo[31:6]} must match the RSAM Hash Mask, i.e. {RSAMHashMaskHi[31:0], RSAMHashMaskLo[31:6]} of the CCIX Device with the Request Agent that is the TgtID of the Snoop Request. See Section 6.2.2.3 for a description of the RSAM data structure and RSAM Hash Mask.	RW

6.2.2.1.2.2 CCIX Software Services Portal

The CCIX Software Services Portal allows CCIX configuration software to define a 4GB SAM Window through which CCIX Software Services descriptors can be provided to the service processor on the CCIX device. The services portal is a CCIX standardized method to invoke a service or function, such as memory zeroing, and provides an alternative to CCIX configuration software invoking a proprietary device driver to achieve the same service or function.

The CCIX Software Services Portal register is at Byte Offset-10h, i.e., XX in Figure 6-18 is 10h, if the Snoop Request Hash Mask Structure is not part of the Primary Port Control structure. The CCIX Software Services Portal register is at Byte Offset-18h, i.e., XX in Figure 6-18 is 18h, if the Snoop Request Hash Mask Structure is part of the Primary Port Control structure.

Table 6-18 describes the CCIX Software Services Portal register fields at Byte Offset-XXh of the Primary Port Control structure.

Table 6-18: Primary Port Control structure fields at Byte Offset-XXh

Bit Location	Register Description	Attributes
31:0	**SoftwareServicesPortalBaseAddr** Indicates the upper 32-bits of the 4GB aligned Base Address of the CCIX Software Services Portal.	RW

6.2.2.1.2.3 PCIe Memory Attribute Mapping Structure

Table 6 19 describes the PCIe Memory Attribute Mapping register fields at Byte Offset-YYh of the Primary Port Control structure.

Table 6-19: Primary Port Control structure fields at Byte Offset-YYh

Bit Location	Register Description	Attributes
2:0	**PCIe_MemAttr_000_Mapping** MEM[2:0] = 000 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 000b. This retains the encoding in Table 8-1.	RW
5:3	**PCIe_MemAttr_001_Mapping** MEM[2:0] = 001 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 001b. This retains the encoding in Table 8-1.	RW
8:6	**PCIe_MemAttr_010_Mapping** MEM[2:0] = 010 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 010b. This retains the encoding in Table 8-1.	RW
11:9	**PCIe_MemAttr_011_Mapping** MEM[2:0] = 011 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 011b. This retains the encoding in Table 8-1.	RW
14:12	**PCIe_MemAttr_100_Mapping** MEM[2:0] = 100 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 100b. This retains the encoding in Table 8-1.	RW
17:15	**PCIe_MemAttr_101_Mapping** MEM[2:0] = 101 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 101b. This retains the encoding in Table 8-1.	RW

Bit Location	Register Description	Attributes
20:18	**PCIe_MemAttr_110_Mapping** MEM[2:0] = 110 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 110b. This retains reserved encoding in Table 8-1.	RW
23:21	**PCIe_MemAttr_111_Mapping** MEM[2:0] = 111 is remapped to the CCIX Memory Attribute 3-bit encoding programmed in this field and described in Table 8-1. Default value of this field is 111b. This retains reserved encoding in Table 8-1.	RW
31:24	Reserved and Preserved	RsvdP

The PCIe Memory Attribute Mapping register is at Byte Offset-10h, i.e., YY in Figure 6-18 is 10h, if the Snoop Request Hash Mask Structure and the CCIX Software Services Portal register are both not part of the Primary Port Control structure.

The PCIe Memory Attribute Mapping register is at Byte Offset-18h, i.e., YY in Figure 6-18 is 18h, if only one of either the Snoop Request Hash Mask Structure or the CCIX Software Services Portal register are part of the Primary Port Control structure.

The PCIe Memory Attribute Mapping register is at Byte Offset-20h, i.e., YY in Figure 6-18 is 20h, if both the Snoop Request Hash Mask Structure and the CCIX Software Services Portal register are part of the Primary Port Control structure.

6.2.2.2 IDM Table Structure

The IDM Table or ID Map Table is used to resolve the destination CCIX PortID and CCIX Link Number for that CCIX PortID for a given CCIX AgentID. The IDM table is referenced by CCIX Ports and CCIX Agents. Figure 6-21 shows the overall layout of the IDM entries of the IDM Table. With a maximum of 64 CCIX AgentIDs allowed in a CCIX topology, the number of IDM entries is 64.

The SR-IDM Table is an auxiliary IDM Table structure that is referenced by a CCIX device in specific topologies for specific CCIX Packet types (see Section 6.2.2.2.4). The SR-IDM Table might or might not contain the same routing attributes as the IDM Table for a particular CCIX AgentID; however, the structure of the SR-IDM Table, and manner in which the SR-IDM Table is referenced, is identical to that of the IDM table.

Each CCIX AgentID entry has a valid bit to indicate whether that entry is enabled. Software is allowed to set the Valid bit for non-contiguous CCIX AgentID entries.

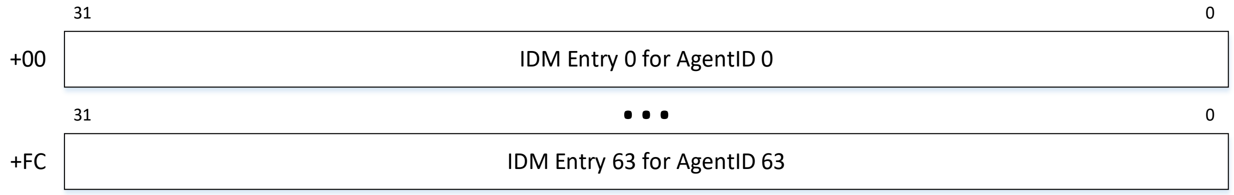


Figure 6-21: IDM Table

Figure 6-22 shows the layout of an IDM entry.

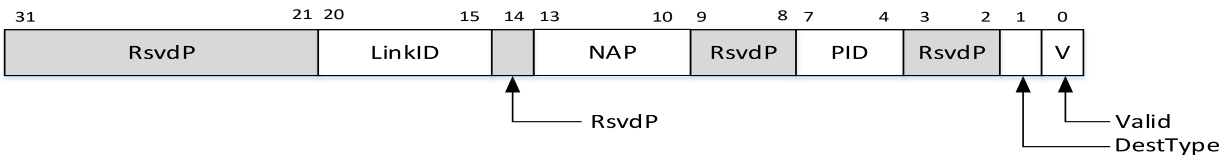


Figure 6-22: IDM Entry

Table 6-20 describes the IDM entry fields.

Table 6-20: IDM Entry

Bit Location	Register Description	Attributes
0	**AgentIDnVal** This field indicates whether the mapping for CCIX AgentIDn is enabled and valid. 0b: Indicates either an invalid CCIX AgentID or disabled mapping. 1b: CCIX AgentID to Local or CCIX PortID mapping is enabled. CCIX Device initializes to 0b after reset (except FLR).	RW
1	**AgentIDnDestType** This field indicates whether the mapping for CCIX AgentIDn is to a Local or CCIX Port Destination. 0b: Local CCIX Agent Destination. 1b: CCIX Port Destination. CCIX Device initializes to 0b after reset (except FLR).	RW
3:2	Reserved and Preserved	RsvdP
7:4	**AgentIDnPortID** This field describes the CCIX PortID that CCIX AgentIDn is mapped to for Entry N of the IDM structure. 0h to Fh: Encodings for CCIX PortID0 to CCIX PortID15. If CCIX Port Aggregation is enabled, the AgentIDnPortID field has encodings for Base CCIX PortID with the following encoding restrictions: 2 CCIX Port Aggregation: Encodings 0h to Eh for Base CCIX PortID0 to Base CCIX PortID14. Encoding Fh: Reserved. 4 CCIX Port Aggregation: Encodings 0h to Ch for Base CCIX PortID0 to Base CCIX PortID12. Encodings Dh to Fh: Reserved. 8 CCIX Port Aggregation: Encodings 0h to 8h for Base CCIX PortID0 to Base CCIX PortID8. Encodings 9h to Fh: Reserved. 16 CCIX Port Aggregation: Encoding 0h for Base CCIX PortID0. Encodings 1h to Fh: Reserved. CCIX Device initializes to 00h after reset (except FLR). AgentIDnPortID must be 0h when IDMEntry.AgentIDnDestType is 0b.	RW
9:8	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
13:10	**NumAggPorts** This field indicates whether CCIX Port Aggregation is enabled for this CCIX AgentID and if enabled, the number of aggregated CCIX Ports. The encoding is such that the number of aggregated CCIX Ports is (NumAggPorts + 1) when the NumAggPorts field contains a valid, non-zero value. However, the total number of aggregated CCIX Ports is restricted to 2^n where valid values for n are 0, 1, 2, 3, and 4. 0h: • Indicates Aggregation is disabled, i.e. $2^n = 1$. • The AgentIDnPortID field in this case is the singular CCIX Port for this CCIX AgentID. 1h, 3h, 7h, Fh: • Indicates the number of additional CCIX Ports aggregated. • The maximum number of additional CCIX Ports that can be expressed is 15, i.e. a total of 16 CCIX Ports can be aggregated. • The AgentIDnPortID field in this case is Base CCIX PortID (BasePortID). The subsequent CCIX Ports must be linearly enumerated from this BasePortID. BasePortID is not required to be 2^n aligned. All other encodings: Reserved. CCIX Device initializes to 0h after reset (except FLR). NumAggPorts must be 0h when IDMEntry.AgentIDnDestType is 0b.	RW
14	Reserved and Preserved	RsvdP
20:15	**AgentIDnLinkID** This field describes the CCIX LinkID that CCIX AgentIDn is mapped to for Entry N of the IDM structure. 00h to 3Fh: Encodings for CCIX LinkID0 to CCIX LinkID63. If CCIX Port Aggregation is enabled, the same CCIX LinkID must be setup on all aggregated CCIX Ports for traffic for this CCIX AgentID. CCIX Device initializes to 00h after reset (except FLR). AgentIDnLinkID must be 00h when IDMEntry.AgentIDnDestType is 0b.	RW
31:21	Reserved and Preserved	RsvdP

6.2.2.2.1 IDM Table Usage by CCIX Components

The IDM Table structure is referenced by different CCIX Components:

- CCIX Port:
    - IDM structure is referenced in order to resolve either the next-hop destination CCIX Port/Link or the last-hop local destination for inbound CCIX AgentID routed CCIX Packets. How the CCIX Packet is routed to the last-hop local destination is implementation specific.
- CCIX Agents (HA/RA/SA):
    - IDM structure is referenced in order to resolve the destination CCIX Port/Link for outbound CCIX AgentID routed CCIX Packets.
    - The CCIX AgentID used to index into the IDM can only belong to destination CCIX Agents. The destination CCIX Agents cannot be both HA and SA for the same index, i.e. separate CCIX AgentIDs, and therefore separate IDM entries, with unique CCIX Port/Link must be used instead.
    - The IDM structure may be referenced in order to resolve to a destination Local CCIX Agent. However, the routing from source CCIX Agents to destination CCIX Agents on the same CCIX Device is implementation specific, i.e. CCIX Configuration Software disabling CCIX AgentID mapping in the IDM structure does not guarantee that access has been disabled from source CCIX Agents to destination CCIX Agents on the same CCIX Device.
    - HA initiated Snoop Request ID routed packets to aggregated CCIX Ports have additional IDM Table usage considerations, described in Section 6.2.2.2.3.
    - RA initiated Snoop Response ID routed packets have additional IDM Table usage considerations for nodes in Mesh Topologies, described in Section 6.2.2.3.4.

6.2.2.2.2 Routing of Responses for CCIX AgentIDs with CCIX Port Aggregation

For the case where a destination CCIX AgentID has aggregation enabled, the source CCIX Agent requires implementation-specific mechanisms other than the IDM Table to resolve the destination CCIX PortID for the routing of Response packets. This is because the IDM Table will only resolve the destination CCIX AgentID to the base destination CCIX PortID.

- **Example:** An RSAM entry has CCIX Port aggregation enabled to two CCIX Ports, CCIX Port1 and CCIX Port2. When inbound CCIX Snoop Requests from a Home Agent are serviced by that Request Agent, the Snoop Responses must have alternative mechanisms to send the Snoop Response back to the CCIX Port#, CCIX Port1 or CCIX Port2, from which the Snoop Request was sent since the IDM Table will only indicate the Base CCIX Port Number, CCIX Port1, of the Aggregated CCIX Port for that Home AgentID.

6.2.2.2.3 Routing of Snoop Requests for CCIX Devices with CCIX Port Aggregation

Snoop Requests are ID-routed CCIX packets. However, for the case of CCIX Port aggregated TgtIDs of Snoop Requests, the IDM Table only indicates the Base CCIX PortID (BasePortID) of the aggregated CCIX Ports for routing the Snoop Request. The Snoop Address and Hash Mask are used to determine the destination CCIX PortID via the Aggregated Port Selection Function (APSF, as described in Section 6.2.2.3.3).

A CCIX Device with at least one Request Agent must reference the RSAM Table Hash Mask (see Figure 6-23) for determining the CCIX PortID for Snoop Requests.

A CCIX Device with no Request Agents must reference the Snoop Request Hash Mask (see Figure 6-18) for determining the CCIX PortID for Snoop Requests.

6.2.2.2.4 Routing of Snoop Responses for CCIX Devices that support Mesh Topologies

CCIX Devices enabled in a Mesh Topology (see MeshTopologyEnable field in Table 6-14) must reference the SR-IDM Table for determining the CCIX PortID for Snoop Responses. A description of this requirement is described in Chapter 3, Protocol Layer.

6.2.2.3 SAM Table Structure

The SAM Table, or System Address Map Table, is used to resolve the destination CCIX Component (CCIX PortID) for a given Address. The SAM Table is referenced by CCIX Ports and CCIX Agents for address routed packets. Figure 6-23 shows the overall layout of the SAM Table. The SAM Table structure contains one or more entries. The SAM Table Size along with the SAM Table Type supported by the CCIX Component indicates the number of SAM entries available for setup and the index into each SAM entry.

Each SAM entry has a valid bit to indicate whether that entry is enabled. Software is allowed to set the Valid bit for non-contiguous SAM entries.

The address range mapped by a particular valid SAM entry must not overlap with an address range mapped in any other valid SAM entry. However, when the HA and SA Address Space are independent of each other, overlap of Address name-space between HBAT/RSAM and SBAT/HSAM Tables is allowed.

When the CCIX Device has CCIX Port Aggregation Capability, as indicated by the ComnCapStat2.PortAggCap field (see Table 6-13), the SAM Table structure is extended by two DW for the Hash Mask Register as shown in Figure 6-23.

The Hash Mask applies to all SAM Entries where the SAMEntryAttr.NumAggPorts field indicates CCIX Port Aggregation (see Table 6-21).

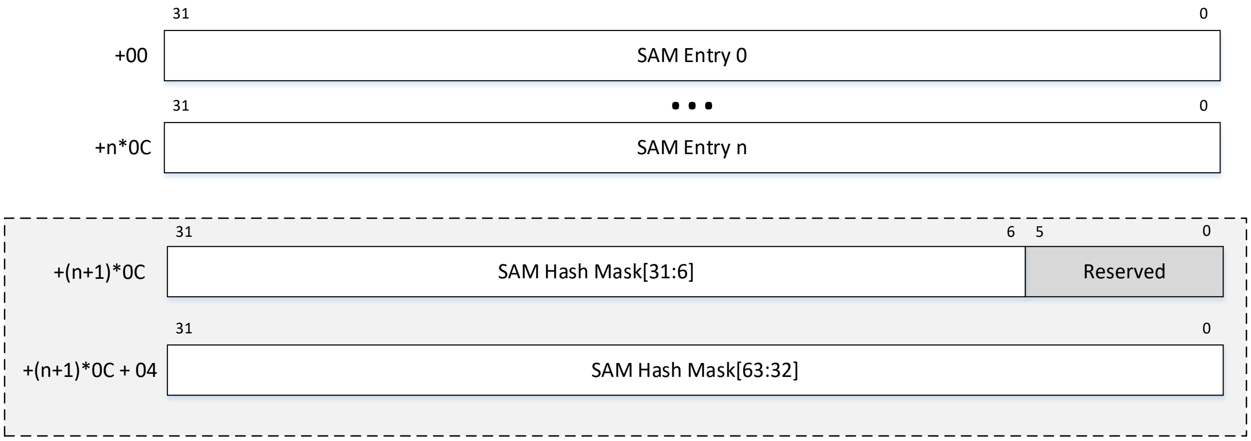


Figure 6-23: SAM Table

6.2.2.3.1 Common SAM Entry

Figure 6-24 shows the layout of a SAM Entry.

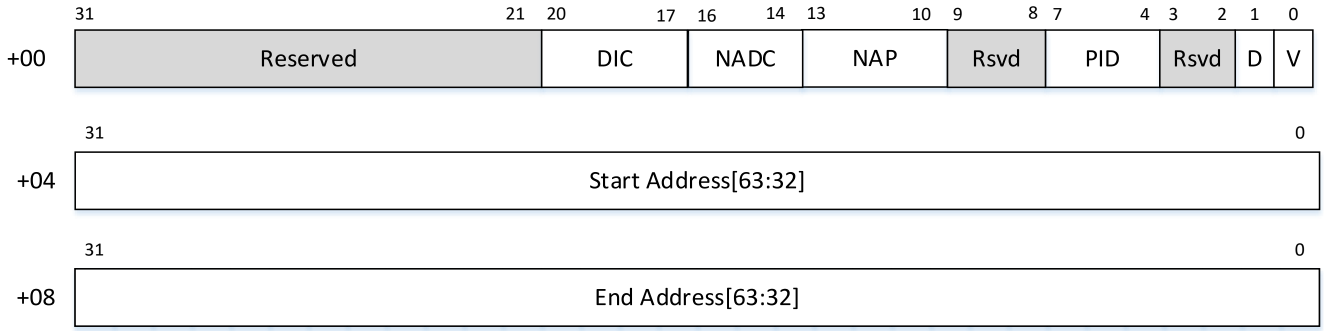


Figure 6-24: SAM Entry

Table 6-21 describes the SAM Entry Attribute (SAMEntryAttr) Register fields at Byte Offset-00h of the SAM Entry.

Table 6-21: SAMEntryAttr Register fields at Byte Offset-00h

Bit Location	Register Description	Attributes
0	**SAMEntryVal** This field indicates whether the SAM Entry is valid. 0b: Indicates no CCIX Port is mapped in this entry. 1b: Indicates a CCIX Port is mapped in this entry.	RW
1	**DestType** Describes whether a CCIX Agent or CCIX Port is mapped to the Base Address described in this SAM entry. 0b: Local CCIX Agent Destination. 1b: CCIX Port Destination.	RW
3:2	Reserved and Preserved	RsvdP
7:4	**PortID** Indicates the CCIX PortID (when CCIX Port is indicated by the SAMEntryAttr.DestType field) mapped to the address range described in this SAM entry. For aggregated CCIX Ports, i.e. when the SAMEntryAttr.NumAggPorts field contains a valid, non-zero value, PortID indicates the Base CCIX PortID (BasePortID). PortID must be 0h when SAMEntryAttr.DestType is 0b.	RW
9:8	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
13:10	**NumAggPorts** This field indicates the number of aggregated CCIX Ports when a CCIX Port destination is indicated in the SAMEntryAttr.DestType field. The encoding is such that the number of aggregated CCIX Ports is (NumAggPorts + 1) when the NumAggPorts field contains a valid, non-zero value. However, the total number of CCIX Aggregated Ports is restricted to 2^n where valid values for n are 0, 1, 2, 3, and 4. 0h: • Indicates Aggregation is disabled, i.e. $2^n = 1$. • The SAMEntryAttr.PortID field in this case is the singular CCIX Port for that SAM entry. 1h, 3h, 7h, Fh: • Indicates the number of additional CCIX Ports aggregated. • The maximum number of additional CCIX Ports that can be expressed is 15, i.e. a total of 16 CCIX Ports can be aggregated. • The SAMEntryAttr.PortID field in this case is Base CCIX PortID (BasePortID). The subsequent CCIX Ports must be linearly enumerated from this BasePortID. BasePortID is not required to be 2^n aligned. All other encodings: Reserved. A SAM Entry must not have both Port Aggregation and SA Device Aggregation enabled, i.e. NumAggPorts and NumAggDevCntl must not both be set to non-zero values for the same SAM Entry. CCIX Device initializes to 0h after reset (except FLR). NumAggPorts must be 0h when SAMEntryAttr.DestType is 0b. Address regions with CCIX Device Memory Type attribute must not have an aggregated CCIX Port type, i.e. NumAggPorts must be 0h for a SAM Entry that describes CCIX Device Memory Type SAM Window destinations. CCIX Port Aggregation is only permitted for address regions with Normal Memory Type attribute.	RW

Bit Location	Register Description	Attributes
16:14	**NumAggDevCntl** This field controls the number of CCIX Devices the SAM entry is aggregated across. This field only applies to HSAM Entries. The number of aggregated devices is $2^{\text{NumAggDevCntl}}$. 0h: SAM entry is not aggregated across Devices. 1h: SAM entry is aggregated across 2-Devices. 2h: SAM entry is aggregated across 4-Devices. 3h: SAM entry is aggregated across 8-Devices. 4h: SAM entry is aggregated across 16-Devices. All other values: Reserved A SAM Entry must not have both Port Aggregation and SA Device Aggregation enabled, i.e. NumAggPorts and NumAggDevCntl must not both be set to non-zero values for the same SAM Entry. This field must be set to 00h when HACapStat.SADevAggSrcCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW
20:17	**DeviceInterleaveCntl** This field controls address interleave of the SAM entry across all aggregated CCIX Devices. The interleave size is $2^{(\text{DeviceInterleaveCntl} + 6)\text{ Bytes.}}$ This field only applies to HSAM Entries. 0h: 64B Interleaved 1h: 128B Interleaved 2h: 256B Interleaved 3h: 512B Interleaved 4h: 1KB Interleaved … Fh: 2MB Interleaved This field must be set to 00h when HACapStat.SADevAggSrcCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW
31:21	Reserved and Preserved	RsvdP

Table 6-22 describes the SAM Entry Address 0 (SAMEntryAddr0) Register field at Byte Offset-04h.

Table 6-22: SAMEntryAddr0 Register field at Byte Offset-04h

Bit Location	Register Description	Attributes
31:0	**StartAddr** Indicates the 4GB or 2^n size aligned Start Address described in this SAM entry. The constraint of 4GB or 2^n size alignment is based on the ComnCapStat2.SAMAlignCap value for this CCIX Device (see Table 6-13).	RW

Table 6-23 describes the SAM Entry Address 1 (SAMEntryAddr1) Register field at Byte Offset-08h.

Table 6-23: SAMEntryAddr1 Register field at Byte Offset-08h

Bit Location	Register Description	Attributes
31:0	**EndAddr** Indicates the 4GB or 2^n size aligned End Address described in this SAM entry. The constraint of 4GB or 2^n size alignment is based on the ComnCapStat2.SAMAlignCap value for this CCIX Device (see Table 6-13).	RW

6.2.2.3.2 Hash Mask

Table 6-24 describes the SAM Hash Mask 0 (SAMHashMask0) Register field at Byte Offset-00h of the SAM Hash Mask in Figure 6-23.

Table 6-24: SAMHashMask0 Register field at Byte Offset-00h of the SAM Hash Mask

Bit Location	Register Description	Attributes
5:0	Reserved and Preserved	RsvdP
31:6	**SAMHashMaskLo** This field indicates which lower address bits are part of the mask result for the aggregated CCIX Ports. This field is only valid when the SAMEntryAttr.NumAggPorts field in Table 6-21 is a valid, non-zero, value. For each bit: 0b: indicates the address bit is not part of the hash mask result. 1b: indicates the address bit is part of the hash mask result SAMHashMaskLo[0] must be 0b if the ComnCntl2.CachelineSizeEnable field indicates 128B Cacheline size is enabled.	RW

Table 6-25 describes the Hash Mask 1 (SAMHashMask1) Register field at Byte Offset-04h of the SAM Hash Mask in Figure 6-23.

Table 6-25: SAMHashMask1 Register field at Byte Offset-04h of the SAM Hash Mask

Bit Location	Register Description	Attributes
31:0	**SAMHashMaskHi** This field indicates which upper address bits are part of the hash function for the aggregated CCIX Ports. This field is only valid when the SAMEntryAttr.NumAggPorts field in Table 6-21 is a valid non-zero value. For each bit: 0b: indicates the address bit is not part of the hash mask result. 1b: indicates the address bit is part of the hash mask result. For balanced distribution across the aggregated CCIX Ports, the total number of bits set to 1b across {SAMHashMaskHi[MAW-1:32], SAMHashMaskLo[31:6]} must be ≥ n, where 2^n is number of aggregated CCIX Ports indicated by the SAMEntryAttr.NumAggPorts field in Table 6-21. If different SAM entries have different SAMEntryAttr.NumAggPorts values, then the rule for balanced distribution is applied based on largest SAMEntryAttr.NumAggPorts value across valid SAM entries.	RW

6.2.2.3.3 Aggregated Port Selection Function

Figure 6-25 illustrates the Aggregated Port Selection Function (APSF) that is applied to the input address when the destination type indicates an aggregated CCIX Port.

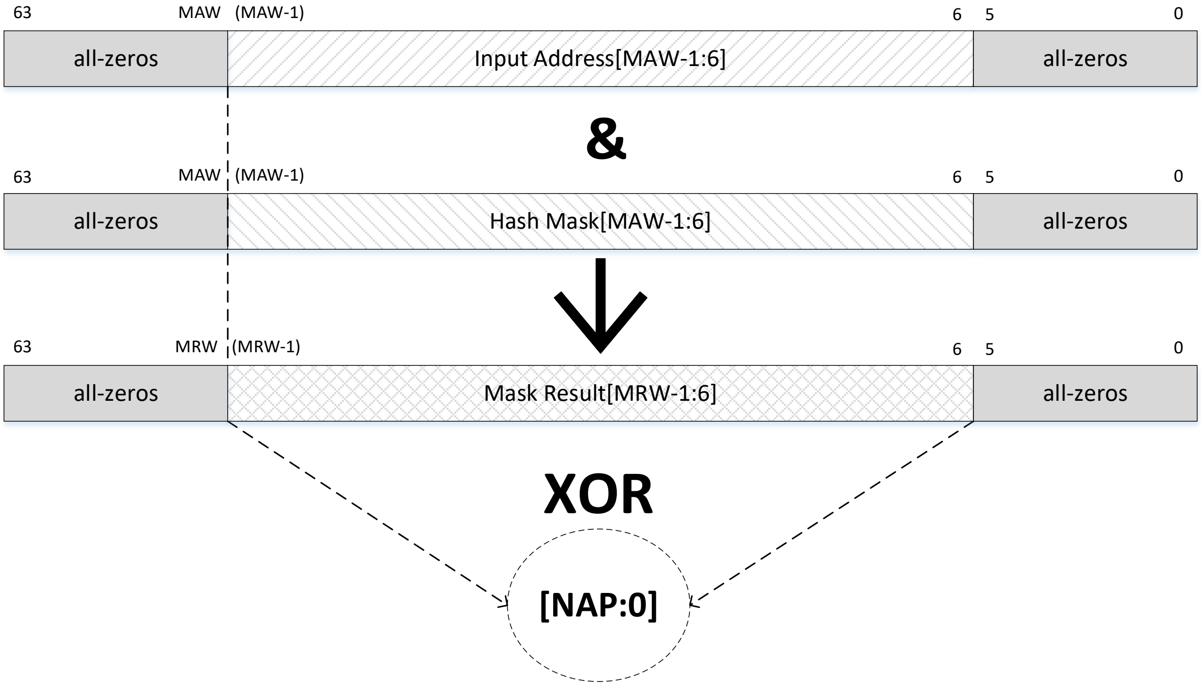


Figure 6-25: Aggregated Port Selection Function

- The Cacheline aligned Input Address is entered into the selection function
  - In Figure 6-25 the CachelineSizeEnable value that is used indicates 64B Cacheline Size but the APSF remains the same if the CachelineSizeEnable value used were to indicate a 128B Cacheline Size.
  - In Figure 6-25 the AddrWidthEnable value that is used indicates the Max Address Width (MAW) enabled is less than 64, and therefore Input Address[63:MAW] must be zero. But the APSF remains the same if the AddrWidthEnable value used were to indicate MAW is 64.
- The Cacheline aligned Input Address is filtered through the Hash Mask to generate the Mask_Result.
- The Mask_Result goes through a hash function to determine the aggregated CCIX Port. M is the integer multiple necessary to achieve a Max Result Width (MRW) greater than or equal to the Max Address Width. The "^" symbol represents the bitwise XOR operator, with the bit-width of the operands being a function of the number of aggregated ports:
  - 2^n CCIX Ports: CCIX Port[n-1:0] = Mask_Result[6+M*n-1:6+(M-1)*n]… ^ Mask_Result[6+2n-1:6+n] ^ Mask_Result[6+n-1:6]
  - 2 CCIX Ports: CCIX Port[0] = …Mask_Result[7] ^ Mask_Result[6]
  - 4 CCIX Ports: CCIX Port[1:0] = …Mask_Result[9:8] ^ Mask_Result[7:6]
  - 8 CCIX Ports: CCIX Port[2:0] = …Mask_Result[11:9] ^ Mask_Result[8:6]
  - 16 CCIX Ports: CCIX Port[3:0] = …Mask_Result[13:10] ^ Mask_Result[9:6]

- The hash function to determine the aggregated CCIX Port for 128B Cacheline size is the same as above due to the SAMHashMaskLo[0] restriction as noted in Table 6-24. If the MAW is not an integer multiple of "n" in 2^n CCIX Ports, then the MSBs of the Mask_Result must be zero-extended up to the Max Result Width.
    - For 64B Cacheline: Mask_Result[6+M*n-1:MAW] are all-zeros
    - For 128B Cacheline: Mask_Result[7+M*n-1:MAW] are all zeros
    - Having Mask_Result zero-extended does not lead to an imbalance in address distribution across those 2^n CCIX Ports because of the "≥ n" Hash Mask restriction as described in Table 6-25.

6.2.2.3.4 Aggregated SA Selection Function

Figure 6-25 illustrates the Aggregated SA Selection Function (ASSF) that is applied to the input address when the destination type indicates an aggregated SA and the SAM Hash Mask single level hash is not used for SA Device Aggregation.

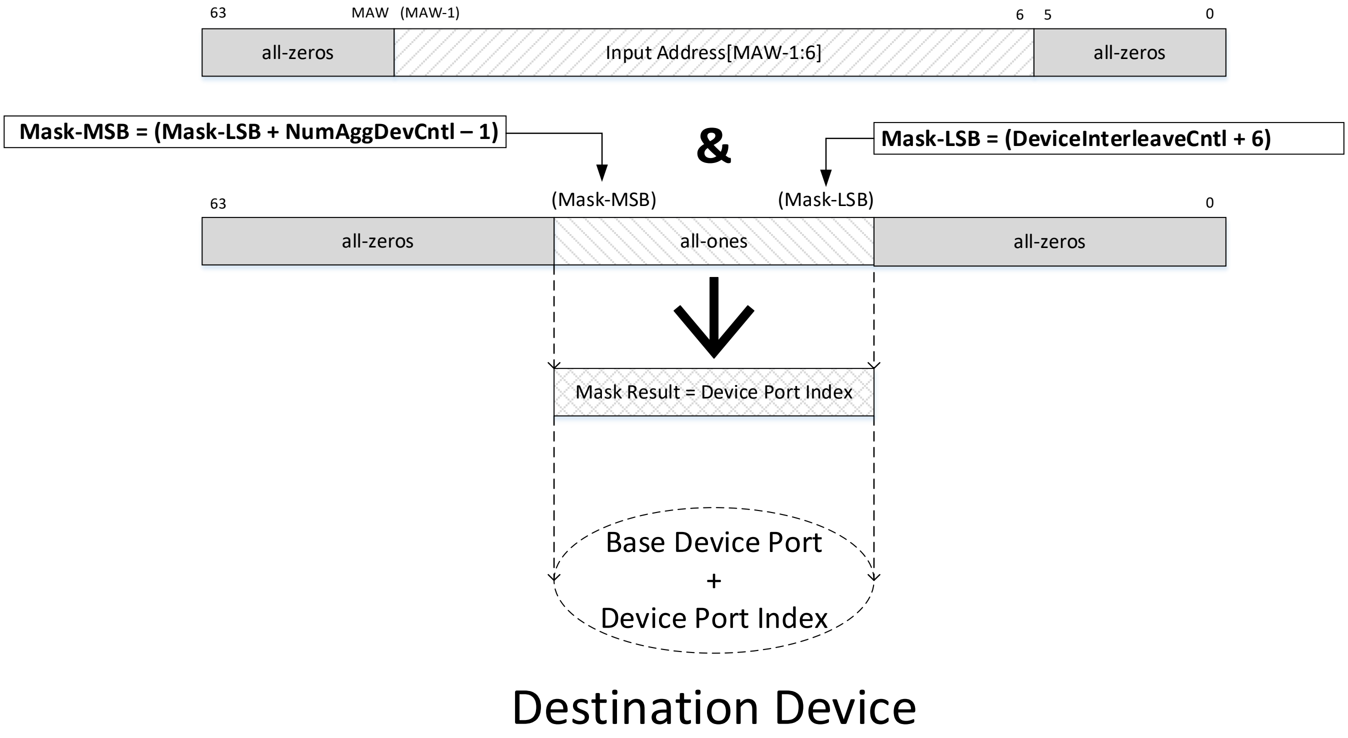


Figure 6-26: Aggregated SA Selection Function

- The Cacheline aligned Input Address is entered into the selection function if the Input Address falls within a SAM Entry with a non-zero NumAggDevCntl value:
    - In Figure 6-25 the CachelineSizeEnable value that is used indicates 64B Cacheline Size but the ASSF remains the same if the CachelineSizeEnable value used were to indicate a 128B Cacheline Size.

  - In Figure 6-25 the AddrWidthEnable value that is used indicates the Max Address Width (MAW) enabled is less than 64, and therefore Input Address[63:MAW] must be zero. But the ASSF remains the same if the AddrWidthEnable value used were to indicate MAW is 64.

- The Cacheline aligned Input Address is filtered through the Hash Mask to generate the Mask_Result:
  - As indicated in Figure 6-25, the LSB of the Hash Mask is based on the DeviceInterleaveCntl field of the SAM Entry with a non-zero NumAggDevCntl value.
  - As indicated in Figure 6-25, the MSB of the Hash Mask is based on the NumAggDevCntl field of the same SAM Entry and the calculated LSB.
- The Mask_Result determines the Device Port Index.
- The Device selected by the ASSF is connected via the CCIX Port at the index position relative to the Base Port ID indicated in the SAM Entry.

6.2.2.3.4.1 HSAM Hash Mask control for SA Device Aggregation

Figure 6-25 illustrates the rules for software to set the HSAM Hash Mask in the HSAM Table when the HA is not Device Aggregation Capable, but SA Device Aggregation is achieved using the Port Aggregation Selection Function instead, as described in Section 6.2.2.3.3. HSAM Hash Mask based address distribution to aggregated SA Devices still requires that the SA Devices must all support SA Device Aggregation.

When the HA to SA Device Aggregation is achieved using the Port Aggregation Selection Function, the programmed HSAM Hash Mask must match the values programmed in following fields of the HSAM Entry's address decode destination SA's BAT Entry:

- As indicated in Figure 6-25, the LSB of the HSAM Hash Mask is based on the DeviceInterleaveCntl field of the BAT Entry with a non-zero BATBaseAddrTypeEntryCntl0.NumAggDevCntl or BATFixedOffsetTypeEntryCntl.NumAggDevCntl value of the Aggregated SA Device that is the address decode destination of the Port Aggregated HSAM Entry.
- As indicated in Figure 6-25, the MSB of the HSAM Hash Mask is based on the NumAggDevCntl field of the BAT Entry with a non-zero BATBaseAddrTypeEntryCntl0.NumAggDevCntl or BATFixedOffsetTypeEntryCntl.NumAggDevCntl value of the Aggregated SA Device that is the address decode destination of the Port Aggregated HSAM Entry, and the calculated LSB.

The resultant SAM Hash Mask illustrated in Figure 6-25 is applied as the APSF's Hash Mask in Figure 6-25.

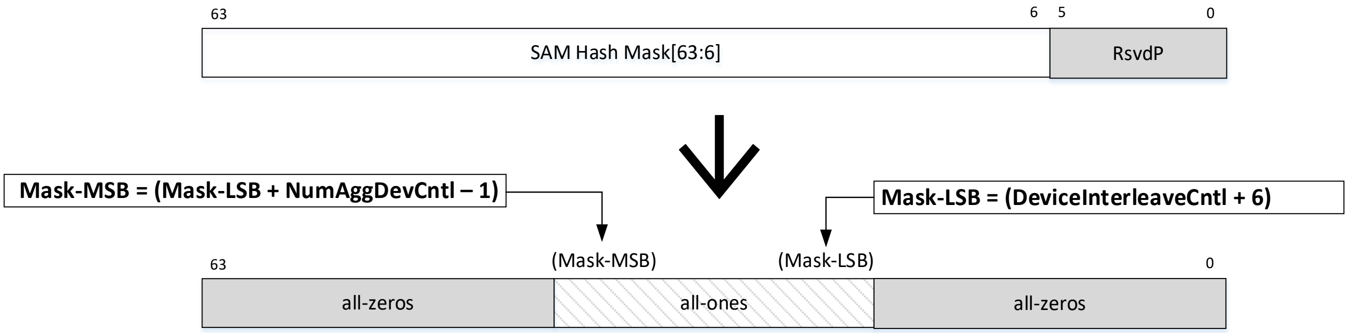

63
6
5
0
SAM Hash Mask[63:6]
RsvdP
Mask-MSB = (Mask-LSB + NumAggDevCntl – 1)
Mask-LSB = (DeviceInterleaveCntl + 6)
63
(Mask-MSB)
(Mask-LSB)
0
all-zeros
all-ones
all-zeros

Figure 6-27: HSAM Hash Mask settings for SA Device Aggregation

6.2.2.3.5 SAM Table Usage and Restrictions for CCIX Components

As described in Section 6.2.1.8 , there are two unique Address Maps, both of which are described by the SAM Table data structure. The Request Agent Address Map across the CCIX System is described in RSAM Tables, present in CCIX Devices that contain Request Agents or multi-port CCIX Devices that support CCIX Port-to-Port forwarding. The Home Agent Address Map across the CCIX System is described in the HSAM Tables, present in CCIX Devices that contain Home Agents that have Memory Expansion enabled, or multi-port CCIX Devices that support CCIX Port-to-Port forwarding.

This section describes how the SAM Table structure is referenced by CCIX Components and restrictions based on the CCIX Component Type:

- CCIX Port:
  - The SAM structure is referenced in order to resolve either the next-hop destination CCIX Port or the last-hop destination Local CCIX Agent for inbound Address routed CCIX Packets. How the CCIX Packet is routed to the last-hop Local CCIX Agent is implementation specific.
  - Based on the CCIX Link Entry Address Type, described further in Table 6-40, a CCIX Link may reference either RSAM Tables, or HSAM Tables:
    - The RSAM Table is referenced for Address routed CCIX Packets that originated in a Request Agent and have a Home Agent destination, whether the SAM entry identifies the destination as a local Home Agent or whether the packet is to be routed to the next destination CCIX Port.
    - Similarly, the HSAM Table is referenced for Address routed CCIX Packets that originated in a Home Agent and have a Subordinate Agent destination, whether the SAM entry identifies the destination as a local Subordinate Agent or whether the packet is to be routed to the next destination CCIX Port.
- CCIX Agents (HA/RA/SA):
  - The SAM structure is referenced in order to resolve if the address routed packet is Local or routed to a CCIX Port.

  - The Address used to match into the SAM may only belong to CCIX Ports, i.e. SAM entries may only have a SAMEntryAttr.DestType encoding of 1b (see Table 6-21).
  - The SAM structure is not referenced in order to resolve to a destination Local CCIX Agent; the routing from source CCIX Agents to destination CCIX Agents on the same CCIX Device is implementation specific, i.e. disabling an Address mapping in the SAM structure may not disable access from source CCIX Agents to destination CCIX Agents on the same CCIX Device.

- HA:

  When Memory Expansion is enabled, the HSAM structure is referenced in order to resolve a Subordinate Agent address to its destination CCIX Port.

  The RSAM structure, if present, can be referenced if the Home Agent is the Aggregated Local CCIX Agent target in order to determine the APSF for Snoops to RAs.

- RA:

  The RSAM structure is referenced in order to resolve a Home Agent address to its destination CCIX Port.

6.2.2.4 Memory Pool and BAT Structures

Memory Pool Capability structures describe the size, type, and attributes of the memory pools. The Base Address Table (BAT) is the corresponding Control structure associated with the Memory Pool Capability structure. The BAT is used to resolve the destination Memory Pool for a given Address in the SAM.

Memory Pool Capability structures are declared by Home Agents and Subordinate Agents only. Similarly, the corresponding BAT Control structure is referenced by Home Agents and Subordinate Agents only and its usage is further described in Section 6.2.2.7 and Section 6.2.2.9 respectively. There is a one-to-one correspondence between a Memory Pool Capability structure, also known as a Memory Pool Entry, and a BAT Control structure, also known as a BAT Entry. Therefore, Home Agents and Subordinate Agents must have the same number of Memory Pool Entries and BAT Entries.

The BAT structure contains one or more entries where the entries are formatted either as Base Address Type entries or Fixed Offset Type entries based on the Memory Pool Addressing Capability field in the CCIX Agent’s corresponding Memory Pool Capabilities & Status structure. When the Memory Pool Addressing Capability field indicated is Fixed Offset type, the enabled BAT entry is at a Fixed Offset from the Base Address of the previous enabled BAT entry associated with a Memory Pool that has Base Address type Addressing Capability.

6.2.2.4.1 Memory Pool Capabilities & Status Structure

Figure 6-28 shows overall the layout of Memory Pool Capabilities & Status structure. The structures from Entry 0 to Entry n linearly describe the Capabilities & Status of Memory Pools 0 to n.

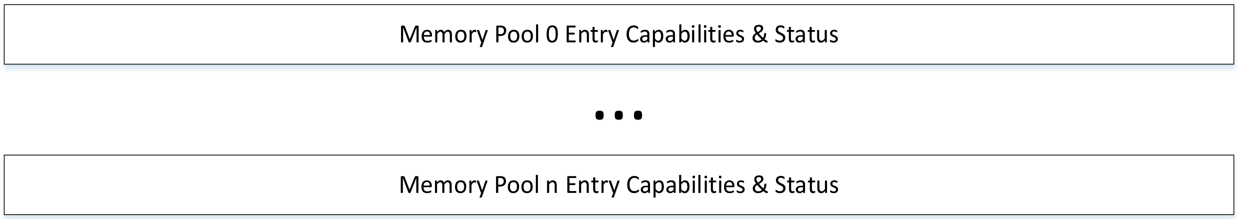

Memory Pool 0 Entry Capabilities & Status
• • •
Memory Pool n Entry Capabilities & Status

Figure 6-28: Memory Pool Capabilities & Status structure

Figure 6-29 shows the layout of registers for a Memory Pool Capabilities & Status Entry.

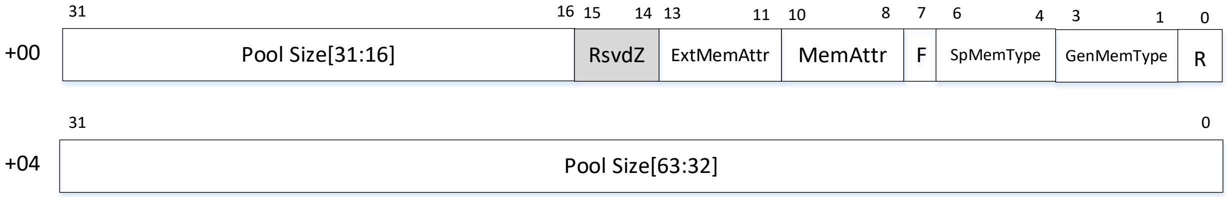


Figure 6-29: Memory Pool Entry Capabilities & Status Registers

Table 6-26 describes the Memory Pool Entry Capabilities & Status 0 (MemPoolEntryCapStat0) register fields at Byte Offset-00h.

Table 6-26: MemPoolEntryCapStat0 Register fields at Byte Offset-00h

Bit Location	Register Description	Attributes
0	**MemPoolDiscRdyStat** This field indicates this Memory Pool's Discovery Ready Status. 0b: Indicates the Memory Pool entry and its capabilities & status are not ready to be discovered and configured. 1b: Indicates the Memory Pool entry and its capabilities & status are ready to be discovered and configured.	RO

Bit Location	Register Description	Attributes
3:1	**MemPoolGenMemTypeCap** This field indicates this Memory Pool's General Memory Type Capability. 0h: Indicates Other and/or Non-Specified Memory Type. 1h: Indicates Expansion Memory Type. • A Memory Expansion Pool must have an MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b (see Table 6-26), i.e. the pool must have 4GB or minimum 4GB, 2^n size aligned addressing capability. • This encoding is only allowed for Home Agent Memory Pool entries and reserved for Subordinate Agent Memory Pool entries. • This encoding is only allowed if the HA claims memory expansion capability via the HACapStat.HAMemExpnCap bit. • The Total Pool Size, [MemPoolSizeCapHi,MemPoolSizeCapLo], indicates the size of the Memory Expansion Pool that can be hosted by this HA. • A Memory Expansion Pool may be mapped to one or more SAs. • The MemPoolSpcificMemTypeCap field must be 0h. 2h: Indicates a Memory Hole. • There is no memory in this Memory Pool, and the CCIX Device has reserved this memory region to prevent it from being addressable as System Memory. • This encoding is allowed for Home Agent and Subordinate Agent Memory Pool entries. • The Total Pool Size, [MemPoolSizeCapHi,MemPoolSizeCapLo], indicates the size of the Memory Hole. • The MemPoolSpcificMemTypeCap field must be 0h. 3h: Indicates ROM Memory Type. 4h: Indicates Volatile Memory Type. 5h: Indicates Non-Volatile Memory Type. 6h: Indicates Device/Register Memory Type. All other encodings: Reserved. The MemPoolGenMemTypeCap encoding of 0h is permitted for declaring a memory type that could also have been described by encodings 3h – 6h.	RO

Bit Location	Register Description	Attributes
6:4	**MemPoolSpcificMemTypeCap** This field indicates this Memory Pool's Specific Memory Type Capability. This is an optional capability, and if not supported, the field must indicate a value of 0h. 0h: Indicates Other and/or Non-Specified Memory Type. 1h: Indicates SRAM Memory Type. 2h: Indicates DDR Memory Type. 3h: Indicates NVDIMM-F Memory Type. 4h: Indicates NVDIMM-N Memory Type. 5h: Indicates HBM Memory Type. 6h: Indicates Flash Memory Type. 7h: Reserved. The MemPoolSpcificMemTypeCap encoding of 0h is permitted for declaring a memory type that could also be described by encodings 1h – 6h.	RO
7	**MemPoolAddrCap** This field indicates this Memory Pool's Addressing Capability. 0b: • Indicates this Memory Pool Entry must be addressed by a 4GB aligned Base Address, or a minimum 4GB, 2^n size aligned Base Address. The constraint of 4GB or 2^n size alignment is based on the ComnCapStat2.SAMAlignCap value for this CCIX Device (see Table 6-13). • The MemPoolAddrCap field of Memory Pool Entry 0 must indicate a value of 0b. 1b: • Indicates that this Memory Pool Entry n must be at a fixed offset from Memory Pool Entry (n-1). • This encoding is allowed for both Home Agent and Subordinate Agent Memory Pool entries. • The fixed offset of Memory Pool Entry n is required to be fixed at the Base Address + Total Pool Size, where Total Pool Size is the sum of the sizes of all Memory Pool Entries in the Memory Pool Group preceding Memory Pool Entry n.	RO
10:8	**MemPoolMemAttr** This field indicates this Memory Pool's General Memory Attribute. 000b: Indicates CCIX Device Memory Attribute. 100b: Indicates CCIX Normal Non-Cacheable Memory Attribute . 101b: Indicates CCIX Normal Cacheable Memory Attribute. All other encodings: Reserved. While memory attribute capabilities are declared in DVSEC data structures, memory attribute control is via Page Table Entries. This includes control of the fine-grained memory attributes expressed in the ReqAttr field of a CCIX packet.	RO

Bit Location	Register Description	Attributes
13:11	**MemPoolExtMemAttr** This field indicates this Memory Pool's Extended Memory Attribute. 000b: Indicates System Memory with no Extended Memory Attributes. 001b: Indicates Private Memory. Unlike System Memory, which has no allocation restrictions across the system, Private Memory has allocation restrictions enforced via the CCIX Device Driver (akin to PCIe Driver controlled MMIO). All other encodings: Reserved.	RO
15:14	Reserved and Zero	RsvdZ
31:16	**MemPoolSizeCapLo** This field indicates the 16 LSB bit encodings for the Pool Size supported by this Memory Pool. The Total Pool Size Capability, in integer multiples of 64KB, is indicated by the combination of the Lower and Upper Bits of the Memory Pool Size Capability field or [MemPoolSizeCapHi, MemPoolSizeCapLo] where MemPoolSizeCapHi is described in the MemPoolEntryCapStat1 Register at Byte Offset 04h of the BAT Capabilities & Status entry. [MemPoolSizeCapHi,MemPoolSizeCapLo]: Size Capability. Examples: [00000000h, 0000h]: 64K Pool Size Capability. [00000000h, 05FFh]: 96MB Pool Size Capability. [00000000h, FFFFh]: 4GB Pool Size Capability. [00000001h, 7FFFh]: 6GB Pool Size Capability.	RO

Table 6-27 describes the Memory Pool Entry Capabilities & Status 1 (MemPoolEntryCapStat1) Register fields at Byte Offset-04h.

Table 6-27: MemPoolEntryCapStat1 Register fields at Byte Offset-04h

Bit Location	Register Description	Attributes
31:0	**MemPoolSizeCapHi** This field indicates the 32 MSB bit encodings for the Pool Size Supported by this Memory Pool. The Total Pool Size Capability, in integer multiples of 64KB, is indicated by the combination of the Lower and Upper Bits of the Memory Pool Size Capability field or [MemPoolSizeCapHi,MemPoolSizeCapLo].	RO

6.2.2.4.2 BAT Control Structure

Figure 6-30 shows overall the layout of BAT Control structure. The structures from Entry 0 to Entry n linearly describe the Controls for Memory Pools 0 to n.

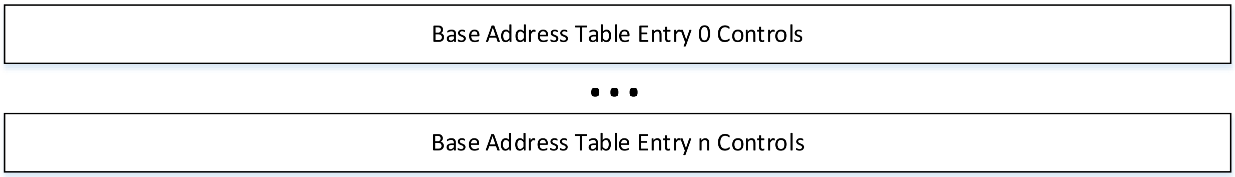

Base Address Table Entry 0 Controls
• • •
Base Address Table Entry n Controls

Figure 6-30: BAT Control structure

The MemPoolEntryCapStat0.MemPoolAddrCap field, described further in Table 6-26, is the only indication whether BAT Entry > 0 is formatted differently from BAT Entry 0.

6.2.2.4.2.1 BAT Base Address Type Control Entry

Figure 6-31 shows the layout of Registers for a BAT Base Address Type Control Entry.

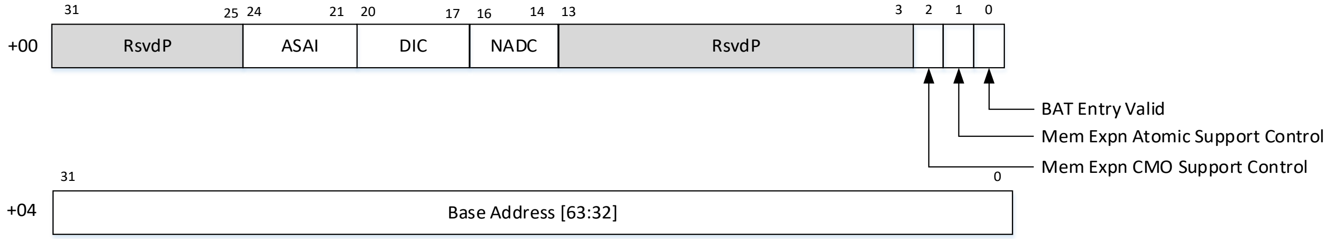


Figure 6-31: BAT Base Address Type Entry (BATBaseAddrTypeEntry) Control Registers

Table 6-28 describes the BAT Base Address Type Entry Control 0 (BATBaseAddrTypeEntryCntl0) Register fields at Byte Offset-00h.

Table 6-28: BATBaseAddrTypeEntryCntl0 Register Fields at Byte Offset-00h

Bit Location	Register Description	Attributes
0	**BATEntryVal** This field indicates whether the BAT Entry is valid. 0b: Indicates this Memory Pool is not enabled, and no SAM Base Address is mapped in this entry. 1b: Indicates this Memory Pool is enabled, and a SAM Base Address is mapped in this entry.	RW

Bit Location	Register Description	Attributes
1	**MemExpnAtomicSupportCntl** This field indicates whether the Subordinate Agent mapped to this BAT Entry has Atomic Support. This field is only valid for Memory Expansion BAT Entries, i.e. BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value of 1h. 0b: Indicates the Subordinate Agent is not capable of supporting Atomic transactions. The HA must service the Atomic transaction itself and can only issue second-order ReadNoSnp and/or WriteNoSnp transactions, if necessary, to the Subordinate Agent as part of the Atomic Operation. 1b: Indicates the Subordinate Agent is capable of supporting Atomic transactions. The HA may optionally offload the Atomic transaction to the Subordinate Agent which is capable of servicing the Atomic transaction. RsvdZ: For BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value other than 1h. CCIX Device initializes to 0b after reset (except FLR).	RW
2	**MemExpnCMOSupportCntl** This field indicates whether the Subordinate Agent mapped to this BAT Entry has CMO Support. This field is only valid for Memory Expansion BAT Entries, i.e. BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value of 1h. 0b: Indicates the Subordinate Agent is not capable of supporting CMO transactions. The HA must service the CMO transaction itself. 1b: Indicates the Subordinate Agent is capable of supporting CMO transactions. The HA may optionally offload the servicing of the CMO transaction to the Subordinate Agent which is capable of servicing the CMO transaction. RsvdZ: For BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value other than 1h. CCIX Device initializes to 0b after reset (except FLR).	RW
13:3	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
16:14	**NumAggDevCntl** This field controls the number of aggregated CCIX Devices. This field only applies to SBAT Entries. The number of aggregated devices is $2^{\text{NumAggDevCntl}}$. 0h: Device Not Aggregated 1h: Device is part of 2-Device Aggregation 2h: Device is part of 4-Device Aggregation 3h: Device is part of 8-Device Aggregation 4h: Device is part of 16-Device Aggregation All other values: Reserved This field must be set to 00h when SACapStat.SADevAggDestCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW
20:17	**DeviceInterleaveCntl** This field controls the interleave size across all aggregated CCIX Devices. The interleave size is $2^{(\text{DeviceInterleaveCntl + 6) Bytes.}}$ This field only applies to SBAT Entries. 0h: 64B Interleaved 1h: 128B Interleaved 2h: 256B Interleaved 3h: 512B Interleaved 4h: 1KB Interleaved … Fh: 2MB Interleaved This field must be set to 00h when SACapStat.SADevAggDestCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW

Bit Location	Register Description	Attributes
24:21	**AggSAIndex** This field indicates this SA's Index position with respect to the other aggregated SAs. This field only applies to HBAT Entries. Encodings > 1h are reserved if NumAggDevCntl is 1h. Encodings > 3h are reserved if NumAggDevCntl is 2h. Encodings > 7h are reserved if NumAggDevCntl is 3h. 0h: This SA is in index position 0. 1h: This SA is in index position 1. 2h: This SA is in index position 2. … Fh: This SA is in index position 15. This field must be set to 00h when SACapStat.SADevAggDestCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW
31:25	Reserved and Preserved	RsvdP

Table 6-29 describes the BAT Base Address Type Entry Control 1 (BATBaseAddrTypeEntryCntl1) Register fields at Byte Offset-04h.

Table 6-29: BATBaseAddrTypeEntryCntl1 Register fields at Byte Offset-04h

Bit Location	Register Description	Attributes
31:0	**BATEntryBaseAddr** Indicates the 4GB aligned Base Address, or a minimum 4GB, 2^n size aligned Base Address, of the Memory Pool described in this BAT entry. The constraint of 4GB or 2^n size alignment is based on the ComnCapStat2.SAMAlignCap value for this CCIX Device (see Table 6-13).	RW

6.2.2.4.2.2 BAT Fixed Offset Type Control Entry

Figure 6-32 shows the layout of the BAT Fixed Offset Type Control Entry Register.

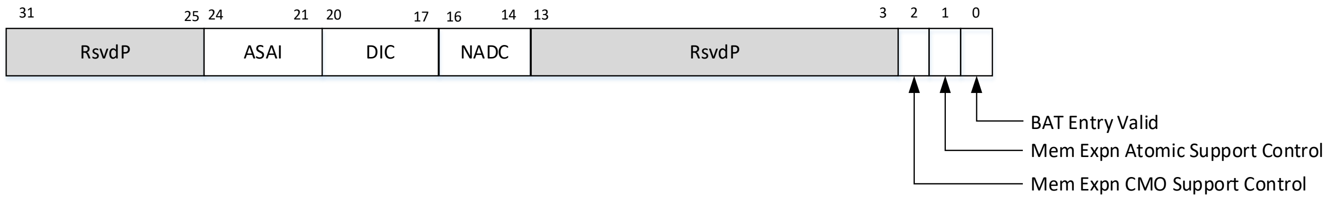


Figure 6-32: BAT Fixed Offset Type Control Entry

Table 6-30 describes the BAT Fixed Offset Type Entry Control (BATFixedOffsetTypeEntryCntl) Register fields.

Table 6-30: BATFixedOffsetTypeEntryCntl Register fields

Bit Location	Register Description	Attributes
0	**BATEntryVal** This field indicates whether the BAT Entry is valid. 0b: Indicates this Memory Pool is not enabled and no SAM Address is mapped in this entry. 1b: Indicates this Memory Pool is enabled, and a SAM Address is mapped in this entry.	RW
1	**MemExpnAtomicSupportCntl** This field indicates whether the Subordinate Agent mapped to this BAT Entry has Atomic Support. This field is only valid for Memory Expansion BAT Entries, i.e. BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value of 1h. 0b: Indicates the Subordinate Agent is not capable of supporting Atomic transactions. The HA must service the Atomic transaction itself and can only issue second-order ReadNoSnp and/or WriteNoSnp transactions, if necessary, to the Subordinate Agent as part of the Atomic Operation. 1b: Indicates the Subordinate Agent is capable of supporting Atomic transactions. The HA may optionally offload the Atomic transaction to the Subordinate Agent which is capable of servicing the Atomic transaction. RsvdZ: For BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value other than 1h. CCIX Device initializes to 0b after reset (except FLR).	RW

Bit Location	Register Description	Attributes
2	**MemExpnCMOSupportCntl** This field indicates whether the Subordinate Agent mapped to this BAT Entry has CMO Support. This field is only valid for Memory Expansion BAT Entries, i.e. BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value of 1h. 0b: Indicates the Subordinate Agent is not capable of supporting CMO transactions. The HA must service the CMO transaction itself. 1b: Indicates the Subordinate Agent is capable of supporting CMO transactions. The HA may optionally offload the servicing of the CMO transaction to the Subordinate Agent which is capable of servicing the CMO transaction. RsvdZ: For BAT Entries where the corresponding Memory Pool Entry indicates a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value other than 1h. CCIX Device initializes to 0b after reset (except FLR).	RW
3:13	Reserved and Preserved	RsvdP
16:14	**NumAggDevCntl** This field controls the number of aggregated CCIX Devices. This field only applies to HBAT Entries. The number of aggregated devices is $2^{NumAggDevCntl}$. 0h: Device Not Aggregated 1h: Device is part of 2-Device Aggregation 2h: Device is part of 4-Device Aggregation 3h: Device is part of 8-Device Aggregation 4h: Device is part of 16-Device Aggregation All other values: Reserved This field must be set to 00h when SACapStat.SADevAggDestCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW

Bit Location	Register Description	Attributes
20:17	**DeviceInterleaveCntl** This field controls the interleave size across CCIX Devices. The interleave size is $2^{(\text{DeviceInterleaveCntl} + 6)\text{ Bytes.}}$ This field only applies to HBAT Entries. 0h: 64B Interleaved 1h: 128B Interleaved 2h: 256B Interleaved 3h: 512B Interleaved 4h: 1KB Interleaved … Fh: 2MB Interleaved This field must be set to 00h when SACapStat.SADevAggDestCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW
24:21	**AggSAIndex** This field indicates this SA's Index position with respect to the other aggregated SAs. This field only applies to HBAT Entries. Encodings > 1h are reserved if NumAggDevCntl is 1h. Encodings > 3h are reserved if NumAggDevCntl is 2h. Encodings > 7h are reserved if NumAggDevCntl is 3h. 0h: This SA is in index position 0. 1h: This SA is in index position 1. 2h: This SA is in index position 2. … Fh: This SA is in index position 15. This field must be set to 00h when SACapStat.SADevAggDestCap is set to 0b. CCIX Device initializes to 00h after reset (except FLR).	RW
31:25	Reserved and Preserved	RsvdP

6.2.2.4.3 Relation between HA Memory Pool Structures and HBAT Entry Structures

Figure 6-33 illustrates the various Home Agent Memory Pool Structure types and the relation to their corresponding HBAT Entry Stucture types.

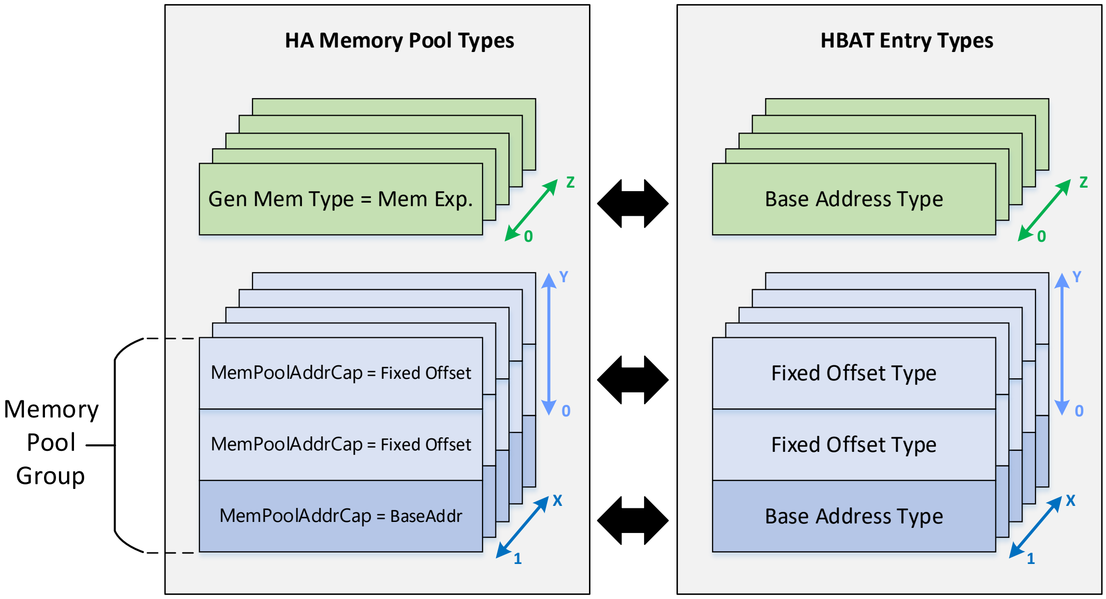


Figure 6-33: Relation between HA Memory Pools and HBAT Entries

Base Address Type Memory Pools and HBAT Entries as illustrated in Figure 6-33:

- A Home Agent must have a minimum of one Memory Pool that is allowed to be mapped to a 4GB or 2^n size aligned Base Address in the G-RSAM, i.e. at least one Memory Pool must declare a MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b (see Table 6-26). The constraint of 4GB or 2^n size alignment is based on the SAMAlignCap value for the HA's CCIX Device (see Table 6-13).
  - Although illustrated separately in Figure 6-33, the minimum requirement of one Memory Pool for an HA can also be satisfied by a Memory Expansion Pool, described further later in this section.
- Memory Pool Entry 0 of an HA must declare a MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b.
- For each 1-to-X number of HA Memory Pool Entries with a declared MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b, there must be a corresponding 1-to-X number of HBAT Entries which can only be programmed with a 4GB or 2^n size aligned Base Address (see Section 6.2.2.4.2.1).

Fixed Offset Type Memory Pools and HBAT Entries as illustrated in Figure 6-33:

- For each 1-to-X number of HA Memory Pool Entries with a declared MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b, an HA is permitted to have 0-to-Y number of Memory Pool Entries that declare they can only be mapped at a fixed offset to the corresponding 1-to-X HA Memory Pools. This means 0-to-Y number of Memory Pool Entries that follow a Memory Pool Entry with a declared MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b, can declare a MemPoolEntryCapStat0.MemPoolAddrCap field value of 1b in their own Memory Pool Entry.
  - For every Memory Pool Entry #Y of Fixed Offset Type, Memory Pool Entry #(Y-1) must either also be a Memory Pool Entry of Fixed Offset Type, or a Memory Pool Entry of Base Address Type, i.e. Memory Pool Entries 1-to-Y must be contiguously numbered up from a Memory Pool Entry #X of Base Address Type. The Memory Pool Entry #X of Base Address Type, followed by contiguously numbered Memory Pool Entries 1-to-Y of Fixed Offset Type, constitutes a Memory Pool Group.

- For each 0-to-Y number of HA Memory Pool Entries with a declared MemPoolEntryCapStat0.MemPoolAddrCap field value of 1b, there must be a corresponding 0-to-Y number of HBAT Entries which can only be enabled to be at a Fixed Offset to the Base Address in their corresponding 1-to-X HBAT Entry.
    - For every HBAT Entry #Y of Fixed Offset Type, the first HBAT Entry #X of Base Address Type, with the smallest value of X < Y, contains the corresponding Base Address from which HBAT Entry #Y is at a Fixed Offset.

Memory Expansion Pools and HBAT Entries as illustrated in Figure 6-33:

- An HA is permitted to have 0-to-Z number of Memory Pools that declare they are capable of Memory Expansion to SA Memory. This means 0-to-Z number of Memory Pool Entries declare a MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value of 1h.
- For each 0-to-Z number of Memory Pool Entries that declare an MemPoolEntryCapStat0.MemPoolGenMemTypeCap field value of 1h, there must be a corresponding 0-to-Z number of HBAT Entries which can only be programmed with a 4GB or 2^n size aligned Base Address. See Section 6.2.2.4.2.1.
- Since an HA is permitted to have only Memory Pools that are capable of Memory Expansion, Memory Expansion Pool Entry 0 of that HA must declare an MemPoolEntryCapStat0.MemPoolAddrCap field value of 0b.

6.2.2.5 CCIX Port Structures

Figure 6-34 shows the overall layout of the CCIX Port Capabilities & Status Registers.

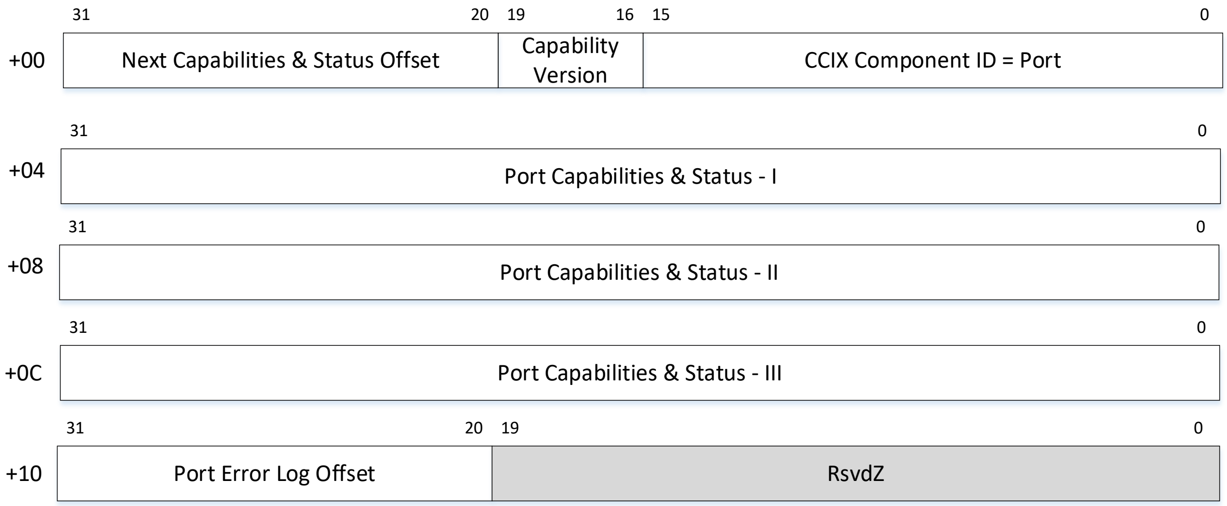

31
20
19
16
15
0
+00
Next Capabilities & Status Offset
Capability Version
CCIX Component ID = Port
31
0
+04
Port Capabilities & Status - I
31
0
+08
Port Capabilities & Status - II
31
0
+0C
Port Capabilities & Status - III
31
20
19
0
+10
Port Error Log Offset
RsvdZ

Figure 6-34: CCIX Port Capabilities & Status Registers

6.2.2.5.1 CCIX Port Capabilities & Status Register

Figure 6-35 shows the layout of the CCIX Port Capabilities & Status 1 (PortCapStat1) Register at Byte Offset-04h.

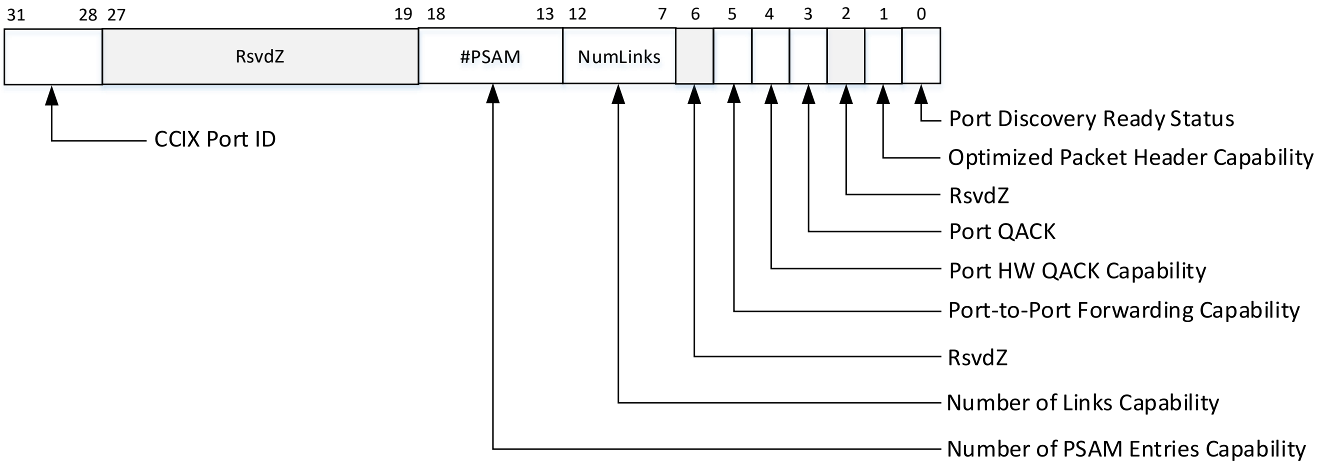

31
28
27
19
18
13
12
7
6
5
4
3
2
1
0
RsvdZ
#PSAM
NumLinks
CCIX Port ID
Port Discovery Ready Status
Optimized Packet Header Capability
RsvdZ
Port QACK
Port HW QACK Capability
Port-to-Port Forwarding Capability
RsvdZ
Number of Links Capability
Number of PSAM Entries Capability

Figure 6-35: PortCapStat1 Register at Byte Offset-04h

Table 6-31 describes the PortCapStat1 Register fields at Byte Offset-04h.

Table 6-31: PortCapStat1 Register fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**PortDiscRdyStat** This field describes the CCIX Port's Discovery Readiness Status. 0b: Indicates the CCIX Port and its capabilities and control are not ready to be discovered and configured. 1b: Indicates the CCIX Port and its capabilities and control are ready to be discovered and configured.	RO
1	**PktHdrTypeCap** This field describes the type of CCIX Packet Header the CCIX Port can send and receive. 0b: Indicates only PCIe Compatible Packet Header format supported. 1b: Indicates both CCIX Optimized Packet Header format and PCIe Compatible Packet Header are supported. The value of PktHdrTypeCap must be consistent with the value of OptimizeTLPFormatSuppport as described in Table 6-73.	RO
2	Reserved and Zero	RsvdZ

Bit Location	Register Description	Attributes
3	**PortQACK** This field describes the CCIX Port's Quiesce Acknowledgement status. 0b: CCIX Port not quiesced. 1b: CCIX Port quiesced. If the CCIX Port has Hardware Quiesce Acknowledgement (HW QACK) capability, as indicated by a PortCapStat1.PortHWQACKCap value of 1b, then PortCapStat.PortQACK is set by implementation specific methods. Software can choose to poll PortCapStat1.PortQACK instead of waiting for the <Port Quiesce Time> value to check PortQACK if PortCapStat1.PortHWQACKCap has a value of 1b. If the CCIX Port does not have HW QACK capability as indicated in PortCapStat1.PortHWQACKCap value of 0b, the following sequence is followed: The CCIX Port sets the PortQACK bit to a value 1b after completing or detecting the following actions in this order: 1. The CCIX Port detects that the CCIX Port Quiesce Request (PortCntl.PortQREQ) Control bit is set. 2. The CCIX Port has not issued any Requests and sent and received all relevant outstanding Responses, for the duration of <Port Quiesce Time>. <Port Quiesce Time> is based on the value of ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue, described further in Table 6-12. 3. Following the transition of the CCIX Port Quiesce Request (PortCntl.PortQREQ in Table 6-31) control bit from 0b to 1b, the CCIX Port must take no longer than 2 * <Port Quiesce Time> to set PortQACK. After 2 * <Port Quiesce Time>, CCIX Software detecting a zero returned for the PortQACK field indicates an error condition such that the CCIX Port was unable to reach a quiescent state. Following the transition of the PortCntl.PortQREQ control bit from 1b to 0b, the CCIX Port must transition the PortQACK bit from 1b to 0b.	RO
4	**PortHWQACKCap** This field describes the CCIX Port's Hardware Quiesce Acknowledgement Capability. 0b: • The CCIX Port does not have a hardware mechanism to achieve a quiescent state. 1b: • The CCIX Port has a hardware mechanism to achieve a quiescent state.	RO

Bit Location	Register Description	Attributes
5	**PortToPortFwdingCap** This field describes the ability of the CCIX Port to internally forward CCIX Packets to another CCIX Port on the same CCIX Device. 0b: Indicates the CCIX Port is not capable of CCIX Port-to-Port transaction forwarding (including Broadcast Snoop transaction forwarding). 1b: Indicates the CCIX Port is capable of CCIX Port-to-Port transaction forwarding (including Broadcast Snoop transaction forwarding) and the Forwarding Vectors, i.e. PortCapStat3.PortFwdingVctr described further in Table 6-33, is part of the Port Capabilities & Status data structure. A PortToPortFwdingCap value of 1b also indicates that the Broadcast Forward Control Vector (BCastFwdCntlVctr), described further in Section 6.2.2.6.2.1.1, is part of the CCIX Link Control structure (see Figure 6-47). PortToPortFwdingCap is a bidirectional property, i.e. the capability indicates that all CCIX Packet Types must be forwarded between CCIX Ports, and in either direction, i.e. the CCIX Port must be able to ingress and egress all CCIX Packet Types. While PortToPortFwdingCap is a per-port capability, a value of 1b indicates that the BCastFwdCntlVctr CCIX Link Control structure is present for all CCIX Links on the CCIX Port.	RO
6	Reserved and Zero	RsvdZ
12:7	**NumLinksCap** This field describes the number of CCIX Links Capability for this CCIX Port. The number of CCIX Links supported by a CCIX Port is typically based on the number of unique Source/Destination TransportID (BDF for PCIe) pairs that CCIX Agents on this CCIX Device require to support CCIX traffic to CCIX Agent's on other CCIX Devices. However, additonal links are required to accommodate RA to HA requests and HA to SA requests which are traveling in the same direction. See Chapter 3 for further details. 00h: Reserved. 01h – 3Fh: Encodings indicating capabilities for 1 to 63 CCIX Links. NumLinksCap also indicates the number of Link Attribute Control Entries and Link TransportID Map Entries in the CCIX Link Control structure, as illustrated in Figure 6-46. Unlike AgentID and CCIX Device ID, whose DVSEC field widths allow for their respective architected maximum of 64 and 256 identifiers, there is no architected maximum for CCIX Links. NumLinksCap maximum of 63 CCIX Links per CCIX Port allows for dedicated resources for communication with 63 other ports. This accommodates topologies where an Agent on a CCIX Device with this CCIX Port has dedicated resources for communication with the architected maximum of 63 other Agents, each Agent being located at the 63 CCIX Port destinations.	RO

Bit Location	Register Description	Attributes
18:13	**NumPSAMEntryCap** This field indicates the number of PSAM Entries in the PSAM Table. The minimum number of PSAM entries, described in Section 6.2.2.5.3, supported by a CCIX Port is dependent on the number of CCIX Links supported by that CCIX Port, as indicated by PortCapStat1.NumLinksCap. 00h - 02h: • Encoding for 0 to 2 PSAM entries. • A CCIX Port must have a PortCapStat1.NumLinksCap value of 1 for the NumPSAMEntryCap Encoding to be 00h – 02h. 03h – 3Fh: • Encoding for 3 to 63 PSAM entries. • If a CCIX Port has PortCapStat1.NumLinksCap > 1, the NumPSAMEntryCap Encoding must reflect at least (PortCapStat1.NumLinksCap + 1) PSAM entries. Therefore, NumPSAMEntryCap ≥ (PortCapStat1.NumLinksCap + 1) when PortCapStat1.NumLinksCap > 1.	RO
27:19	Reserved and Zero	RsvdZ
31:28	**PortID** Indicates the CCIX PortID for this CCIX Port. This field must have a value that is unique across a CCIX Device when the device has multiple CCIX capable ports. The CCIX PortID is not required to be unique system-wide, i.e. the CCIX PortID namespace can be re-used across CCIX Devices. 0h – Fh: Encodings for up to 16 CCIX PortIDs. Multi-port CCIX Devices must have linear enumeration of CCIX PortIDs starting with CCIX PortID0. CCIX Ports that can be aggregated with other CCIX Ports on the CCIX Device must have sequential CCIX PortID numbering with their aggregation group as indicated by the PortCapStat1.PortAggVctr, described further in Table 6-32.	RO

Figure 6-36 shows the layout of the CCIX Port Capabilities & Status 2 (PortCapStat2) Register at Byte Offset-08h.

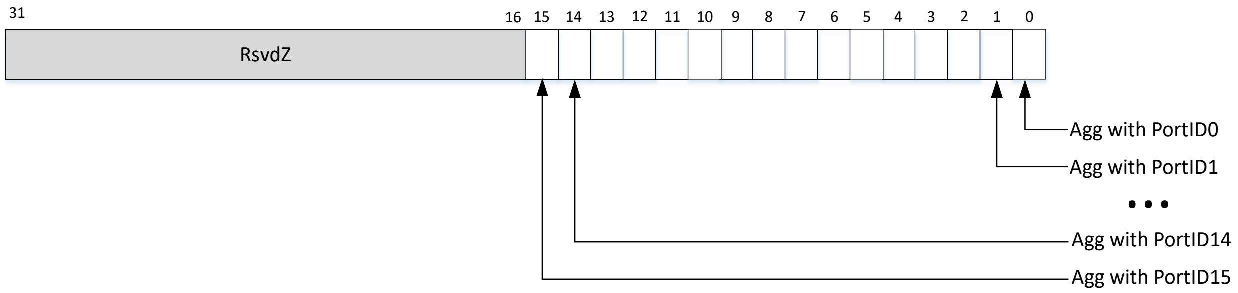


Figure 6-36: PortCapStat2 Register at Byte Offset-08h

Table 6-32 describes the PortCapStat2 Register fields at Byte Offset-08h.

5 **Table 6-32: PortCapStat2 Register Fields at Byte Offset-08h**

Bit Location	Register Description	Attributes
15:0	**PortAggVctr** PortAggVctr[0] to PortAggVctr[15] indicate the CCIX Port Aggregation candidates of this CCIX Port with CCIX PortID0 to CCIX PortID15, including this CCIX PortIDn. PortAggVctr must be 0000h if ComnCapStat2.PortAggCap is 0b. Only contiguous bits adjacent to PortAggVctr[n] can be 1b, i.e. only sequentially enumerated CCIX PortID can be candidates of an aggregation group. So if PortAggVctr[n] is 1b, then only contiguous PortAggVctr[n+1,n+2,etc.] and/or PortAggVctr[n-1,n-2,etc.] can be 1b. 0b: Indicates the CCIX PortID associated with this bit-position cannot be aggregated with this CCIX PortID. 1b: Indicates the CCIX PortID associated with this bit-position can be aggregated with this CCIX PortID. There isn't an explicit control register indicating which of the capable CCIX Port aggregation groups are being enabled. The control fields that implicitly identify the contiguous set of CCIX PortIDs that have been enabled as part of a CCIX Port Aggregation group are the BasePortID and SAMEntryAttr.NumAggPorts fields of a SAM Entry. The aggregated CCIX Ports range from BasePortID to BasePortID + (SAMEntryAttr.NumAggPorts - 1). These aggregated CCIX Ports must be consistent with contiguous bits in PortAggVctr.	RO
31:16	Reserved and Zero	RsvdZ

Figure 6-37 shows the layout of the CCIX Port Capabilities & Status 3 (PortCapStat3) Register at Byte Offset-0Ch.

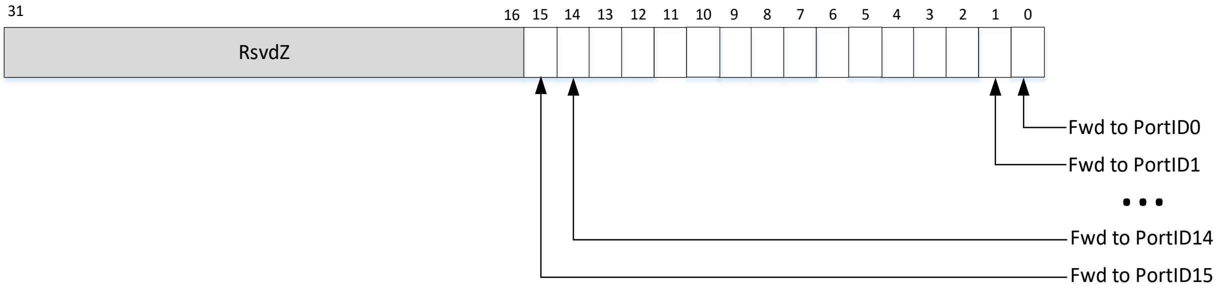


Figure 6-37: PortCapStat3 Register at Byte Offset-0Ch

Table 6-33 describes the PortCapStat3 Register fields at Byte Offset-0Ch.

Table 6-33: CCIX Port Capabilities&Status Register Fields at Byte Offset-0Ch

Bit Location	Register Description	Attributes
15:0	**PortFwdingVctr** PortFwdingVctr[0] to PortFwdingVctr[15] indicate the CCIX Port Forwarding Capability of this CCIX Port to CCIX PortID0 to CCIX PortID15, excluding this CCIX PortIDn. PortFwdingVctr must be 0000h if PortCapStat1.PortToPortFwdingCap is 0b. 0b: Indicates this CCIX Port cannot forward CCIX Packets to the CCIX PortID associated with this bit-position. 1b: Indicates this CCIX Port can forward CCIX Packets to the CCIX PortID associated with this bit-position. If PortFwdingVctr[x] is 1b in the CCIX Port Capability structure of CCIX PortIDy, then PortFwdingVctr[y] must be 1b in the CCIX Port Capability structure of CCIX PortIDx due to the bi-directional capability noted in the description of PortCapStat1.PortToPortFwdingCap in Table 6-31 PortFwdingVctr also describes the Forwarding Capability of this CCIX Port for Port-Aggregated traffic, if the ComnCapStat2.MultiHopPortAggCap field indicates this capability, described further in Table 6-13.	RO
31:16	Reserved and Zero	RsvdZ

The CCIX Port Capabilities & Status Error Log Pointer (PortErrLogPtr) Register at Byte Offset-10h contains the CCIX Port Error Log Offset which is described in CCIX RAS Overview. The remaining bits in this register are Reserved and Zero.

6.2.2.5.2 CCIX Port Control Register

Figure 6-38 shows the overall layout of the CCIX Port Control structure. Each CCIX Port has a SAM data structure, called the PSAM Table, used to resolve the output CCIX Link for CCIX packets output CCIX Link for CCIX packets. The PSAM Table contains the number of entries indicated in the PortCapStat1.NumPSAMEntryCap field of the CCIX Port Capabilities & Status data structure.

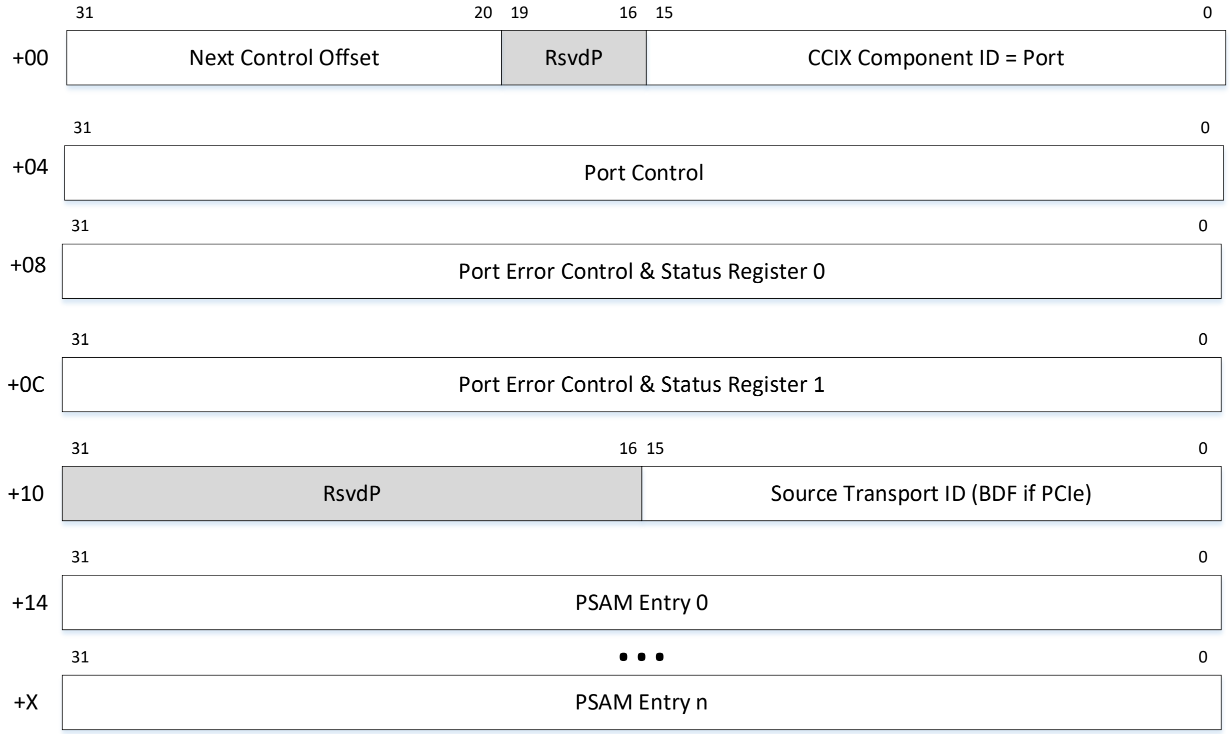


Figure 6-38: Layout of the CCIX Port Control Structure

Figure 6-39 shows the layout of the CCIX Port Control (PortCntl) Register at Byte Offset-04h.

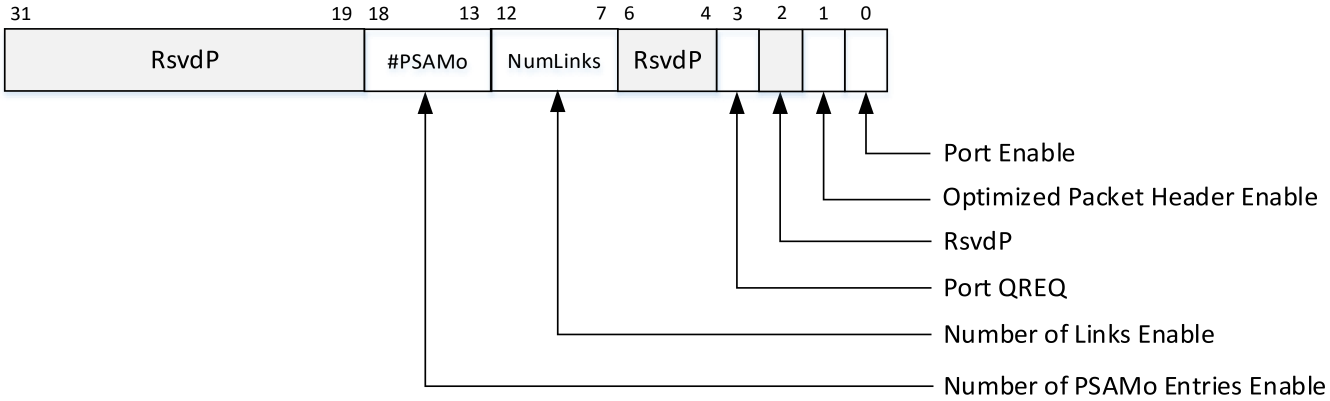


Figure 6-39: PortCntl Register at Byte Offset-04h

Table 6-34 describes the CCIX Port Control Register fields at Byte Offset-04h.

Table 6-34: PortCntl Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**PortEnable** This field controls enabling the CCIX Port. 0b: • Indicates that either the CCIX Port has not been configured, or the previously configured CCIX Port has been taken offline. 1b: • Indicates the CCIX Port is configured and enabled. This bit must not be set if CCIX Port Discovery Ready Status is not set. CCIX Device initializes to 0b after reset (except FLR). CCIX Port Enable control takes precedence over CCIX Link and CCIX Agent Enable control, i.e. CCIX traffic from all enabled CCIX Links and CCIX Agents to this CCIX Port must be disabled if the CCIX Port is disabled.	RW
1	**PktHdrTypeEnable** This field enables the CCIX Packet Header Type. 0b: • Indicates only PCIe Compatible Packet Header format enabled. 1b: • Indicates only CCIX Optimized Packet Header format is enabled. • The value of PktHderTypeEnable must be consistent with the value programmed in OptimizeTLPGenerationReceptionRoutingEnable as described in Table 6-74. CCIX Device initializes to 0b after reset (except FLR).	RW
2	Reserved and Preserved	RsvdP
3	**PortQREQ** Port Quiesce Request controls when the CCIX Port will act to achieve a quiesced state. There can only be a change in the value, 0b or 1b, of the corresponding PortCapStat1.PortQACK bit after CCIX configuration software changes the value, 0b or 1b, of this PortQREQ control bit. 0b: CCIX Port is not required to be quiesced. 1b: CCIX Port must be quiesced. CCIX Device initializes to 0b after reset (except FLR).	RW
6:4	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
12:7	**NumLinksEnable** This field indicates the number of unique CCIX Port-pairs that connect through this CCIX Port. However, additonal links may be enabled to accommodate RA to HA requests and HA to SA requests which are traveling in the same direction. See Chapter 3 for further details. 00h – 3Fh: Encoding for 1 to 63 CCIX Links enabled. CCIX Device initializes to 00h after reset (except FLR). There is no requirement for a contiguous set of CCIX Link Control Entries to be enabled, i.e. NumLinksEnable only reflects the total number of enabled CCIX Links.	RW
18:13	**NumPSAMEntryEnable** This field indicates the number of PSAM entries enabled. 00h: No PSAM entries are enabled. 01h – 3Fh: Encoding for enabling up to 63 PSAM entries. See Section 6.2.2.5.3 for a description of PSAM entries. CCIX Device initializes to 00h after reset (except FLR). There is no requirement for a contiguous set of PSAM Entries to be enabled, i.e. NumPSAMEntryEnable only reflects the total number of enabled PSAM Entries.	RW
31:19	Reserved and Preserved	RsvdP

The CCIX Port Error Control & Status Registers, PortErrCntlStat0 and PortErrCntlStat1, at Byte Offset-08h and Byte Offset-0Ch respectively, are described in the Port Control Register at Byte Offset-10h, the PortSrcIDCntl Register, contains the Source TransportID field, SrcTransportID[15:0]. PortSrcIDCntl.SrcTransportID is the BDF when the CCIX Transport Layer is PCIe. The remaining bits in this register are Reserved and Preserved.

The Port's PSAM Table, starting with PSAM Entry 0, is located at the Port Control structure at Byte Offset-14h.

6.2.2.5.3 PSAM Entry

Figure 6-40 shows the layout of a PSAM Entry.

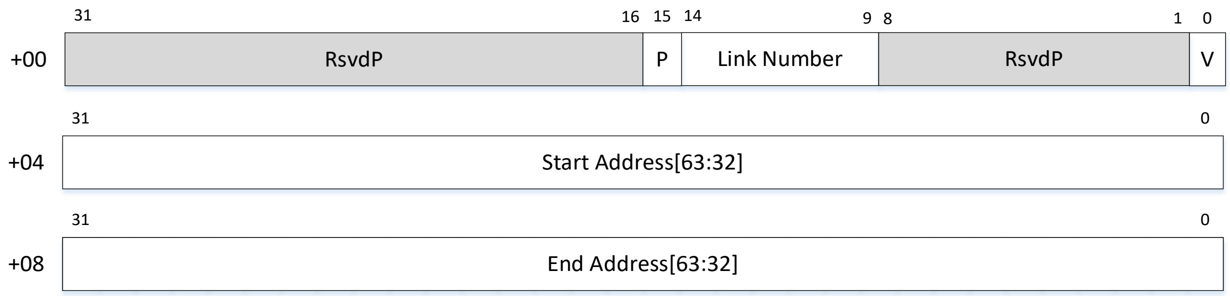


Figure 6-40: PSAM Entry

Table 6-35 describes the PSAM Register fields at Byte Offset-00h. PSAM Register fields at Byte Offset-04h and Byte Offset-08h have the same field descriptions as the field descriptions in Table 6-22 and Table 6-23 respectively.

Table 6-35: PSAM Register Fields at Byte Offset-00h

Bit Location	Register Description	Attributes
0	**PSAMEntryVal** This field indicates whether the PSAM Entry is valid. 0b: Indicates no CCIX Link is mapped in this entry. 1b: Indicates a CCIX Link is mapped in this entry.	RW
8:1	Reserved and Preserved	RsvdP
14:9	**LinkNum** Indicates the CCIX Port's Link Number mapped to the address range described in this PSAM entry. 00h – 3Fh: Encodings for up to 64 CCIX Link Numbers. CCIX Link Numbers are unique per CCIX Port (i.e. different CCIX Ports can re-use the CCIX Link Number space) and for each CCIX Port, CCIX Link Enumeration is required to start from 0, and be sequentially enumerated, without skipping a number. As described in the IDMEntry.AgentIDnLinkID field of Table 6-20 if CCIX Port Aggregation is enabled, the same LNUM must be setup on all aggregated CCIX Ports for traffic for the same CCIX AgentID.	RW
15	**PSAMEntryAddrType** 0b: Indicates the CCIX Port-Pair associated with the PSAMEntryAttr.LinkNum in this Entry is RA-to-HA and the Address mapped in this entry is part of the G-RSAM 1b: Indicates the CCIX Port-Pair associated with the PSAMEntryAttr.LinkNum in this Entry is HA-to-SA and the Address mapped in this entry is part of the G-HSAM.	RW
31:16	Reserved and Preserved	RsvdP

6.2.2.6 CCIX Link Structures

CCIX Link structures describe the CCIX Links associated with a particular CCIX Port, also illustrated in Figure 6-4.

6.2.2.6.1 CCIX Link Capabilities & Status Structure

Figure 6-41 shows the overall layout of CCIX Link Capabilities & Status structure. The Capabilities are common across the CCIX Links indicated in the NumLinksCap field in the CCIX Port Capabilities & Status structure and Status is common across the CCIX Links enabled, as indicated in the NumLinksEnable field in the CCIX Port Control structure.

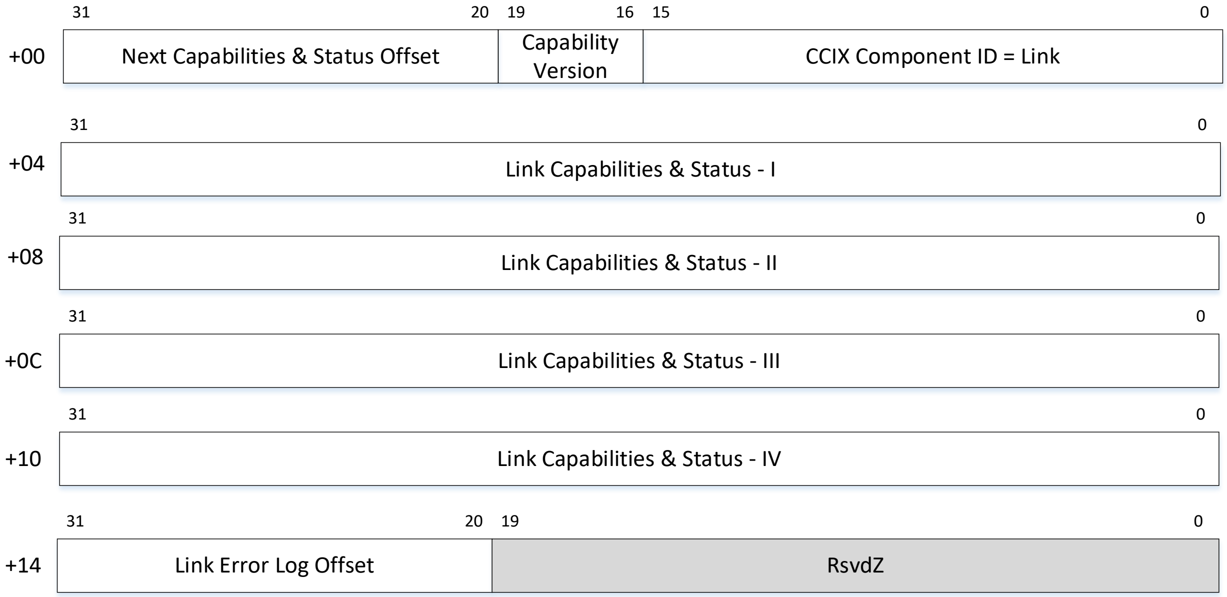

31
20
19
16
15
0
+00
Next Capabilities & Status Offset
Capability Version
CCIX Component ID = Link
31
0
+04
Link Capabilities & Status - I
31
0
+08
Link Capabilities & Status - II
31
0
+0C
Link Capabilities & Status - III
31
0
+10
Link Capabilities & Status - IV
31
20
19
0
+14
Link Error Log Offset
RsvdZ

Figure 6-41: CCIX Link Capabilities & Status Structure

Figure 6-42 shows the layout the CCIX Link Capabilities & Status (LinkCapStat) Register at Byte Offset-04h.

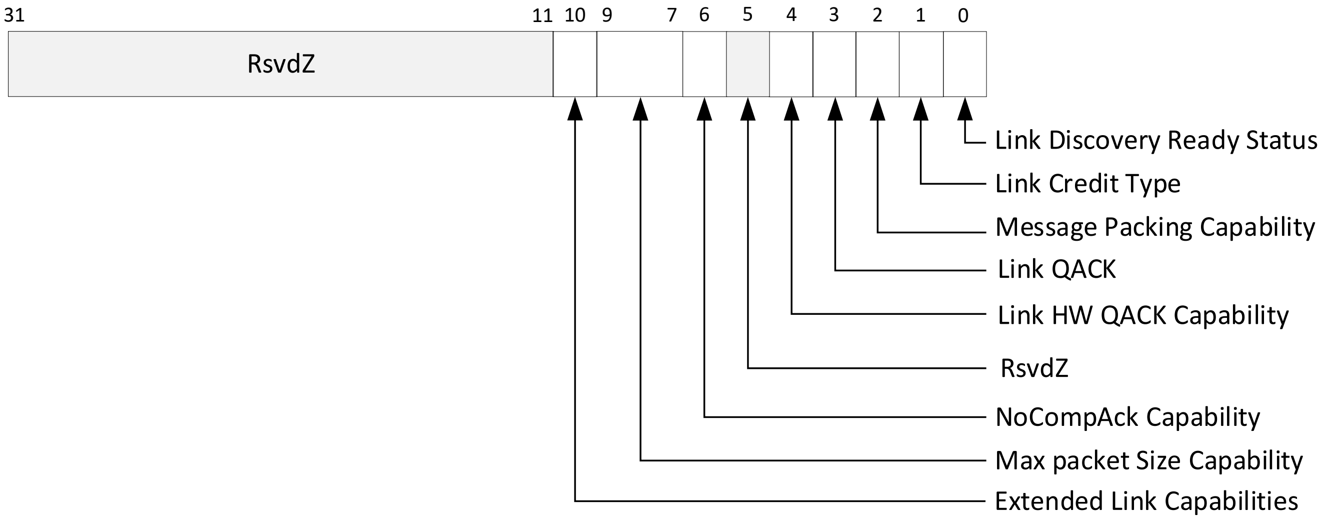

31
11
10
9
7
6
5
4
3
2
1
0
RsvdZ
Link Discovery Ready Status
Link Credit Type
Message Packing Capability
Link QACK
Link HW QACK Capability
RsvdZ
NoCompAck Capability
Max packet Size Capability
Extended Link Capabilities

Figure 6-42: CCIX Link Capabilities and Status Register at Byte Offset-04h

Table 6-36 describes the LinkCapStat Register fields at Byte Offset-04h.

Table 6-36: LinkCapStat Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**LinkDiscRdyStat** This field describes the CCIX Link's Discovery Readiness Status. 0b: • Indicates the CCIX Links and their capabilities and control are not ready to be discovered and configured. • Also indicates the CCIX Links are not ready to receive credit Grant Messages. 1b: • Indicates the CCIX Links and their capabilities and control are ready to be discovered and configured. • Also indicates the CCIX Links are ready to receive credit Grant Messages.	RO

Bit Location	Register Description	Attributes
1	**LinkCreditType** 0b: • Indicates fixed or static credits/Link capability, with the same credit capability indicating values per CCIX Link. • The values of the credit capability fields in Table 6-37, Table 6-38, and Table 6-39 are for each CCIX Link. 1b: • Indicates variable or dynamic credits/Link capability, with the credit capability indicating total value across all CCIX Links. • The values of the credit capability fields in Table 6-37, Table 6-38, and Figure 6-37 are total values across all CCIX Links.	RO
2	**MsgPackingCap** 0b: Indicates the CCIX Links do not have Message Packing capability. 1b: Indicates the CCIX Links have Message Packing capability.	RO

Bit Location	Register Description	Attributes
3	**LinkQACK** This field describes the CCIX Link's Quiesce Acknowledgement status. 0b: CCIX Link not quiesced. 1b: CCIX Link quiesced. If the CCIX Link has Hardware Quiesce Acknowledgement (HW QACK) capability, as indicated by a LinkCapStat.LinkHWQACKCap value of 1b, LinkCapStat.LinkQACK is set by implementation specific methods. Software can choose to poll LinkCapStat.LinkQACK instead of waiting for the <Link Quiesce Time> value to check LinkQACK if LinkCapStat.LinkHWQACKCap has a value of 1b. If the CCIX Link does not have HW QACK capability as indicated in LinkCapStat.LinkHWQACKCap value of 0b, the following sequence is followed: The CCIX Link sets the LinkQACK bit to a value 1b after completing or detecting the following actions in this order: 1. The Link detects that the Link Quiesce Request (LinkCntl.LinkQREQ) Control bit is set. 2. The Link has not issued any Requests or received any Responses, other than CreditReturn Messages, for the duration of <Link Quiesce Time>. <Link Quiesce Time> is based on the value of ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue, described further in Table 6-12. 3. The Link then returns all outstanding credits to the Link partner if it hasn't already done so. 4. The Link detects all outstanding credits from the Link partner have been returned and also waits for the duration of <Link Quiesce Time> to detect return of all credits. 5. Following the transition of the Link Quiesce Request (LinkCntl.LinkQREQ in Table 6-40) control bit from 0b to 1b, the Link must take no longer than 3 * <Link Quiesce Time> to set LinkQACK. After 4 * <Link Quiesce Time>, CCIX Software detecting a zero returned for the LinkQACK field indicates an error condition such that the Link was unable to reach a quiescent state. Following the transition of the LinkCntl.LinkQREQ control bit from 1b to 0b, the CCIX Link must transition the LinkQACK bit from 1b to 0b.	RO
4	**LinkHWQACKCap** This field describes the CCIX Link's Hardware Quiesce Acknowledgement Capability. 0b: The CCIX Link does not have a hardware mechanism to achieve a quiescent state. 1b: The CCIX Link has a hardware mechanism to achieve a quiescent state.	RO
5	Reserved and Zero	RsvdZ
6	**NoCompAckCap** 0b: Indicates the CCIX Links require receiving Completion Acks. 1b: Indicates the CCIX Links have the capability to not receive Completion Acks.	RO

Bit Location	Register Description	Attributes
9:7	**MaxPktSizeCap** 000b: 128B Max Packet Size capability. 001b: 256B Max Packet Size capability. 010b: 512B Max Packet Size capability. All other encodings: Reserved. A chip that declares 128B Max Cacheline Size Capability, i.e. a ComnCapStat2.CachelineSizeCap value of 1b, is not permitted to declare a MaxPktSizeCap value of 000b.	RO
10	Extended Link Capabilities 0: Link capabilities registers are not extended. 1: Link capabilities registers are extended.	
31:11	Reserved and Zero	RsvdZ

Figure 6-43 shows the layout the CCIX Link Send Capability (LinkSendCap) Register at Byte Offset-08h.

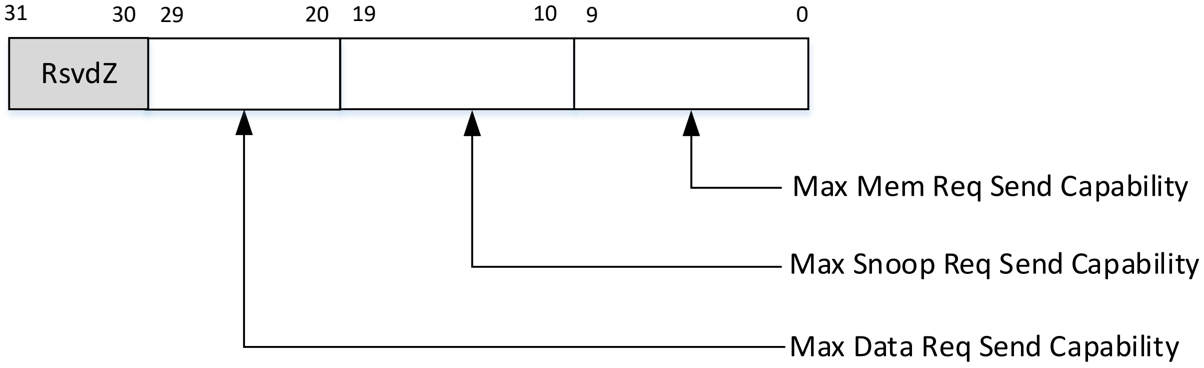

31
30
29
20
19
10
9
0
RsvdZ
Max Mem Req Send Capability
Max Snoop Req Send Capability
Max Data Req Send Capability

Figure 6-43: LinkSendCap Register at Byte Offset-08h

Table 6-37 describes the LinkSendCap Register fields at Byte Offset-08h.

5

Table 6-37: LinkSendCap Register Fields at Byte Offset-08h

Bit Location	Register Description	Attributes
9:0	**MaxMemReqSendCap** This field describes the maximum number of outstanding Memory requests that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of sending. CCIX Configuration Software is not required to allocate the credits that match this send capability since there are other factors that influence the allocated credits including the destination CCIX Link's receive capability. Encodings 0 to 1023 indicate send capability of 0 to 1023 Memory Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port. Encoding 000h indicates there are no RAs or HAs on this CCIX Device and hence is not capable of sending Memory requests.	RO
19:10	**MaxSnpReqSendCap** This field describes the maximum number of outstanding Snoops that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of sending. CCIX Configuration Software is not required to allocate the credits that match this send capability since there are other factors that influence the allocated credits including the destination CCIX Link's receive capability. Encodings 0 to 1023 indicate send capability of 0 to 1023 Snoop Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port. Encoding 000h indicates there are no HAs on this CCIX Device and hence is not capable of sending Snoop requests.	RO
29:20	**MaxDatReqSendCap** This field describes the maximum number of outstanding data packets that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of sending. CCIX Configuration Software is not required to allocate the credits that match this send capability since there are other factors that influence the allocated credits including the destination CCIX Link's receive capability. Encodings 0 to 1023 indicate send capability of 0 to 1023 Data Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port.	RO
31:30	Reserved and Zero	RsvdZ

Figure 6-44 shows the layout of the CCIX Link Capabilities & Status (LinkRcvCAp) Register at Byte Offset-0Ch.

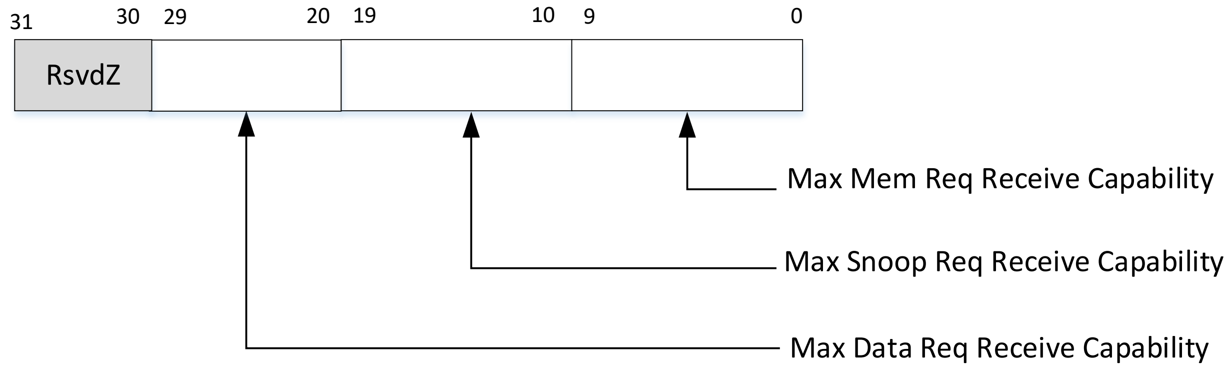

31
30
29
20
19
10
9
0
RsvdZ
Max Mem Req Receive Capability
Max Snoop Req Receive Capability
Max Data Req Receive Capability

Figure 6-44: LinkRcvCap Register at Byte Offset-0Ch

Table 6-38 describes the LinkRcvCap Register fields at Byte Offset-0Ch.

Table 6-38: LinkRcvCap Register Fields at Byte Offset-0Ch

Bit Location	Register Description	Attributes
9:0	**MaxMemReqRcvCap** This field describes the maximum number of outstanding Memory requests that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of receiving. Thus this field also indicates the maximum number of Memory request send credits that can be allocated to all destination CCIX Links (located at the Destination TransportIDs). Encodings 0 to 1023 indicate receive capability of 0 to 1023 Memory Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port. Encoding 000h indicates there are no SAs or HAs on this CCIX Device and hence is not capable of receving Memory requests.	RO

Bit Location	Register Description	Attributes
19:10	**MaxSnpReqRcvCap** This field describes the maximum number of outstanding Snoops that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of receiving. Thus this field also indicates the maximum number of Snoop send credits that can be allocated to all destination CCIX Links (located at the Destination TransportIDs). Encodings 0 to 1023 indicate receive capability of 0 to 1023 Snoop Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port. Encoding 000h indicates there are no RAs on this CCIX Device and hence is not capable of receiving Snoop requests.	RO
29:20	**MaxDatReqRcvCap** This field describes the maximum number of outstanding data request packets that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of receiving. Thus this field also indicates the maximum number of data send credits that can be allocated to all destination CCIX Links (located at the Destination TransportIDs). Encodings 0 to 1023 indicate receive capability of 0 to 1023 Data Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port.	RO
31:30	Reserved and Zero	RsvdZ

Figure 6-45 shows the layout of the CCIX Link Credited Miscellaneous Message Capabilities (LinkCreditMiscMsgCap) Register at Byte Offset-10h.

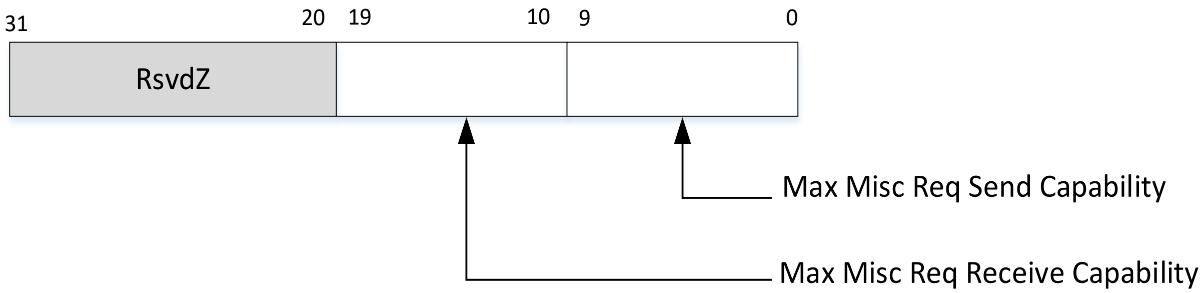


Figure 6-45: LinkCreditMiscMsgCap Register at Byte Offset-10h

Table 6-39 describes the LinkCreditMiscMsgCap Register fields at Byte Offset-10h.

Table 6-39: LinkCreditMiscMsgCap Register Fields at Byte Offset-10h.

Bit Location	Register Description	Attributes
9:0	**MaxMiscReqSendCap** This field describes the maximum number of outstanding credited Misc requests that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of sending. CCIX Configuration Software is not required to allocate the credits that match this send capability since there are other factors that influence the allocated credits including the destination CCIX Link's receive capability. Encodings 0 to 1023 indicate send capability of 0 to 1023 Memory Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port. Encoding 000h indicates this CCIX Device is not capable of sending credited Misc requests.	RO
19:10	**MaxMiscReqRcvCap** This field describes the maximum number of outstanding credited Misc requests that each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port are capable of receiving. Thus this field also indicates the maximum number of credited Misc send credits that can be allocated to all destination CCIX Links (located at the Destination TransportIDs). Encodings 0 to 1023 indicate receive capability of 0 to 1023 Misc Request Credits across each CCIX Link (if LinkCapStat.LinkCreditType is 0b) or all CCIX Links (if LinkCapStat.LinkCreditType is 1b) on this CCIX Port. Because the receipt of Credited Misc Requests is optional, encoding 000h indicates the CCIX Link is not capable of receiving Credited Misc Requests.	RO
31:20	Reserved and Zero	RsvdZ

The CCIX Link Capabilities & Status Error Log Pointer (LinkErrLogPtr) Register at Byte Offset-14h contains the CCIX Link Error Log Offset which is described in the CCIX RAS Overview. The remaining bits in this register are Reserved and Zero.

6.2.2.6.2 CCIX Link Control Structure

Figure 6-46 shows the overall layout of CCIX Link Control structure. The PortCapStat1.NumLinksCap field, described in Table 6-31, indicates the number of CCIX Link Control entries, whereas the PortCntl.NumLinksEnable field, described in Table 6-34, indicates the number of enabled CCIX Link Control entries.

The number of CCIX Link Control entries enabled in the CCIX Link Control structure for a given CCIX Port must equal the number of unique CCIX Port-pairs that connect through that CCIX Port. When multiple CCIX Agent-types share the same CCIX AgentID name space on a CCIX Device, and those CCIX Agents connect through a given CCIX Port, each CCIX Agent shares the same CCIX Link Control Entry on that CCIX Port.

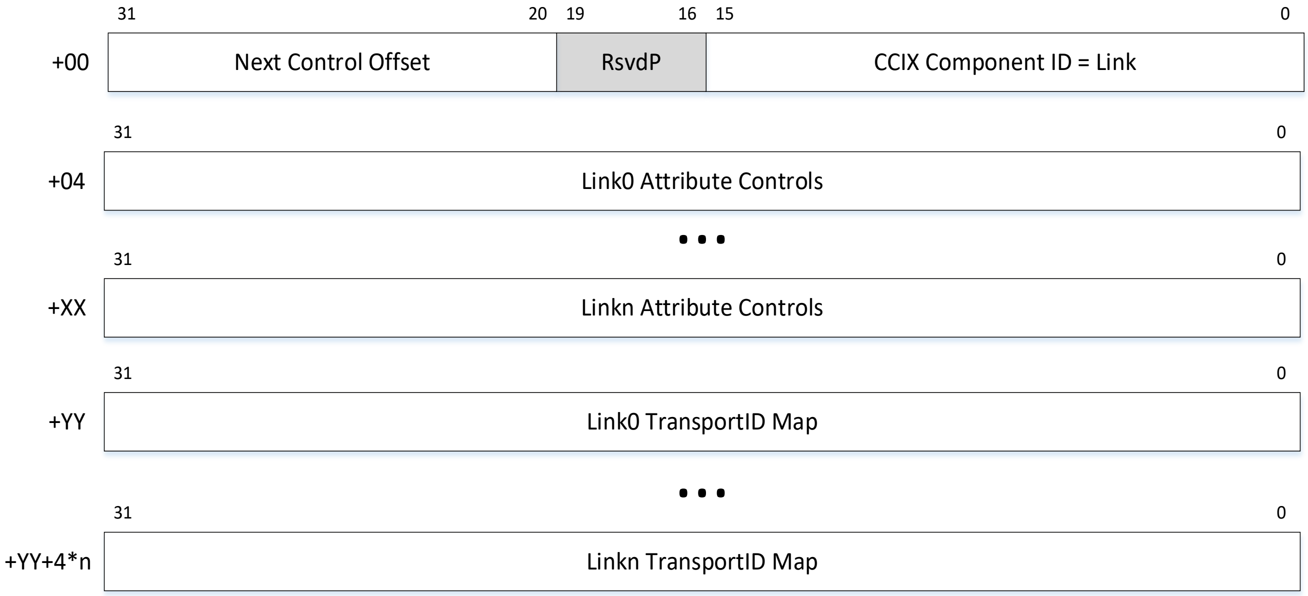


Figure 6-46: CCIX Link Control structure

6.2.2.6.2.1 CCIX Link Attribute Control Structure

Figure 6-47 shows the overall layout of the CCIX Link Attribute Control entries. The CCIX Link Attribute Control entries are setup uniquely per CCIX Link and are arranged in CCIX Link number order, starting with the entry for CCIX Link #0 and ending with CCIX Link #(PortCapStat1.NumLinksCap – 1).

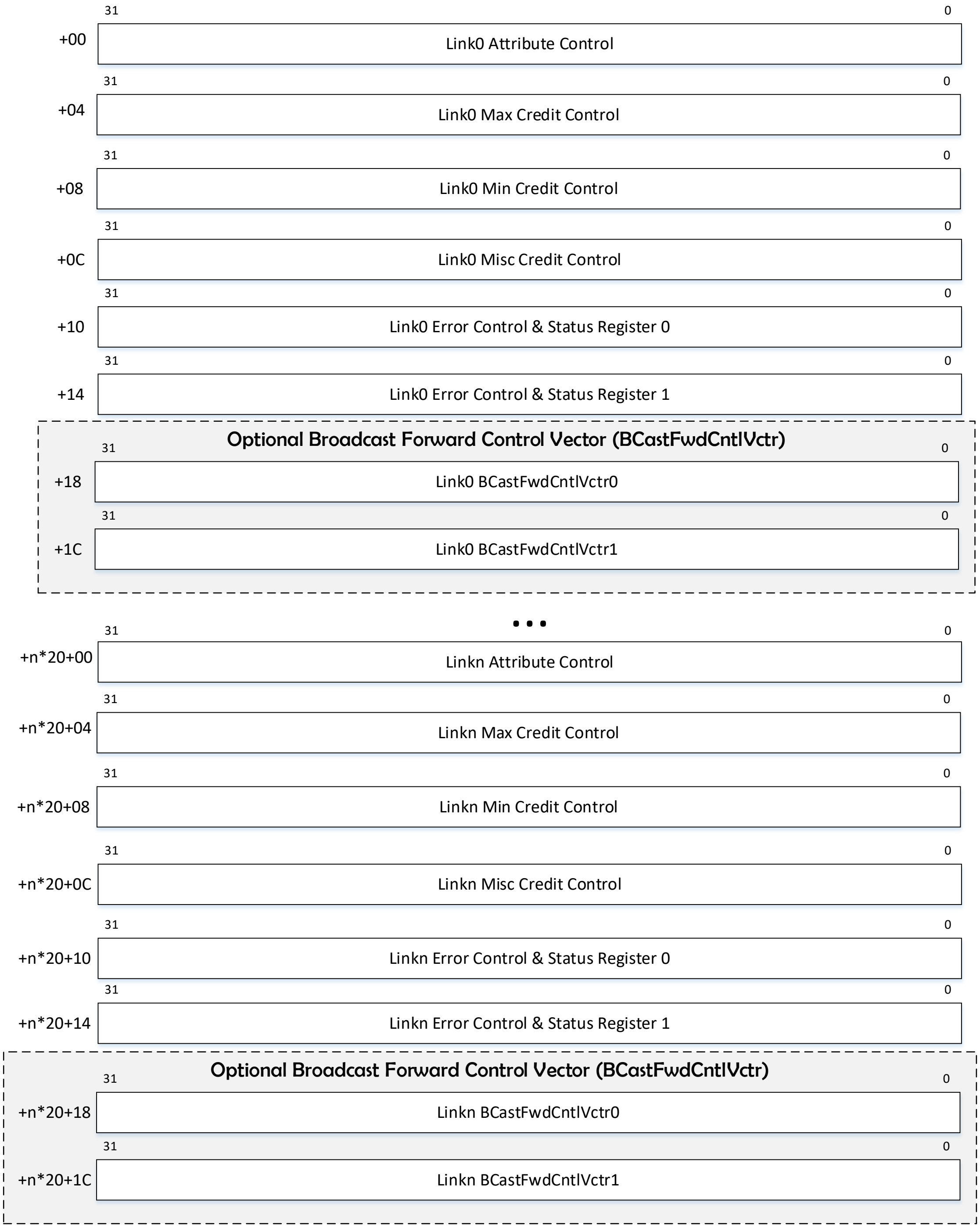


Figure 6-47: CCIX Link Attribute Control Entries

Figure 6-48 shows the layout of the Extended CCIX Link Attribute Control entries.

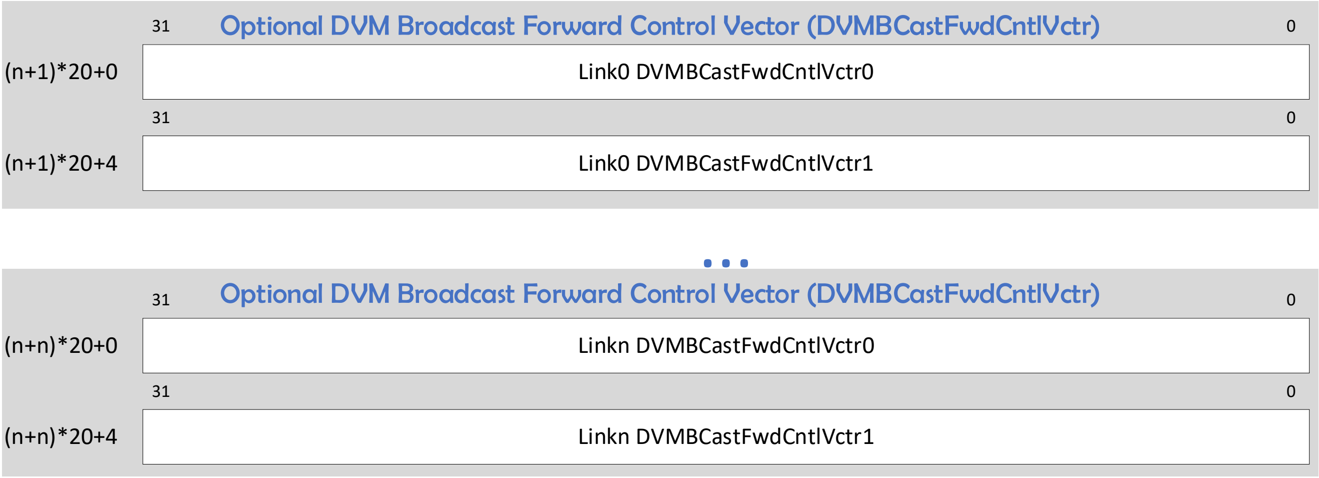

31
Optional DVM Broadcast Forward Control Vector (DVMBCastFwdCntlVctr)
0
(n+1)*20+0
Link0 DVMBCastFwdCntlVctr0
31
0
(n+1)*20+4
Link0 DVMBCastFwdCntlVctr1
• • •
31
Optional DVM Broadcast Forward Control Vector (DVMBCastFwdCntlVctr)
0
(n+n)*20+0
Linkn DVMBCastFwdCntlVctr0
31
0
(n+n)*20+4
Linkn DVMBCastFwdCntlVctr1

Figure 6-48: Extended CCIX Link Attribute Control Structure

Figure 6-49 shows the layout of the CCIX Link Attribute Control (LinkAttrCntl) Entry at Byte Offset-00h.

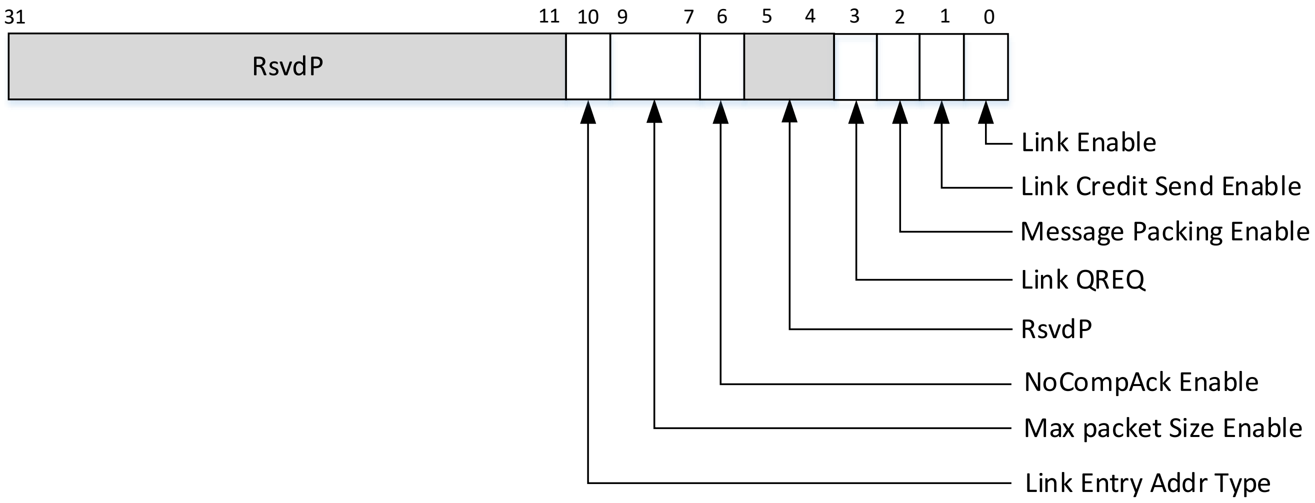


Figure 6-49: LinkAttrCntl Entry at Byte Offset-00h

Table 6-40 describes the LinkAttrCntl Entry fields at Byte Offset-00h.

Table 6-40: LinkAttrCntl Entry Fields at Byte Offset-00h

Bit Location	Register Description	Attributes
0	**LinkEnable** This field controls enabling the CCIX Link. 0b: • Indicates that either the CCIX Link has not been configured, or the previously configured CCIX Link has been taken offline. • Also indicates the CCIX Link must clear all previously granted credits from the destination TransportID. 1b: • Indicates the CCIX Link is configured and enabled. This bit must not be set if CCIX Link Discovery Ready Status is not set. • Also indicates the CCIX Link must receive credit Messages. CCIX Device initializes to 0b after reset (except FLR).	RW
1	**LinkCreditSendEnable** This field controls enabling the CCIX Link to send Credit Messages. 0b: • Indicates the CCIX Link must not send credit Grant Messages to the Destination TransportID. 1b: • Indicates the CCIX Links can send credit Grant Messages to the Destination TransportID. The LinkEnable field of the Link at the Destination TransportID must be 1b prior to a 0b-to-1b transition of LinkCreditSendEnable. • If the LinkCapStat.LinkCreditType indicated is 0b, the number of initial credits sent following a 0b-to-1b transition of LinkCreditSendEnable is determined by the Maximum Request Receive Capability values, described in Table 6-38 and Table 6-39. • If the LinkCapStat.LinkCreditType indicated is 1b, the number of initial credits sent following a 0b-to-1b transition of LinkCreditSendEnable is determined by the Reserved Request Credit Enable values, described in Table 6-42 and Table 6-43. CCIX Device initializes to 0b after reset (except FLR).	RW
2	**MsgPackingEnable** 0b: Indicates Message Packing is not enabled. 1b: Indicates Message Packing is enabled. CCIX Device initializes to 0b after reset (except FLR).	RW

Bit Location	Register Description	Attributes
3	**LinkQREQ** Link Quiesce Request controls when the CCIX Link will act to achieve a quiesced state. There can only be a change in the value, 0b or 1b, of the corresponding LinkCapStat.LinkQACK bit after CCIX configuration software changes the value, 0b or 1b, of this LinkQREQ control bit. 0b: CCIX Link is not required to be quiesced. 1b: CCIX Link must be quiesced. CCIX Device initializes to 0b after reset (except FLR).	RW
5:4	Reserved and Preserved	RsvdP
6	**NoCompAckEnable** 0b: Indicates the CCIX Agents on the CCIX Link are required to send CompAck response where relevant. 1b: Indicates the CCIX Agents on the CCIX Link must not send CompAck response where relevant. CCIX Device initializes to 0b after reset (except FLR).	RW
9:7	**MaxPktSizeEnable** 000b: 128B Max Packet Size enabled. 001b: 256B Max Packet Size enabled. 010b: 512B Max Packet Size enabled. All other encodings: Reserved. If a CCIX Device has 128B Cacheline Size enabled, i.e. a ComnCntl2.CachelineSizeEnable value of 1b, a MaxPktSizeEnable value of 000b is not permitted, i.e. the Max Packet Size enabled must be programmed to 256B or greater. CCIX Device initializes to 000b after reset (except FLR).	RW
10	**LinkEntryAddrType** 0b: • Indicates RSAM Type Address Reference for inbound RA-to-HA Request traffic for the CCIX Port-Pair associated with this CCIX Link Entry. 1b: • Indicates HSAM Type Address Reference for inbound HA-to-SA Request traffic for the CCIX Port-Pair associated with this CCIX Link Entry. CCIX Device initializes to 0b after reset (except FLR). The LinkEntryAddrType encodings to indicate inbound RA-to-HA Request traffic vs. inbound HA-to-SA traffic are mutually exclusive, 0b vs. 1b, for a Link. This is because these two traffic types cannot share the same Link in the same (inbound) direction. CCIX Devices, therefore, must have separate Links for RA-to-HA Request traffic and HA-to-SA traffic in the same direction, on the same CCIX Port.	RW
31:11	Reserved and Preserved	RsvdP

Figure 6-50 shows the layout of the CCIX Link Maximum Credit Control (LinkMaxCreditCntl) Entry at Byte Offset-04h.

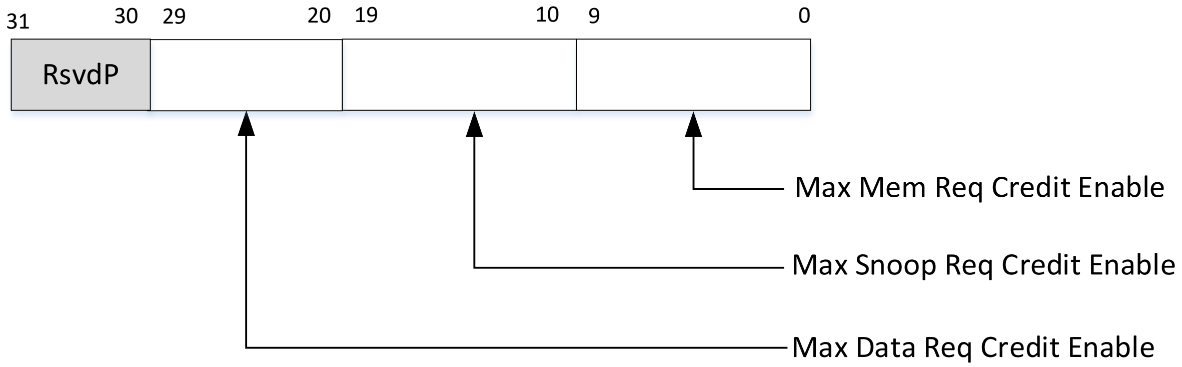

31
30
29
20
19
10
9
0
RsvdP
Max Mem Req Credit Enable
Max Snoop Req Credit Enable
Max Data Req Credit Enable

Figure 6-50: LinkMaxCreditCntl Entry at Byte Offset-04h

Table 6-41 describes the LinkMaxCreditCntl Entry fields at Byte Offset-04h.

Table 6-41: LinkMaxCreditCntl Entry Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
9:0	**MaxMemReqCreditEnable** This field describes the maximum number of outstanding Memory request credits that this CCIX Link is enabled to send. Thus this field also indicates the maximum number of outstanding Memory requests that the destination CCIX Link (located at the Destination TransportID) can send. Encodings 0 to 1023 indicate 0 to 1023 Memory Request Credits are enabled. The CCIX Device at the destination CCIX Link is not required to issue outstanding Memory Requests that use up the maximum Memory Request credits it has been sent on this CCIX Link. CCIX Configuration Software can set this field to encoding 000h when there are no RAs or HAs on the CCIX Device at the destination CCIX Link and hence the destination CCIX Link is not capable of sending Memory requests. CCIX Device initializes to 000h after reset (except FLR).	RW

Bit Location	Register Description	Attributes
19:10	**MaxSnpReqCreditEnable** This field describes the maximum number of outstanding Snoop Request credits that this CCIX Link is enabled to send. Thus this field also indicates the maximum number of outstanding Snoops that the destination CCIX Link (located at the Destination TransportID) can send. Encodings 0 to 1023 indicate 0 to 1023 Snoop Request Credits are enabled. The CCIX Device is not required to consume the maximum Snoop Request credits it has been enabled with on this CCIX Link. CCIX Configuration Software can set this field to encoding 000h when there are no HAs on the CCIX Device at the destination CCIX Link and hence the destination CCIX Link is not capable of sending Snoop requests. CCIX Device initializes to 000h after reset (except FLR).	RW
29:20	**MaxDatReqCreditEnable** This field describes the maximum number of outstanding data credits that this CCIX Link is enabled to send. Thus this field also indicates the maximum number of outstanding Data packets that the destination CCIX Link (located at the Destination TransportID) can send. Encodings 0 to 1023 indicate 0 to 1023 Data Request Credits are enabled. The CCIX Device is not required to consume the maximum Data Request credits it has been enabled with on this CCIX Link. CCIX Device initializes to 000h after reset (except FLR).	RW
31:30	Reserved and Preserved	RsvdP

Figure 6-51 shows the layout of the CCIX Link Minimum Credit Control (LinkMinCreditCntl) Entry at Byte Offset-08h.

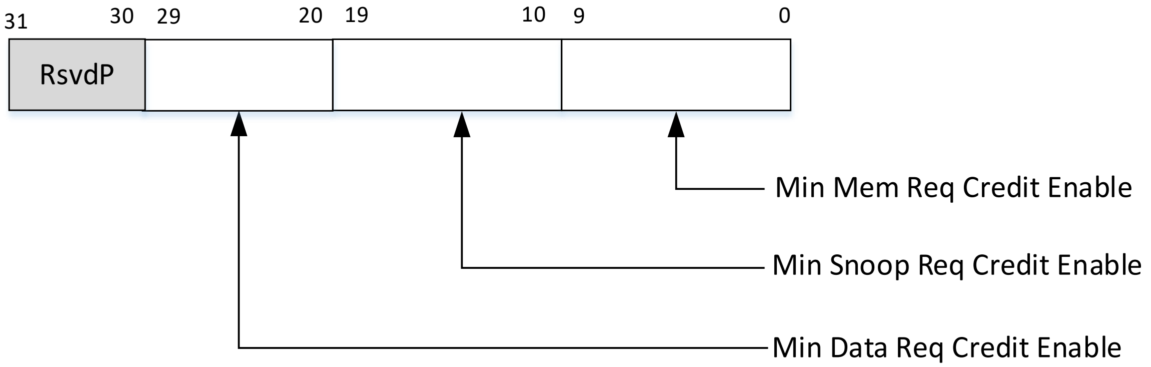

31
30
29
20
19
10
9
0
RsvdP
Min Mem Req Credit Enable
Min Snoop Req Credit Enable
Min Data Req Credit Enable

Figure 6-51: LinkMinCreditCntl Entry at Byte Offset-08h

Table 6-42 describes the LinkMinCreditCntl Entry fields at Byte Offset-08h.

Table 6-42: LinkMinCreditCntl Entry Fields at Byte Offset-08h

Bit Location	Register Description	Attributes
9:0	**MinMemReqCreditEnable** This field describes the minimum Memory Request credits that are reserved for the CCIX Link-partner and that this CCIX Link is enabled to send. Encodings 0 to 1023 indicate 0 to 1023 Memory Request Credits are enabled. The CCIX Device at the destination CCIX Link is not required to issue outstanding Memory Requests that use up the minimum Memory Request credits it has been sent on this CCIX Link. CCIX Device initializes to 000h after reset (except FLR). CCIX Configuration Software can set this field to encoding 000h when there are no RAs or HAs on the CCIX Device at the destination CCIX Link and hence the destination CCIX Link is not capable of sending Memory requests. The sum of the Minimum Request Credit Enable Values set per CCIX Link must factor in the maximum receive capability across all CCIX Links on this CCIX Port, i.e. ∑ (MinMemReqCreditEnable/Link) ≤ MaxMemReqRcvCap.	RW

Bit Location	Register Description	Attributes
19:10	**MinSnpReqCreditEnable** This field describes the minimum Snoop Request credits that are reserved for the CCIX Link-partner and that this CCIX Link is enabled to send. Encodings 0 to 1023 indicate 0 to 1023 Snoop Request Credits are enabled. The CCIX Device is not required to consume the minimum Snoop Request credits it has been sent on this CCIX Link. CCIX Configuration Software can set this field to encoding 000h when there are no HAs on the CCIX Device at the destination CCIX Link and hence the destination CCIX Link is not capable of sending Snoop requests. CCIX Device initializes to 000h after reset (except FLR). The sum of the Minimum Request Credit Enable Values set per CCIX Link must factor in the maximum receive capability across all CCIX Links on this CCIX Port, i.e. ∑ (MinSnpReqCreditEnable/Link) ≤ MaxSnpReqRcvCap.	RW
29:20	**MinDatReqCreditEnable** This field describes the minimum Data Request credits that are reserved for the CCIX Link-partner and that this CCIX Link is enabled to send. Encodings 0 to 1023 indicate 0 to 1023 Data Request Credits are enabled. The CCIX Device is not required to consume the minimum Data Request credits it has been sent on this CCIX Link. CCIX Device initializes to 000h after reset (except FLR). The sum of the Minimum Request Credit Enable Values set per CCIX Link must factor in the maximum receive capability across all CCIX Links on this CCIX Port, i.e. ∑ (MinDatReqCreditEnable/Link) ≤ MaxDatReqRcvCap.	RW
31:30	Reserved and Preserved	RsvdP

5 Figure 6-52 shows the layout of the CCIX Link Miscellaneous Credit Control (LinkMiscCreditCntl) Entry at Byte Offset-0Ch.

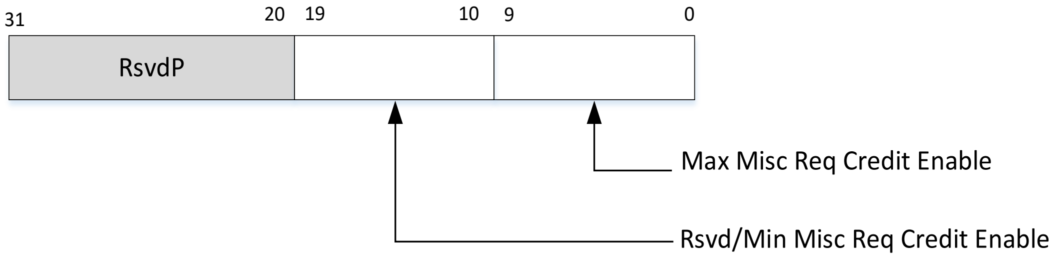


Figure 6-52: LinkMiscCreditCntl Entry at Byte Offset-0Ch

Table 6-43 describes the LinkMiscCreditCntl Entry fields at Byte Offset-0Ch.

Table 6-43: LinkMiscCreditCntl Entry at Byte Offset-0Ch

Bit Location	Register Description	Attributes
9:0	**MaxMiscReqCreditEnable** This field describes the maximum number of outstanding Credited Misc credits that this CCIX Link is enabled to send. Thus this field also indicates the maximum number of outstanding Misc requests that the destination CCIX Link (located at the Destination TransportID) can send. Encodings 0 to 1023 indicate 0 to 1023 Credited Misc Request Credits are enabled. CCIX Configuration Software can set this field to encoding 000h when the destination CCIX Link is not capable of sending Credited Misc requests. The CCIX Device is not required to consume the maximum Credited Misc Request credits it has been enabled with on this CCIX Link. CCIX Device initializes to 000h after reset (except FLR).	RW
19:10	**MinMiscReqCreditEnable** This field describes the minimum Credited Misc Request credits that are reserved for the CCIX Link-partner and that this CCIX Link is enabled to send. Encodings 0 to 1023 indicate 0 to 1023 Credited Misc Request Credits are enabled. The CCIX Device is not required to consume the minimum Credited Misc Request credits it has been sent on this CCIX Link. CCIX Configuration Software can set this field to encoding 000h when the destination CCIX Link is not capable of sending Credited Misc requests. CCIX Device initializes to 000h after reset (except FLR). The sum of the Minimum Request Credit Enable Values set per CCIX Link must factor in the maximum receive capability across all CCIX Links on this CCIX Port, i.e. ∑ (MinMiscReqCreditEnable/Link) ≤ MaxMiscReqRcvCap.	RW
31:20	Reserved and Preserved	RsvdP

The CCIX Link Error Control & Status 0 (LinkErrCntlStat0) and Link Error Control & Status 1 (LinkErrCntlStat1) Registers, at Byte Offset-10h and Byte Offset-14h respectively, are described in CCIX RAS Overview.

6.2.2.6.2.1.1 Broadcast Forward Control Vector

The Broadcast Forward Control Vector (BCastFwdCntlVctr) is an optional structure, present only when CCIX Port-to-Port forwarding capability exists, i.e. a PortCapStat1.PortToPortFwdingCap value of 1b is indicated.

Figure 6-53 shows the layout of Broadcast Forward Control Vector0 (BCastFwdCntlVctr0) Register. The vector is Home AgentID indexed from HAID0 to HAID31.

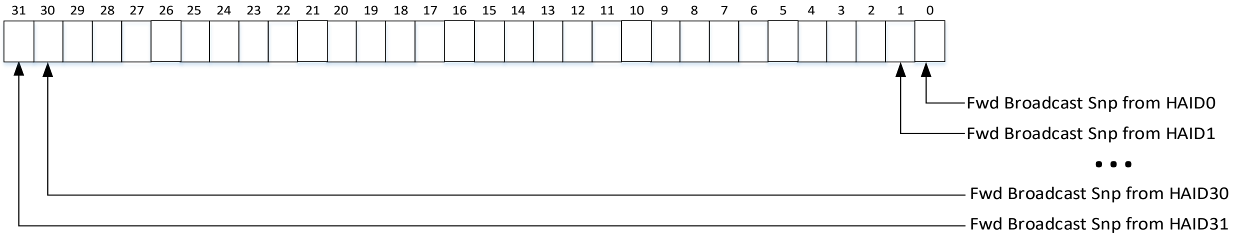

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Fwd Broadcast Snp from HAID0
Fwd Broadcast Snp from HAID1
• • •
Fwd Broadcast Snp from HAID30
Fwd Broadcast Snp from HAID31

Figure 6-53: BCastFwdCntlVctr0

Table 6-44 describes the fields of the BCastFwdCntlVctr0 Register.

Table 6-44: BCastFwdCntlVctr0 Register Fields

Bit Location	Register Description	Attributes
31:0	**BCastFwdCntlVctr0** BCastFwdCntlVctr0[0] to BCastFwdCntlVctr0[31] indicate the Broadcast Snoop Forward Controls for HAID0 to HAID31 respectively, for this CCIX Link: 0b: Indicates that Broadcast Snoops from the HAID associated with this bit-position must not be forwarded on this CCIX Link. 1b: Indicates that Broadcast Snoops from the HAID associated with this bit-position must be forwarded on this CCIX Link. CCIX Device initializes to 0000h after reset (except FLR).	RW

Figure 6-54 shows the layout of the Broadcast Forward Control Vector1 (BCastFwdCntlVctr1) Register. The vector is Home AgentID indexed from HAID32 to HAID63.

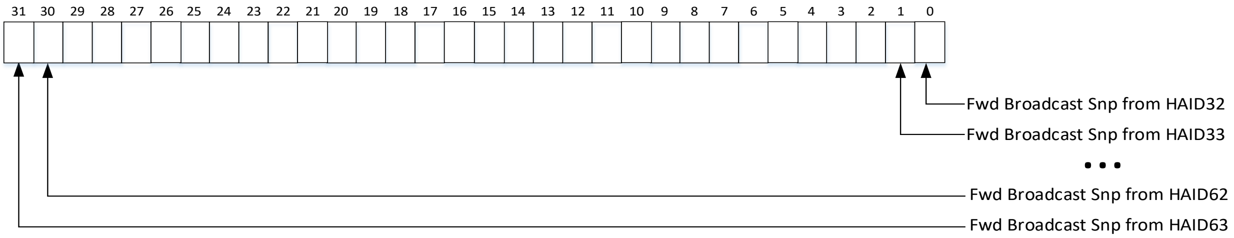

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Fwd Broadcast Snp from HAID32
Fwd Broadcast Snp from HAID33
• • •
Fwd Broadcast Snp from HAID62
Fwd Broadcast Snp from HAID63

Figure 6-54: BCastFwdCntlVctr1

Table 6-45 describes the fields of the BCastFwdCntlVctr1 Register.

Table 6-45: BCastFwdCntlVctr1 Register Fields

Bit Location	Register Description	Attributes
31:0	**BCastFwdCntlVctr1** BCastFwdCntlVctr1[0] to BCastFwdCntlVctr1[31] indicate the Broadcast Snoop Forward Controls for HAID32 to HAID63 respectively, for this CCIX Link: 0b: Indicates that Broadcast Snoops from the HAID associated with this bit-position must not be forwarded on this CCIX Link. 1b: Indicates that Broadcast Snoops from the HAID associated with this bit-position must be forwarded on this CCIX Link. CCIX Device initializes to 0000h after reset (except FLR).	RW

6.2.2.6.2.1.2 **DVM Broadcast Forward Control Vector**

The DVM Broadcast Forward Control Vector (DVMBCastFwdCntlVctr) is an optional structure, present only when DVM transactions forward capability exists, i.e. ComnCapStat2.DVMPortToPortForwardingCap value of 1b is indicated.

Figure 6-55 shows the layout of DVM Broadcast Forward Control Vector0 (DVMBCastFwdCntlVctr0) Register. The vector is Request or DVM Agent indexed from 0 to 31.

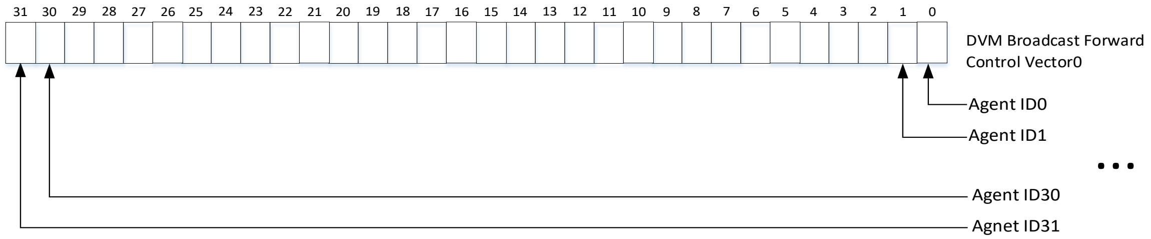


Figure 6-55: DVMBCASTFwdCntlVctr0 Register

Table 6-46 describes the fields of the DVMBCastFwdCntlVctr0 Register.

Table 6-46: DVMBCastFwdCntlVctr0 Register Fields

Bit Location	Register Description	Attributes
31:0	**DVMBCastFwdCntlVctr0** DVMBCastFwdCntlVctr0[0] to DVMBCastFwdCntlVctr0[31] indicate the Broadcast DVM request Forward Controls for RAID0/DAID0 to RAID31/DAID31 respectively, for this CCIX Link: 0b: Indicates that Broadcast DVM request from the RAID/DAID associated with this bit-position must not be forwarded on this CCIX Link. 1b: Indicates that Broadcast DVM request from the RAID/DAID associated with this bit-position must be forwarded on this CCIX Link. CCIX Device initializes to 0000h after reset (except FLR).	RW

Figure 6-56 shows the layout of DVM Broadcast Forward Control Vector1 (DVMBCastFwdCntlVctr1) Register. The vector is Request or DVM Agent indexed from 32 to 63.

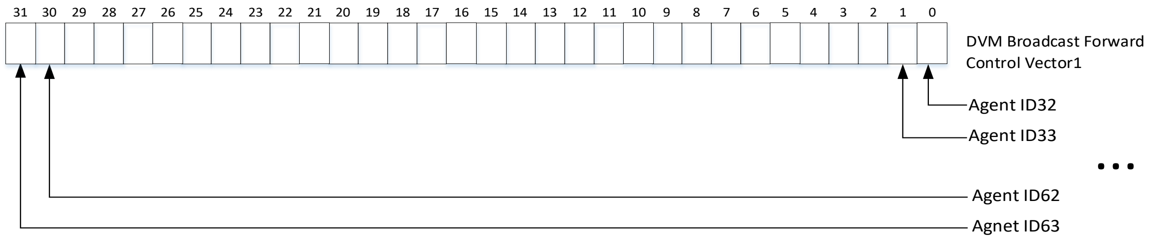


Figure 6-56: DVMBCASTFwdCntlVctr1 Register

Table 6-47 describes the fields of the DVMBCastFwdCntlVctr1 Register

Table 6-47: DVMBCastFwdCntlVctr1 Register Fields

Bit Location	Register Description	Attributes
31:0	**DVMBCastFwdCntlVctr1** DVMBCastFwdCntlVctr1[0] to DVMBCastFwdCntlVctr1[31] indicate the Broadcast DVM request Forward Controls for RAID32/DAID32 to RAID63/DAID63 respectively, for this CCIX Link: 0b: Indicates that Broadcast DVM request from the RAID/DAID associated with this bit-position must not be forwarded on this CCIX Link. 1b: Indicates that Broadcast DVM request from the RAID/DAID associated with this bit-position must be forwarded on this CCIX Link. CCIX Device initializes to 0000h after reset (except FLR).	RW

6.2.2.6.2.2 CCIX Link TransportID Map Entry

As illustrated in Figure 6-46, the Link TransportID Map Table is located in the Link Control Structure, after the Link Attribute Control structures.

Figure 6-57 shows the layout of a CCIX Link TransportID Map Entry (LinkTransportIDMapEntry) Register. The LinkTransportIDMapEntry Register contains the Destination TransportID field (DestTransportID), which is the BDF when the CCIX Transport Layer is PCIe. The remaining bits in this entry are Reserved and Preserved. The LinkTransportIDMapEntry Registers are arranged in CCIX Link number order, starting with the entry for CCIX Link #0 and ending with CCIX Link #(PortCapStat1.NumLinksCap – 1).

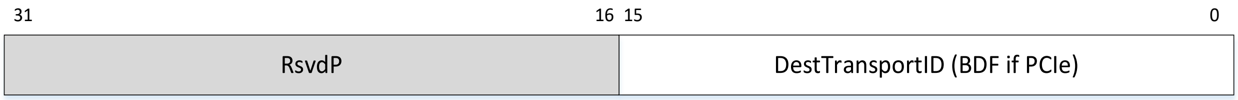


Figure 6-57: LinkTransportIDMapEntry Register

6.2.2.7 Home Agent Structures

Home Agent structures, as indicated by the Home Agent Component ID, contain attributes that describe the Home Agent. These structures include HBAT entries as described in Section 6.2.2.4.2. A Home Agent also has the optional capability to access Subordinate Agents for Memory Expansion. This optional capability requires SAM Table structures described in Section 6.2.2.3 , that contain the attributes for the G-HSAM Windows routed to Subordinate Agents that are made accessible to this Home Agent.

6.2.2.7.1 Home Agent Capabilities & Status Structure

Figure 6-58 shows the overall layout of the Home Agent Capabilities & Status structure. The HA Memory Pool Capabilities & Status structure is described in Section 6.2.2.4.1.

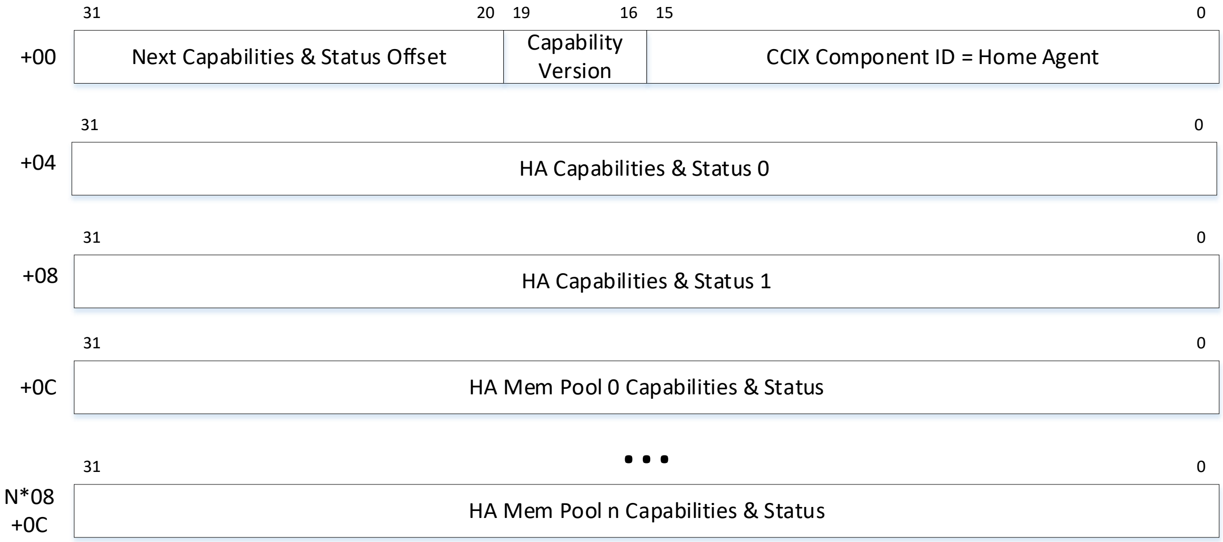

31
20
19
16
15
0
+00
Next Capabilities & Status Offset
Capability Version
CCIX Component ID = Home Agent
31
0
+04
HA Capabilities & Status 0
31
0
+08
HA Capabilities & Status 1
31
0
+0C
HA Mem Pool 0 Capabilities & Status
• • •
31
0
N*08 +0C
HA Mem Pool n Capabilities & Status

Figure 6-58: Home Agent Capabilities & Status Structure

Figure 6-59 shows the layout of the Home Agent Capabilities & Status 0 (HACapStat0) Register at Byte Offset-04h.

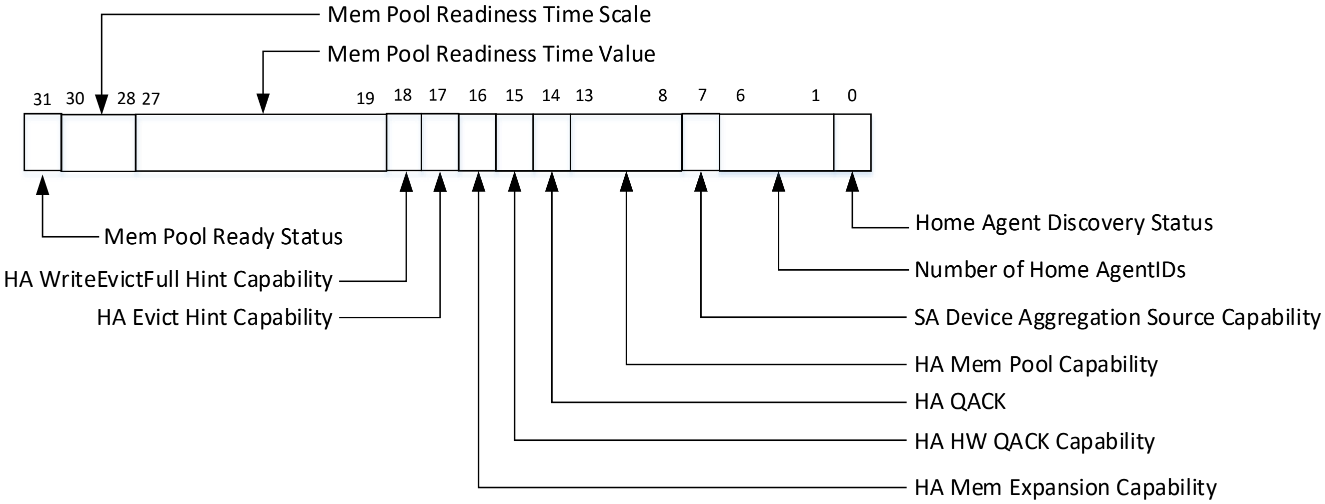

Mem Pool Readiness Time Scale
Mem Pool Readiness Time Value
31
30
28
27
19
18
17
16
15
14
13
8
7
6
1
0
Mem Pool Ready Status
HA WriteEvictFull Hint Capability
HA Evict Hint Capability
Home Agent Discovery Status
Number of Home AgentIDs
SA Device Aggregation Source Capability
HA Mem Pool Capability
HA QACK
HA HW QACK Capability
HA Mem Expansion Capability

Figure 6-59: HACapStat0 Register at Byte Offset-04h

Table 6-48 describes the HACapStat0 Register fields at Byte Offset-04h. Note, the HACapStat0 Register was formerly labeled HACapStat in previous revisions of the CCIX specification.

Table 6-48: HACapStat0 Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**HADiscRdyStat** This field describes the Home Agent's Discovery Readiness Status. 0b: • Indicates the Home Agent and its capabilities and control are not ready to be discovered and configured. 1b: • Indicates the Home Agent and its capabilities and control are ready to be discovered and configured. This also indicates the number of Memory Pools and the Pool Sizes are ready to be discovered and configured, i.e. Memory training is completed and the Memory Sizes are accurately reflected.	RO
6:1	**NumHAID** This field describes the number of Home AgentIDs that this Home Agent can express. 00h – 3Fh: Encodings for indicating capabilities for 1 to 64 HAIDs. CCIX Configuration Software is required to allocate for this CCIX Device as many HAIDs as indicated by NumHAID field. CCIX Configuration Software may choose not to enable a Home Agent if the CCIX AgentID name space cannot accommodate the total NumHAID from HAs across the system. The allocated SAM Window(s), PortID, BasePortID or Local attributes must be the same for all CCIX AgentIDs allocated to this Home Agent.	RO

Bit Location	Register Description	Attributes
7	**SADevAggSrcCap** This field indicates whether the CCIX Device and this Home Agent supports SA Device Aggregation. 0b: This Home Agent is not capable of SA Device Aggregation. 1b: This Home Agent is capable of SA Device Aggregation. The Home Agent must also be Memory Expansion Capable, i.e. HAMemExpnCap must have a value of 1b. Note: HA to SA Device Aggregation need not rely on only the HA indicating an SADevAggSrcCap value of 1b. HA to SA Device Aggregation can also be enabled when the HA is on a CCIX Device that supports Port Aggregation, as indicated by a ComnCapStat2.PortAggCap value of 1b, and the SA Devices connected to those aggregated CCIX Ports support SA Device Aggregation, as indicated by a SACapStat.SADevAggDestCap value of 1b. For further details, see Section 6.2.2.3.4.1.	RO
13:8	**HAMemPoolCap** This field describes the number of Memory Pools, and therefore, the number of unique G-HSAM Address windows that can target this HA. HAMemPoolCap, therefore, also indicates the number of HBAT entries. Because Memory Pool entry capabilities include the Memory Type attribute (MemPoolEntryCapStat0.MemPoolGenMemTypeCap) for each Memory Pool, the minimum number of Memory Pools must match the number of unique Memory Types attached to this HA. Because Memory Pool entry capabilities include the Memory Expansion Pool attribute (MemPoolEntryCapStat0.MemPoolGenMemTypeCap value of 1h), at least one Memory Pool entry must claim that attribute if the HA indicates this capability via the HACapStat.HAMemExpnCap capability bit. 00h: Reserved. 01h: Structures have 1 HA Mem Pool Entry. ... Nh: Structures have N HA Mem Pool Entries. An HA can have a maximum 63 Mem Pool Entries.	RO

Bit Location	Register Description	Attributes
14	**HAQACK** This field describes the CCIX HA's Quiesce Acknowledgement status. 0b: CCIX HA not quiesced 1b: CCIX HA quiesced If the HA has Hardware Quiesce Acknowledgement (HW QACK) capability, as indicated by a HACapStat.HAHWQACKCap value of 1b, HACapStat.HAQACK is set by implementation specific methods. Software can choose to poll HACapStat.HAQACK instead of waiting for the <HA Quiesce Time> value to check HAQACK if HACapStat.HAHWQACKCap has a value of 1b. If the HA does not have HW QACK capability as indicated in HACapStat.HAHWQACKCap value of 0b, the following sequence is followed: The HA sets the HAQACK bit to a value 1b after completing or detecting the following actions in this order: 1. The HA detects that the HA Quiesce Request (HACntl.HAQREQ) Control bit is set. 2. The HA has not issued any Requests and sent and received all relevant outstanding Responses, for the duration of <HA Quiesce Time>. <HA Quiesce Time> is based on the value of ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue, described further in Table 6-12. 3. Following the transition of the HA Request (HACntl.HAQREQ in Table 6-50) control bit from 0b to 1b, the HA must take no longer than 2 * <HA Quiesce Time> to set HAQACK. After 2 * <HA Quiesce Time>, CCIX Software detecting a zero returned for the HAQACK field indicates an error condition such that the HA was unable to reach a quiescent state. Following the transition of the HACntl.HAQREQ control bit from 1b to 0b, the HA must transition the HAQACK bit from 1b to 0b.	RO
15	**HAHWQACKCap** This field describes the HA's Hardware Quiesce Acknowledgement Capability. 0b: • The HA does not have a hardware mechanism to achieve a quiescent state. 1b: • The HA has a hardware mechanism to achieve a quiescent state.	RO

Bit Location	Register Description	Attributes
16	**HAMemExpnCap** This field controls enabling the HA for HA Memory Expansion. 0b: • Indicates this HA doesn't support Memory Expansion via accessing Subordinate Agents. 1b: • Indicates this HA supports Memory Expansion via accessing Subordinate Agents.	RO
17	**HAEvictHintCap** This field indicates the HA's preferred behavior of Evict transactions and may influence the Request Agent's Evict send behavior. 0b: • Sending Evict transactions from the RA is not recommended for best system performance. However, it is permitted to send the transaction. 1b: • Sending Evict transactions from the RA is recommended for best system performance. However, it is permitted not to send the transaction.	RO
18	**HAWriteEvictFullHintCap** This field indicates the HA's preferred behavior of WriteEvictFull transactions and may influence Request Agent's WriteEvictFull send behavior. 0b: • Sending WriteEvictFull transactions from the RA is not recommended for best system performance. However, it is permitted to send the transaction. 1b: • Sending WriteEvictFull transactions from the RA is recommended for best system performance. However, it is permitted not to send the transaction.	RO

Bit Location	Register Description	Attributes
27:19	**HAMemPoolRdyTimeValue** This field describes the HA's Memory Pool Readiness Time Value Encoding where the overall <HA Mem Pool Readiness Time Reported> is HAMemPoolRdyTimeValue*<HA Mem Pool Readiness Time Multiplier>. <HA Mem Pool Readiness Time Multiplier> is $32^{HAMemPoolRdyTimeScale}$ns. 000h – 1FFh: Encodings for HAMemPoolRdyTimeValue of 0 through 511 A HAMemPoolRdyTimeValue value of 000h indicates the Memory Pools are always ready to be discovered and configured, i.e. HAMemPoolRdyTimeScale is always 1b. The <HA Mem Pool Readiness Time Reported> must be the longer of the following two readiness times: 1. The readiness time following a Conventional Reset and 2. The readiness time following a Function Level Reset. The validity of the HAMemPoolRdyTimeValue field is not subject to the values of the HADiscRdyStat or HAMemPoolRdyTimeScale fields. The HAMemPoolRdyTimeValue field is valid at the same time the CCIX Protocol Layer DVSEC Header is valid.	RO
30:28	**HAMemPoolRdyTimeScale** This field describes the HA's Memory Pool Readiness Time Scale Encoding in order to generate the <Readiness Time Multiplier> where: <HA Mem Pool Readiness Time Multiplier> is $32^{HAMemPoolRdyTimeScale}$ns. 0h – 7h: Encodings for HAMemPoolRdyTimeScale of 0 through 7 which allows for <HA Mem Pool Readiness Time Multiplier> values of 32ns through 34359738368ns. The validity of the HAMemPoolRdyTimeScale field is not subject to the value of the HADiscRdyStat or HAMemPoolRdyTimeScale fields. The HAMemPoolRdyTimeScale field is valid at the same time the CCIX Protocol Layer DVSEC Header is valid. During initialization, should an HA determine that it's capable of a lower readiness time for its MemPools, the HA is permitted to reduce its <HA Mem Pool Readiness Time Reported> by either reducing its HAMemPoolRdyTimeValue, HAMemPoolRdyTimeScale, or both. However, CCIX Configuration Software may use either the original or reduced <HA Mem Pool Readiness Time Reported>. An HA is not permitted to increase its <HA Mem Pool Readiness Time Reported>.	RO

Bit Location	Register Description	Attributes
31	**HAMemPoolRdyStat** This field describes the Home Agent Memory Pool's Readiness Status. 0b: • Indicates the Home Agent's Memory Pool attributes are not ready to be discovered. 1b: • Indicates the Home Agent's Memory Pool capabilities are ready to be discovered, e.g. the Memory Pool Size(s) have been determined, but the Memory Pool(s) may not be trained and therefore, the final determination for Memory Pool Size(s) may not have occured. A HA indicates a HAMemPoolRdyStat value of 1b and HADiscRdyStat value of 0b when the Memory Pool size(s) and other attributes have been determined, but memory training has not been completed. When memory training has been completed and the final, post-training, Memory Pool size(s) have been accurately indicated, then the HA indicates a value of 1b for both HAMemPoolRdyStat and HADiscRdyStat. A HA must take no longer than <HA Mem Pool Readiness Time Reported> to set HAMemPoolRdyStat. After <HA Mem Pool Readiness Time Reported>, CCIX Software detecting a zero returned for the HAMemPoolRdyStat field indicates that the HA cannot be configured/enumerated.	RO

Figure 6-60 shows the layout of the Home Agent Capabilities & Status 2 (HACapStat1) Register at Byte Offset-08h.

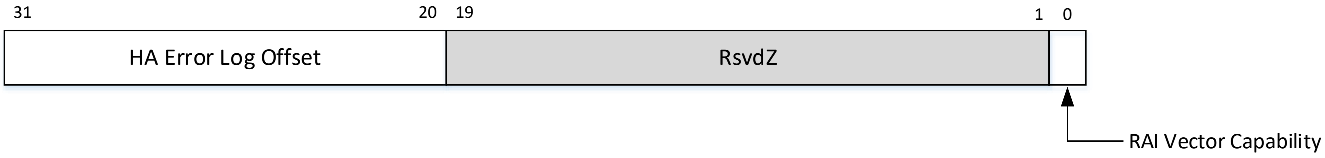


Figure 6-60: HACapStat1 Register at Byte Offset-08h

Table 6-49 describes the HACapStat1 Register fields at Byte Offset-08h.

5

Table 6-49: HACapStat1 Register Fields at Byte Offset-08h

Bit Location	Register Description	Attributes
0	**RAIVctrCap** This field describes whether the HA includes support for the RA-I Present Vector. 0b: The HA does not support the RA-I Present Vector. 1b: The HA does support the RA-I Present Vector.	RO
19:1	Reserved and Zero	RsvdZ
31:20	**HAErrLogOffset** This field contains the HA Error Log Offset which is described in CCIX RAS Overview.	RO

6.2.2.7.2 Home Agent Control Structure

Figure 6-61 shows the overall layout of the Home Agent Control structure. The HA BAT Entry structure is described in Section 6.2.2.4.2.

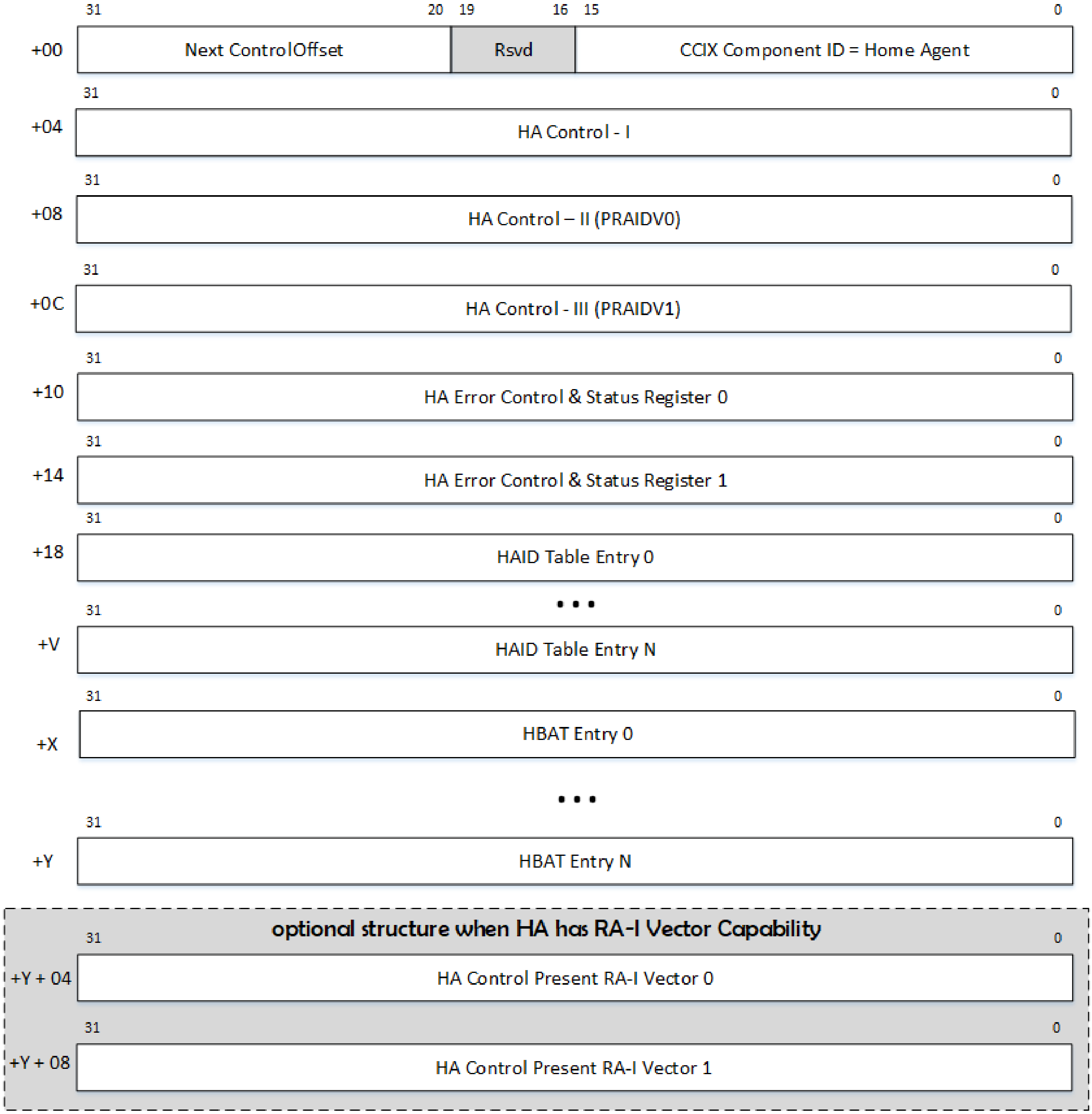

31
20
19
16
15
0
+00
Next ControlOffset
Rsvd
CCIX Component ID = Home Agent
+04
HA Control - I
+08
HA Control – II (PRAIDV0)
+0C
HA Control - III (PRAIDV1)
+10
HA Error Control & Status Register 0
+14
HA Error Control & Status Register 1
+18
HAID Table Entry 0
• • •
+V
HAID Table Entry N
+X
HBAT Entry 0
• • •
+Y
HBAT Entry N
optional structure when HA has RA-I Vector Capability
+Y + 04
HA Control Present RA-I Vector 0
+Y + 08
HA Control Present RA-I Vector 1

Figure 6-61: Home Agent Control Registers

5 Figure 6-62 shows the layout of the Home Agent Control (HACntl) Register at Byte Offset-04h.

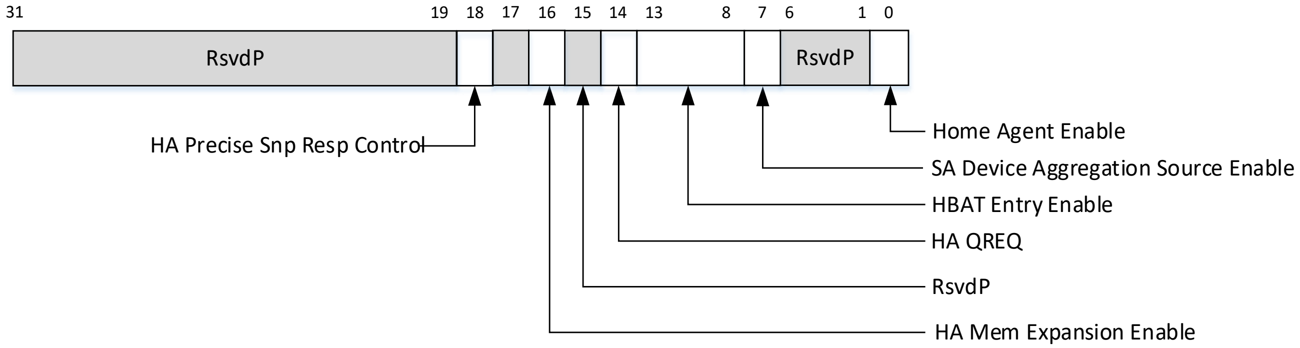


Figure 6-62: HACntl Register at Byte Offset-04h

Table 6-50 describes the HACntl Register fields at Byte Offset-04h.

Table 6-50: HACntl Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**HAEnable** This field controls enabling the Home Agent. 0b: • Indicates either the HA has not been configured, or the previously configured HA has been taken offline. 1b: • Indicates the HA is enabled. The HA has been configured, must service Requests from RAs, and can send Snoops to RAs. The HA can also sent Requests to SAs if Memory Expansion has been enabled. CCIX Device initializes to 0b after reset (except FLR).	RW
6:1	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
7	**SADevAggSrcEnable** This field controls enabling this Home Agent with SA Device Aggregation. SADevAggSrcEnable is the single enable across all HSAM Entries with their address range routed to aggregated SA Devices. The NumAggDevCntl field in each HSAM Entry, described inTable 6-21, further controls whether that particular HSAM Entry's address range is routed to aggregated SA Devices. 0b: This Home Agent is not enabled for SA Device Aggregation. 1b: This Home Agent is enabled for SA Device Aggregation. The Home Agent must also be enabled for Memory Expansion, i.e. HAMemExpnEnable must have a value of 1b.	RW
13:8	**HBATEntryEnable** This field describes the number of Memory Pools enabled and therefore the number of unique G-HSAM Address windows that have been allocated to this HA. HBATEntryEnable therefore also indicates the corresponding number of HBAT Control entries. 00h: Control Structures do not have any HBAT Entries enabled. 01h: Control Structures have 1 HBAT Entry. … Nh: Control Structures have N HBAT Entries. A HA can have a maximum 63 HBAT Entries enabled. CCIX Device initializes to 00h after reset (except FLR).	RW
14	**HAQREQ** HA Quiesce Request controls when the HA will act to achieve a quiesced state. There can only be a change in the value, 0b or 1b, of the corresponding HACapStat.HAQACK bit after CCIX configuration software changes the value, 0b or 1b, of this HAQREQ control bit. 0b: • HA is not required to be quiesced. 1b: • HA must be quiesced. CCIX Confguration software must set the bit to a value 1b after first quiescing the RAs that can access this HA. CCIX Device initializes to 0b after reset (except FLR).	RW

Bit Location	Register Description	Attributes
15	Reserved and Preserved	RsvdP
16	**HAMemExpnEnable** This field enables memory expansion in a Home Agent. 0b: • Indicates this HA is not enabled for HA Memory Expansion via accessing Subordinate Agents. 1b: • Indicates this HA is enabled for HA Memory Expansion via accessing Subordinate Agents. If this HA is enabled for HA Memory Expansion, then at least one Memory Expansion Capable (MemPoolEntryCapStat0.MemPoolGenMemTypeCap value of 1h) Memory Pool Entry must be valid in order to indicate the 4GB or 2^n size aligned Base Address for the mapped Memory Expansion Pool. CCIX Device initializes to 0b after reset (except FLR).	RW
17	Reserved and Preserved	RsvdP
18	**HAPreciseSnpRespCntl** This field controls whether the HA is allowed to precisely track UC and UD state for SnpRespData type Snoop Response from an RA. HAs must allow for an imprecise SnpRespData type Snoop Response from an RA, i.e. HAs must support a HAPreciseSnpRespCntl value of 0b. HAs are permitted to support behavior consistent with a HAPreciseSnpRespCntl value of 0b even if HAPreciseSnpRespCntl is programmed with a value of 1b. 0b: • Indicates at least one RA is not capable of communicating a precise UC or UD state when providing either a SnpRespData_UC or SnpRespData_UD type Snoop Response. 1b: • Indicates all RAs are capable of communicating a precise UC or UD state when providing SnpRespData_UC or SnpRespData_UD type Snoop Response. CCIX Device initializes to 0b after reset (except FLR).	RW
31:19	Reserved and Preserved	RsvdP

Figure 6-63 shows the layout of the Home Agent Control Present RAID Vector 0 (HACntlPresentRAIDVctr0) Register at Byte Offset-08h.

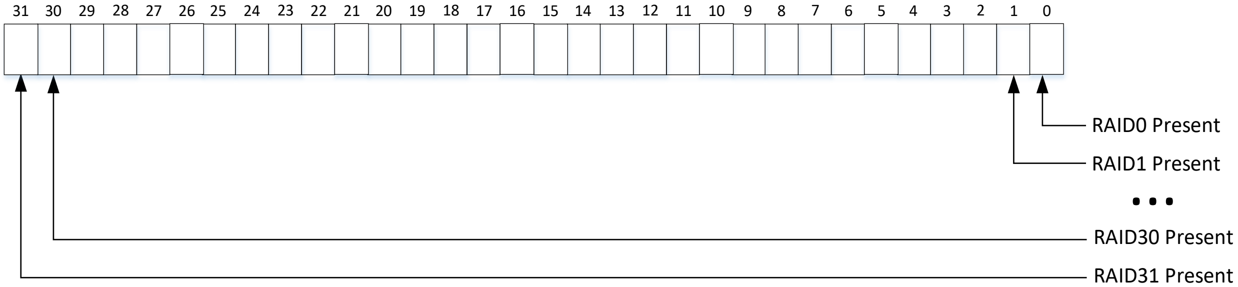


Figure 6-63: HACntlPresentRAIDVctr0 Register

Table 6-51 describes the HACntlPresentRAIDVctr0 Register fields at Byte Offset-08h.

Table 6-51: HACntlPresentRAIDVctr0 Register Fields

Bit Location	Register Description	Attributes
31:0	**PresentRAIDVctr0** PresentRAIDVctr0[0] to PresentRAIDVctr0[31] indicate the Request AgentID is present for RAID0 to RAID31 respectively: 0b: Indicates that the RAID associated with this bit-position is not present. 1b: Indicates that the RAID associated with this bit-position is present. A 1b setting is permitted for an RA that is present and enumerated with an RAID, but not enabled to send traffic. The opposite, however, is not permitted, i.e. a 0b setting is not allowed if an RA is present and enumerated. CCIX Device initializes to 0000h after reset (except FLR).	RW

Figure 6-64 shows the layout of the Home Agent Control Present RAID Vector 1 (HACntlPresentRAIDVctr1) Register at Byte Offset-0Ch.

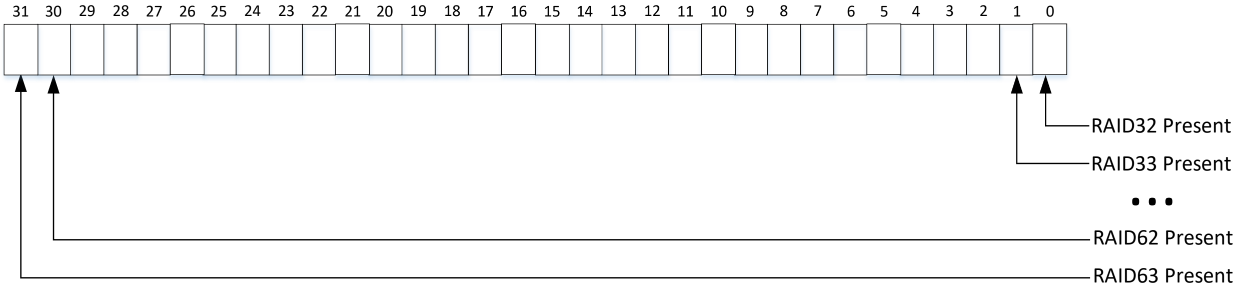


Figure 6-64: HACntlPresentRAIDVctr1 Register

Table 6-52 describes the HACntlPresentRAIDVctr1 Register fields at Byte Offset-0Ch.

Table 6-52: HACntlPresentRAIDVctr1 Register Fields

Bit Location	Register Description	Attributes
31:0	**PresentRAIDVctr1** PresentRAIDVctr1[0] to PresentRAIDVctr1[31] indicate the Request AgentID is present for RAID32 to RAID63 respectively: 0b: Indicates that the RAID associated with this bit-position is not present. 1b: Indicates that the RAID associated with this bit-position is present. A 1b setting is permitted for an RA that is present and enumerated with an RAID, but not enabled to send traffic. The opposite, however, is not permitted, i.e. a 0b setting is not allowed if an RA is present and enumerated. CCIX Device initializes to 0000h after reset (except FLR).	RW

The HA Error Control & StatusRegisters, HAErrCntlStat0 and HAErrCntlStat1, at Byte Offset-10h and Byte Offset-14h respectively, are described in CCIX RAS Overview. Figure 6-65 shows the layout of the Home AgentID Table Entry 0 (HAIDTblEntry0) Control Register at Byte Offset-18h. The Home AgentID Table is AgentID indexed and CCIX Configuration Software can program the CCIX AgentID of up to 4 CCIX Agents per entry. The total number of entries in the Home AgentID Table is the quotient of HACapStat.NumHAID divided by 4, rounded to the next higher DW boundary. Unused entries in a DW are Reserved and Preserved. Thus the maximum number of Home AgentID Table entries is 16.

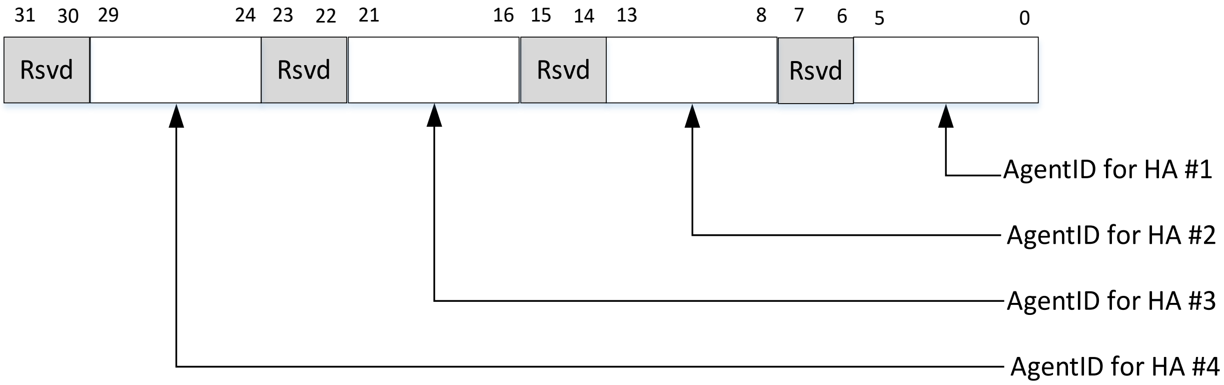


Figure 6-65: HAIDTblEntry0 Register at Byte Offset-18h

Table 6-53 describes an individual HAIDTblEntry Register in the Home AgentID Table.

Table 6-53: HAIDTblEntry Register Fields

Bit Location	Register Description	Attributes
5:0	**AgentIDforHAn** This field contains the CCIX AgentID to be expressed for Home Agent number N, where N ranges from 1 up to the number indicated by the HACapStat.NumHAID encoding described in Table 6-48. AgentIDforHAn is required to be unique across all Home Agents in a CCIX system. Also, an CCIX AgentID is unique to a CCIX Device and cannot be re-used to enumerate a CCIX Agent on any other CCIX Device. An HA and RA on the same CCIX Device is permitted to share a CCIX AgentID. 00h – 3Fh: Encodings for HAID0 through HAID63.	RW
7:6	Reserved and Preserved	RsvdP

The HBAT Control structure, located after the HAID Table, and illustrated in Figure 6-61, has the structure and definition of BAT Control structures described in detail in Section 6.2.2.4.2.

Table 6-54: HA Control Present RA-I Vector 1 Register Fields

Bit Location	Register Description	Attributes
31:0	**Present RAIVctr0** PresentRAIVctr0[0] to PresentRAIVctr0[31] indicates the Request Agent is an IO Coherent Request Agent, RA-I, for RAID0 to RAID31 respectively: 0b: Indicates that the RAID associated with this bit-position is not an IO Coherent Request Agent, RA-I. 1b: Indicates that the RAID associated with this bit-position is an IO Coherent Request Agent, RA-I.	RW

	A 0b setting is permitted for an RA-I that is present. The opposite, however, is not permitted, i.e. a 1b setting is not allowed if an RA is present, but not an RA-I. A 1b setting is only permitted if the corresponding bit in PresentRAIDVctr0 is also set to 1b. CCIX Device initializes to 0000h after reset (except FLR).	

Table 6-55: HA Control Present RA-I Vector 1 Register Fields

Bit Location	Register Description	Attributes
31:0	**Present RAIVctr1** PresentRAIVctr1[0] to PresentRAIVctr1[31] indicates the Request Agent is an IO Coherent Request Agent, RA-I, for RAID32 to RAID63 respectively: 0b: Indicates that the RAID associated with this bit-position is not an IO Coherent Request Agent, RA-I. 1b: Indicates that the RAID associated with this bit-position is an IO Coherent Request Agent, RA-I. A 0b setting is permitted for an RA-I that is present. The opposite, however, is not permitted, i.e. a 1b setting is not allowed if an RA is present, but not an RA-I. A 1b setting is only permitted if the corresponding bit in PresentRAIDVctr1 is also set to 1b. CCIX Device initializes to 0000h after reset (except FLR).	RW

6.2.2.8 Request Agent Structures

6.2.2.8.1 Request Agent Capabilities & Status Structure

Figure 6-66 shows the overall layout of the RA Capabilities & Status Structure.

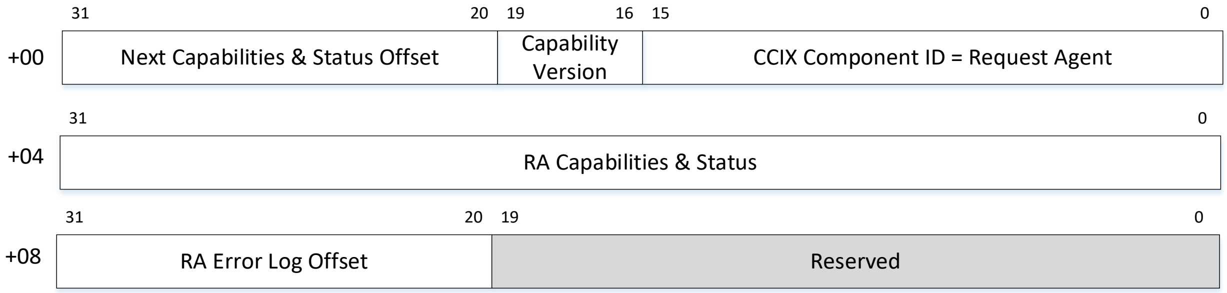


Figure 6-66: RA Capabilities & Status Structure

5 Figure 6-67 shows the layout of the RA Capabilities & Status (RACapStat) Register at Byte Offset-04h.

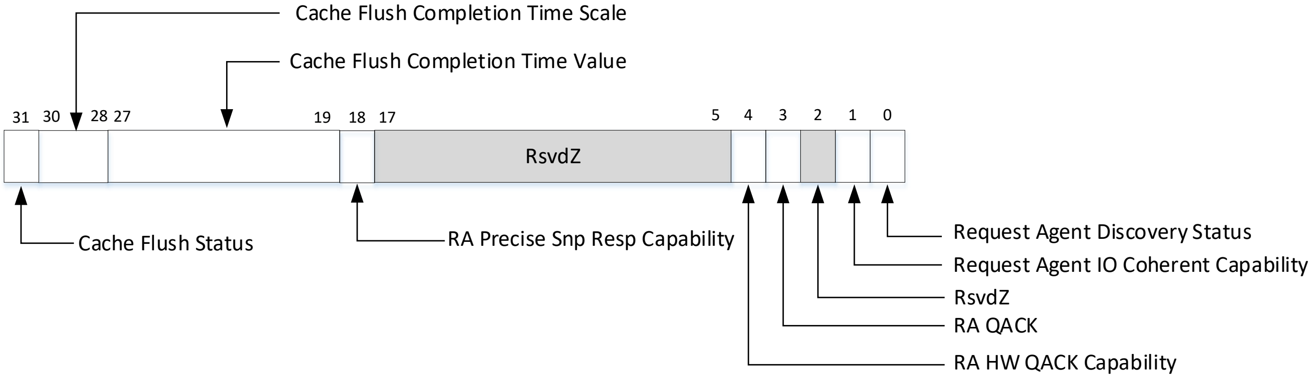

Cache Flush Completion Time Scale
Cache Flush Completion Time Value
31
30
28
27
19
18
17
5
4
3
2
1
0
RsvdZ
Cache Flush Status
RA Precise Snp Resp Capability
Request Agent Discovery Status
Request Agent IO Coherent Capability
RsvdZ
RA QACK
RA HW QACK Capability

Figure 6-67: RACapStat Register at Byte Offset-04h

Table 6-56 describes the RACapStat Register fields at Byte Offset-04h.

Table 6-56: RACapStat Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**RADiscRdyStat** This field describes the Request Agent's Discovery Readiness Status. 0b: • Indicates the Request Agent and its capabilities and control are not ready to be discovered and configured. 1b: • Indicates the Request Agent and its capabilities and control are ready to be discovered and configured.	RO
1	**RAICap** RAICap – Request Agent IO Coherent Capability. This field indicates if the Request Agent is IO Coherent and will not issue transactions that can allocate in a coherent cache. 0b: Indicates the Request Agent is fully coherent and can issue transactions that allow the associated cache line to be allocated in a coherent cache. 1b: Indicates the Request Agent is IO coherent and will not issue transactions that allow the associated cache line to be allocated in a coherent cache. Note: For backwards compatibility the 0b setting indicates fully coherent.	RO
2	Reserved and Zero	RsvdZ

Bit Location	Register Description	Attributes
3	**RAQACK** This field describes the CCIX RA's Quiesce Acknowledgement status. 0b: CCIX RA not quiesced 1b: CCIX RA quiesced If the RA has Hardware Quiesce Acknowledgement (HW QACK) capability, as indicated by a RACapStat.RAHWQACKCap value of 1b, RACapStat.RAQACK is set by implementation specific methods. Software can choose to poll RACapStat.RAQACK instead of waiting for the <RA Quiesce Time> value to check RAQACK if RACapStat.RAHWQACKCap has a value of 1b. If the RA does not have HW QACK capability as indicated in RACapStat.RAHWQACKCap value of 0b, the following sequence is followed: The RA sets the RAQACK bit to a value 1b after completing or detecting the following actions in this order: 1. The RA detects that the RA Quiesce Request (RACntl.RAQREQ) Control bit is set. 2. The RA has not issued any Requests and sent and received all relevant outstanding Responses, for the duration of <RA Quiesce Time>. <RA Quiesce Time> is based on the value of ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue, described further in Table 6-12. 3. Following the transition of the RA Request (RACntl.RAQREQ in Table 6-52) control bit from 0b to 1b, the RA must take no longer than 2 * <RA Quiesce Time> to set RAQACK. After 2 * <RA Quiesce Time>, CCIX Software detecting a zero returned for the RAQACK field indicates an error condition such that the RA was unable to reach a quiescent state. Following the transition of the RACntl.RAQREQ control bit from 1b to 0b, the RA must transition the RAQACK bit from 1b to 0b.	RO
4	**RAHWQACKCap** This field describes the RA's Hardware Quiesce Acknowledgement Capability. 0b: The RA does not have a hardware mechanism to achieve a quiescent state. 1b: The RA has a hardware mechanism to achieve a quiescent state.	RO
17:5	Reserved and Zero	RsvdZ

Bit Location	Register Description	Attributes
18	**RAPreciseSnpRespCap** This field indicates the RA's capability to communicate a precise UC or UD state for SnpRespData type Snoop Response. 0b: • Indicates the RA is not capable of communicating a precise UC or UD state when providing either a SnpRespData_UC or SnpRespData_UD type Snoop Response. 1b: • Indicates the RA is capable of communicating a precise UC or UD state when providing SnpRespData_UC or SnpRespData_UD type Snoop Response.	RO
27:19	**CacheFlushTimeValue** This field describes the RA's Cache Flush Time Value Encoding where the overall <Cache Flush Time Reported> is CacheFlushTimeValue *<Cache Flush Time Multiplier>. <Cache Flush Time Multiplier> is $32^{\text{CacheFlushTimeScale}}$ns. 000h – 1FFh: Encodings for CacheFlushTimeValue of 0 through 511. • The validity of CacheFlushTimeValue is not subject to the RACapStat.RADiscRdyStat indicator. CacheFlushTimeValue is valid at the same time the CCIX Protocol Layer DVSEC Header is valid.	RO
30:28	**CacheFlushTimeScale** This field describes the RA's Cache Flush Time Scale Encoding in order to generate the <Cache Flush Time Multiplier> where <Cache Flush Time Multiplier> is $32^{\text{CacheFlushTimeScale}}$ns. 0h – 7h: Encodings for CacheFlushTimeScale of 0 through 7 which allows for <Cache Flush Time Multiplier> values of 32ns through 34359738368ns. The validity of CacheFlushTimeScale is not subject to the RACapStat.RADiscRdyStat indicator. CacheFlushTimeScale is valid at the same time the CCIX Protocol Layer DVSEC Header is valid.	RO
31	**RACacheFlushStat** 0b: Cache Flush operation not completed 1b: Cache Flush operation completed. Following the transition of the Request Agent Flush Enable (RACntl.RACacheFlushEnable in Table 6-57) control bit from 0b to 1b, the RA must take no longer than <Cache Flush Time Reported> to set RACacheFlushStat. After <Cache Flush Time Reported>, CCIX Software detecting a zero returned for the RACacheFlushStat field indicates that an error has occurred in the RA during the cache flush operation. Following the transition of the RACntl.RACacheFlushEnable control bit from 1b to 0b, the RA must transition the RACacheFlushStat bit from 1b to 0b. CCIX Device initializes to 0b after reset (except FLR).	RO

The RA Capabilities & Status Error Log Pointer (RAErrLogPtr) Register at Byte Offset-08h contains the RA Error Log Offset which is described in CCIX RAS Overview. The remaining bits in this register are Reserved and Zero.

6.2.2.8.2 Request Agent Control Structure

Figure 6-68 shows the overall layout of the Request Agent Control structure.

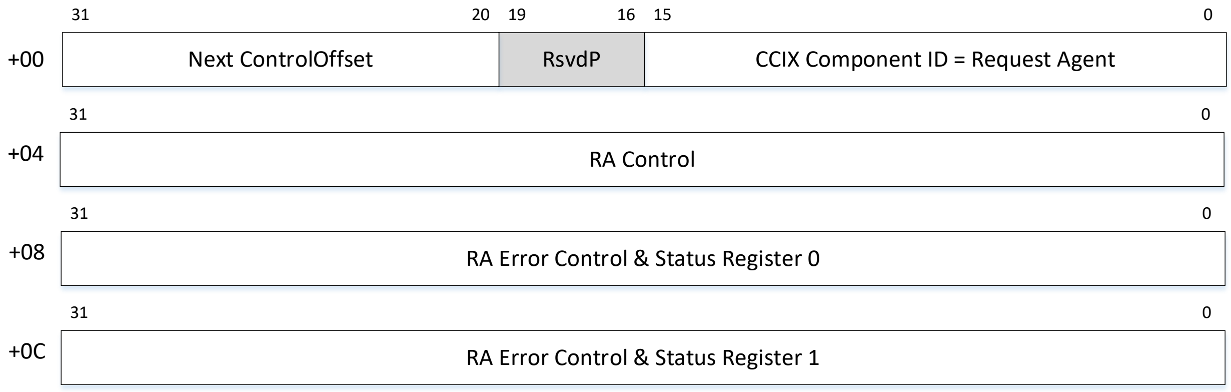

31
20
19
16
15
0
+00
Next ControlOffset
RsvdP
CCIX Component ID = Request Agent
31
0
+04
RA Control
31
0
+08
RA Error Control & Status Register 0
31
0
+0C
RA Error Control & Status Register 1

Figure 6-68: Request Agent Control Registers

Figure 6-69 shows the layout of the Request Agent Control (RACntl) Register at Byte Offset-04h.

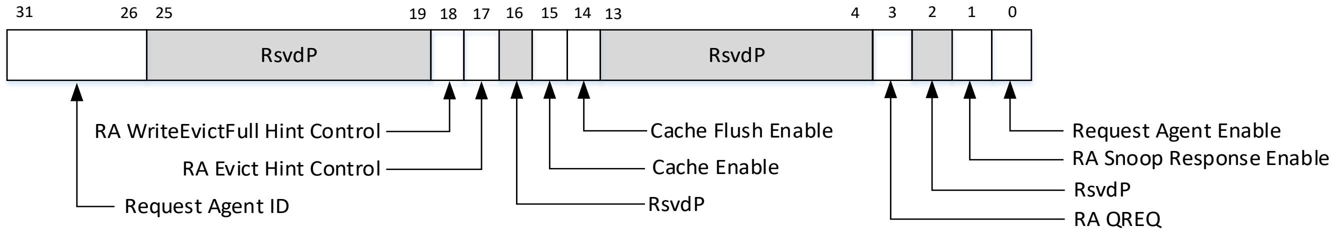


Figure 6-69: RACntl Register at Byte Offset-04h

Table 6-57 describes the RACntl Register fields at Byte Offset-04h.

Table 6-57: RACntl Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**RAEnable** This field controls enabling the Request Agent. 0b: Indicates either the RA has not been configured, or the previously configured RA has been taken offline. 1b: Indicates the RA is enabled. A 0-to-1 transition indicates the RA has been configured, and can send Requests to HAs. CCIX Device initializes to 0b after reset (except FLR).	RW

Bit Location	Register Description	Attributes
1	**RASnpRspEnable** This field controls enabling Snoop Responses from the Request Agent. 0b: Indicates the RA has not been enabled to send Snoop Responses. 1b: Indicates the RA is enabled to send Snoop Responses. The RA must receive Snoop Requests and must send a protocol compliant Snoop Response. RASnpRspEnable must be set to 1b along with, or prior to, RAEnable being set to 1b. CCIX Device initializes to 0b after reset (except FLR).	RW
2	Reserved and Preserved	RsvdP
3	**RAQREQ** RA Quiesce Request controls when the RA will act to achieve a quiesced state. There can only be a change in the value, 0b or 1b, of the corresponding RACapStat.RAQACK bit after CCIX configuration software changes the value, 0b or 1b, of this RAQREQ control bit. 0b: RA is not required to be quiesced. 1b: RA must be quiesced. CCIX Confguration software must set the bit to a value 1b after first quiescing the AFs being serviced by this RA. CCIX Device initializes to 0b after reset (except FLR).	RW
13:4	Reserved and Preserved	RsvdP
14	**RACacheFlushEnable** This field controls initiating a Flush and Invalidate operation of the Request Agent Cache. 0b: Indicates Cache Flush disabled. 1b: Indicates Cache Flush enabled. The RA must flush all Dirty copies back to the HA and invalidate all entries. Upon completion of the Cache Flush operation, the RA sets the RACapStat.RACacheFlushStat bit. The RA must continue to service Snoops from the HA during the Cache Flush operation. CCIX Configuration Software must clear RACacheFlushEnable on detecting that the Request Agent Cache Flush Status, RACapStat.RACacheFlushStat, indicates completion of the Cache Flush operation. CCIX Device initializes to 0b after reset (except FLR).	RW

Bit Location	Register Description	Attributes
15	**RACacheEnable** This field enables the Request Agent Cache. 0b: • Disable allocation of Home Agent Cachelines into the RA cache. • The RA must continue to service Snoops from the HA when RASnpRspEnable is set to 1b, even if the RACacheEnable bit indicates caching is disabled. 1b: • Enable Caching of Home Agent Cachelines. CCIX Device initializes to 0b after reset (except FLR).	RW
16	Reserved and Preserved	RsvdP
17	**RAEvictHintCntl** This field indicates the preferred behavior of Evict transactions in the system, and optionally controls the Request Agent's Evict send behavior. 0b: • Sending Evict transactions from the RA is not recommended for best system performance. However, it is permitted to send the transaction. 1b: • Sending Evict transactions from the RA is recommended for best system performance. However, it is permitted not to send the transaction. CCIX Device initializes to 0b after reset (except FLR).	RW
18	**RAWriteEvictFullHintCntl** This field indicates the preferred behavior of WriteEvictFull transactions in the system, and optionally controls the Request Agent's WriteEvictFull send behavior. 0b: • Sending WriteEvictFull transactions from the RA is not recommended for best system performance. However, it is permitted to send the transaction. 1b: • Sending WriteEvictFull transactions from the RA is recommended for best system performance. However, it is permitted not to send the transaction. CCIX Device initializes to 0b after reset (except FLR).	RW
25:19	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
31:26	**RAID** The Request AgentID (RAID) is required to be unique across across all Request Agents in a CCIX system. Also, a CCIX AgentID is unique to a CCIX Device and cannot be re-used to enumerate a CCIX Agent on any other CCIX Device. An HA and RA on the same CCIX Device is permitted to share a CCIX AgentID. 00h – 3Fh: Encodings for RAID0 through RAID63. RAID must be programmed along with, or prior to, RASnpRspEnable being set to 1b. CCIX Device initializes to 0b after reset (except FLR).	RW

Subordinate Agent Structures

Subordinate Agent structures, as indicated by the Subordinate Agent Component ID, contain attributes that describe the Subordinate Agent. These structures include SBAT entries as described in Section 6.2.2.4.2.

6.2.2.8.3 Subordinate Agent Capabilities & Status Structure

Figure 6-70 shows the overall layout of the Subordinate Agent Capabilities & Status structure.

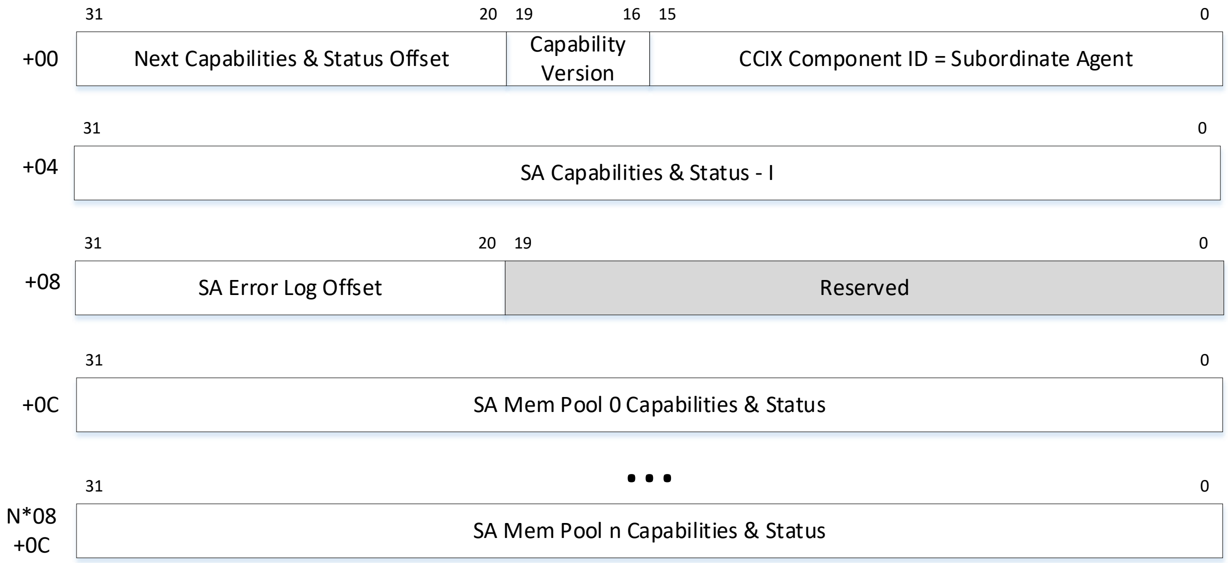

31
20
19
16
15
0
+00
Next Capabilities & Status Offset
Capability Version
CCIX Component ID = Subordinate Agent
31
0
+04
SA Capabilities & Status - I
31
20
19
0
+08
SA Error Log Offset
Reserved
31
0
+0C
SA Mem Pool 0 Capabilities & Status
• • •
31
0
N*08 +0C
SA Mem Pool n Capabilities & Status

Figure 6-70: Subordinate Agent Capabilities & Status Structure

Figure 6-71 shows the layout of the Subordinate Agent Capabilities & Status (SACapStat) Register at Byte Offset-04h.

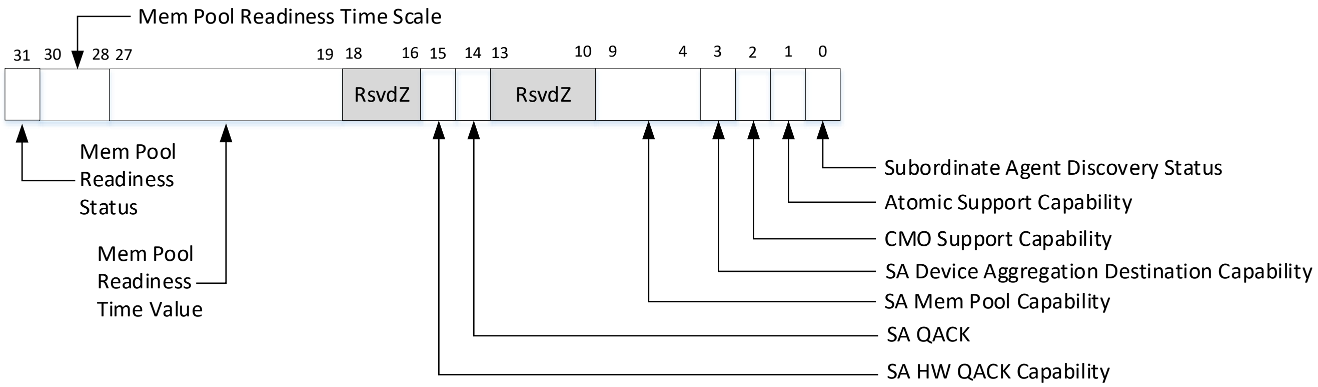

Mem Pool Readiness Time Scale
31
30
28
27
19
18
16
15
14
13
10
9
4
3
2
1
0
RsvdZ
RsvdZ
Mem Pool Readiness Status
Mem Pool Readiness Time Value
Subordinate Agent Discovery Status
Atomic Support Capability
CMO Support Capability
SA Device Aggregation Destination Capability
SA Mem Pool Capability
SA QACK
SA HW QACK Capability

Figure 6-71: SACapStat Register at Byte Offset-04h

Table 6-58 describes the SACapStat Register fields at Byte Offset-04h.

Table 6-58: SACapStat Register Fields at Byte Offset-04h

Bit Location	Register Description	Attributes
0	**SADiscRdyStat** This field describes the Subordinate Agent's Discovery Readiness Status. 0b: Indicates the Subordinate Agent and its capabilities and control are not ready to be discovered and configured. 1b: Indicates the Subordinate Agent and its capabilities and control are ready to be discovered and configured. This also indicates the number of Memory Pools and the Pool Sizes are ready to be discovered and configured, i.e. Memory training is completed and the Memory Sizes are accurately reflected.	RO
1	**SAAtomicSupportCap** This field describes the Subordinate Agent's Atomic Support Capability. 0b: Indicates the Subordinate Agent is not capable of supporting Atomic transactions. 1b: Indicates the Subordinate Agent is capable of supporting Atomic transactions.	RO
2	**SACMOSupportCap** This field describes the Subordinate Agent's CMO Support Capability. 0b: Indicates the Subordinate Agent is not capable of supporting CMO transactions. 1b: Indicates the Subordinate Agent is capable of supporting CMO transactions.	RO
3	**SADevAggDestCap**	RO

Bit Location	Register Description	Attributes
	This field indicates whether the Subordinate Agent supports SA Device Aggregation. 0b: This Subordinate Agent is not capable of SA Device Aggregation. 1b: This Subordinate Agent is capable of SA Device Aggregation.	
9:4	**SAMemPoolCap** This field describes the number of Memory Pools, and therefore, the number of unique G-HSAM Address windows that can target this SA. SAMemPoolCap, therefore, also indicates the corresponding number of SBAT entries. Because Memory Pool entry capabilities include the Memory Type attribute (MemPoolEntryCapStat0.MemPoolGenMemTypeCap) for each Memory Pool, the minimum number of Memory Pools must match the number of unique Memory Types attached to this SA. 00h: Reserved. 01h: Structure has 1 SA Mem Pool Entry. … Nh: Structures have N SA Mem Pool Entries. An SA can have a maximum 63 SA Mem Pool Entries.	RO
10:13	Reserved and Zero	RsvdZ
14	**SAQACK** This field describes the CCIX SA's Quiesce Acknowledgement status. 0b: CCIX SA not quiesced 1b: CCIX SA quiesced If the SA has Hardware Quiesce Acknowledgement (HW QACK) capability, as indicated by a SACapStat.SAHWQACKCap value of 1b, SACapStat.SAQACK is set by implementation specific methods. Software can choose to poll SACapStat.SAQACK instead of waiting for the <SA Quiesce Time> value to check SAQACK if SACapStat.SAHWQACKCap has a value of 1b. If the SA does not have HW QACK capability as indicated in SACapStat.SAHWQACKCap value of 0b, the following sequence is followed: The SA sets the SAQACK bit to a value 1b after completing or detecting the following actions in this order: 1. The SA detects that the SA Quiesce Request (SACntl.SAQREQ) Control bit is set. 2. The SA has not issued any Requests and sent and received all relevant outstanding Responses, for the duration of <SA Quiesce	RO

Bit Location	Register Description	Attributes
	Time>. <SA Quiesce Time> is based on the value of ComnCntl2.QACKTimeScale and ComnCntl2.QACKTimeValue, described further in Table 6-12. 3. Following the transition of the SA Request (SACntl.SAQREQ in Table 6-56) control bit from 0b to 1b, the SA must take no longer than 2 * <SA Quiesce Time> to set SAQACK. After 2 * <SA Quiesce Time>, CCIX Software detecting a zero returned for the SAQACK field indicates an error condition such that the SA was unable to reach a quiescent state. Following the transition of the SACntl.SAQREQ control bit from 1b to 0b, the SA must transition the SAQACK bit from 1b to 0b.	
15	**SAHWQACKCap** This field describes the SA's Hardware Quiesce Acknowledgement Capability. 0b: The SA does not have a hardware mechanism to achieve a quiescent state. 1b: The SA has a hardware mechanism to achieve a quiescent state.	RO
18:16	Reserved and Zero	RsvdZ
27:19	**SAMemPoolRdyTimeValue** This field describes the SA's Memory Pool Readiness Time Value Encoding where the overall <SA Mem Pool Readiness Time Reported> is SAMemPoolRdyTimeValue*<SA Mem Pool Readiness Time Multiplier>. <SA Mem Pool Readiness Time Multiplier> is $32^{SAMemPoolRdyTimeScale}$ns. 000h – 1FFh: Encodings for SAMemPoolRdyTimeValue of 0 through 511 A SAMemPoolRdyTimeValue value of 000h indicates the Memory Pools are always ready to be discovered and configured, i.e. SAMemPoolRdyStat is always 1b. The <SA Mem Pool Readiness Time Reported> must be the longer of the following two readiness times: 1. The readiness time following a Conventional Reset. 2. The readiness time following a Function Level Reset. The validity of SAMemPoolRdyTimeValue is not subject to the SADiscRdyStat or SAMemPoolRdyStat indicators. SAMemPoolRdyTimeValue is valid at the same time the CCIX Protocol Layer DVSEC Header is valid.	RO
30:28	**SAMemPoolRdyTimeScale** This field describes the SA's Memory Pool Readiness Time Scale Encoding in order to generate the <SA Mem Pool Readiness Time Multiplier> where <SA Mem Pool Readiness Time Multiplier> is $32^{SAMemPoolRdyTimeScale}$ns. 0h – 7h: Encodings for SAMemPoolRdyTimeScale of 0 through 7 which allows for <SA Mem Pool Readiness Time Multiplier> values of 32ns through 34359738368ns.	RO

Bit Location	Register Description	Attributes
	The validity of SAMemPoolRdyTimeScale is not subject to the SADiscRdyStat or SAMemPoolRdyStat indicators. SAMemPoolRdyTimeScale is valid at the same time the CCIX Protocol Layer DVSEC Header is valid. During initialization, should an SA determine that it's capable of a lower readiness time for its MemPools, the SA is permitted to reduce its <SA Mem Pool Readiness Time Reported> by either reducing its SAMemPoolRdyTimeValue, SAMemPoolRdyTimeScale, or both. However, CCIX Configuration Software may use either the original or reduced <SA Mem Pool Readiness Time Reported>. An SA is not permitted to increase its <SA Mem Pool Readiness Time Reported>.	
31	**SAMemPoolRdyStat** This field describes the Subordinate Agent Memory Pool's Readiness Status. 0b: Indicates the Subordinate Agent's Memory Pool attributes are not ready to be discovered. 1b: Indicates the Subordinate Agent's Memory Pool capabilities are ready to be discovered, e.g. the Memory Pool Size(s) have been determined, but the Memory Pool(s) may not be trained and therefore, the final determination for Memory Pool Size(s) may not have occured. A SA indicates a SAMemPoolRdyStat value of 1b and SADiscRdyStat value of 0b when the Memory Pool size(s) and other attributes have been determined, but memory training has not been completed. When memory training has been completed and the final, post-training, Memory Pool size(s) have been accurately indicated, then the SA indicates a value of 1b for both SAMemPoolRdyStat and SADiscRdyStat. A SA must take no longer than <SA Mem Pool Readiness Time Reported> to set SAMemPoolRdyStat. After <SA Mem Pool Readiness Time Reported>, CCIX Software detecting a zero returned for the SAMemPoolRdyStat field indicates that the SA cannot be configured/enumerated.	RO

The Capabilities & Status Register at Byte Offset-08h contains the SA Error Log Offset which is described in CCIX RAS Overview. The remaining bits in this register are Reserved and Zero.

6.2.2.8.4 Subordinate Agent Control Structure

Figure 6-72 shows the overall layout of the Subordinate Agent Control structure. The SA BAT Entry structure is described in Section 6.2.2.4.2.1.

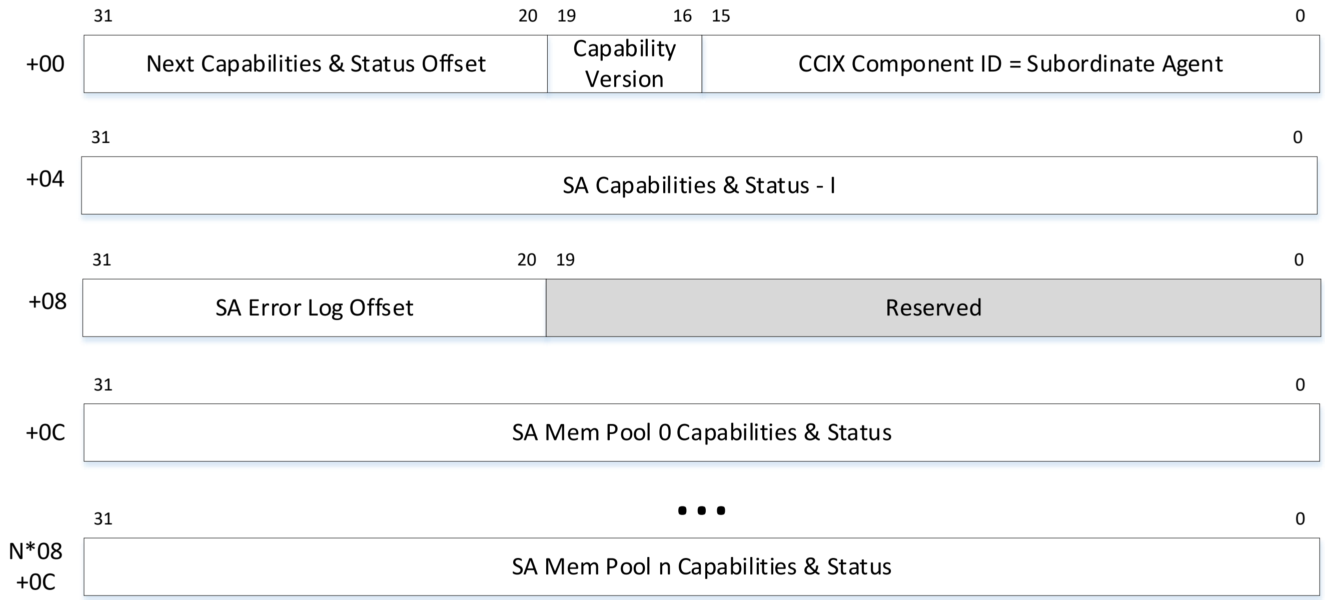

31
20
19
16
15
0
+00
Next Capabilities & Status Offset
Capability Version
CCIX Component ID = Subordinate Agent
31
0
+04
SA Capabilities & Status - I
31
20
19
0
+08
SA Error Log Offset
Reserved
31
0
+0C
SA Mem Pool 0 Capabilities & Status
• • •
31
0
N*08 +0C
SA Mem Pool n Capabilities & Status

Figure 6-72: Subordinate Agent Control Structure

Figure 6-73 shows the layout of the Subordinate Agent Control (SACntl) Register at Byte Offset-04h.

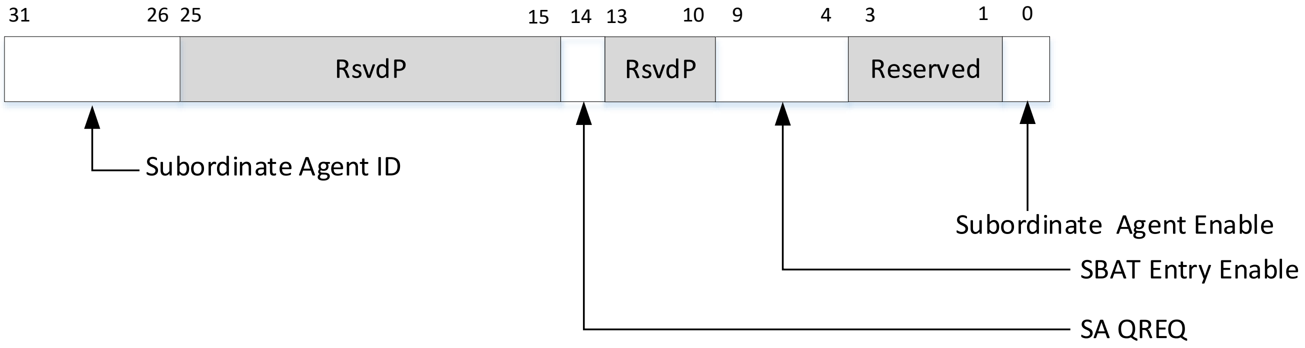


Figure 6-73: SACntl Register at Byte Offset-04h

Table 6-59 describes the SACntl Register fields at Byte Offset-04h.

5 **Table 6-59: SACntl Register Fields at Byte Offset-04h**

Bit Location	Register Description	Attributes
0	**SAEnable** This field controls enabling the Subordinate Agent. 0b: Indicates either the SA has not been configured, or the previously configured SA has been taken offline. 1b: Indicates the SA is enabled. The SA has been configured and must service Requests from an HA. CCIX Device initializes to 0b after reset (except FLR).	RW
3:1	Reserved and Preserved	RsvdP
9:4	**SBATDepthEnable** This field describes the number of Memory Pools enabled and therefore the number of unique G-HSAM Address windows that have been allocated to this SA. SBATDepthEnable therefore also indicates the corresponding number of SBAT Control entries. 00h: Control Structures do not have any SBAT Entries enabled. 01h: Control Structures have 1 SBAT Entry. … Nh: Control Structures have N SBAT Entries. An SA can have a maximum 63 SBAT Entries enabled. CCIX Device initializes to 00h after reset (except FLR).	RW
13:10	Reserved and Preserved	RsvdP
14	**SAQREQ** SA Quiesce Request controls when the SA will act to achieve a quiesced state. 0b: SA is not required to be quiesced. There can only be a change in the value, 0b or 1b, of the corresponding SACapStat.SAQACK bit after CCIX configuration software changes the value, 0b or 1b, of this SAQREQ control bit. 1b: SA must be quiesced. CCIX Confguration software must set the bit to a value 1b after first quiescing the HAs being serviced by this SA. CCIX Device initializes to 0b after reset (except FLR).	RW
25:15	Reserved and Preserved	RsvdP
31:26	**SAID** The Subordinate AgentID (SAID) is required to be unique across all Subordinate Agents in a CCIX system. Also, a CCIX AgentID is unique to a CCIX Device and cannot be re-used to enumerate a CCIX Agent on any other CCIX Device. 00h – 3Fh: Encodings for SAID0 through SAID63.	RW

The SA Error Control & Status Registers, SAErrCntlStat0 and SAErrCntlStat1, at Byte Offset-08h and Byte Offset-0Ch respectively, are described in CCIX RAS Overview.

The SBAT Control structure, located after the SA Error Control & Status Registers and illustrated in Figure 6-72, has the structure and definition of BAT Control structures described in detail in Section 6.2.2.4.2.

6.2.2.9 AF Properties Stuctures

CCIX Acceleration Function (AF) Properties Structures provide the ability to identify the binding between Acceleration Functions and the Request Agents servicing those Acceleration Functions, and the optional ability to control that binding. The AF Properties Structures must be located in Function 0 of the Primary CCIX Port. Secondary CCIX Ports do not have AF Properties Structures.

6.2.2.9.1 AF Properties Capabilities & Status Structure

Figure 6-74 shows the overall layout of the AF Properties Capabilities & Status Structure. There are pointers to three data structures contained within the AF Properties Capabilities & Status Structure. Two of the pointers are the RA Reference Index data structure, and the AFtoRA Binding Capability data structure.

The third pointer is to an optional AF Reference Index data structure. Section 6.2.2.9.1.2 describes the conditions under which this optional structure is declared.

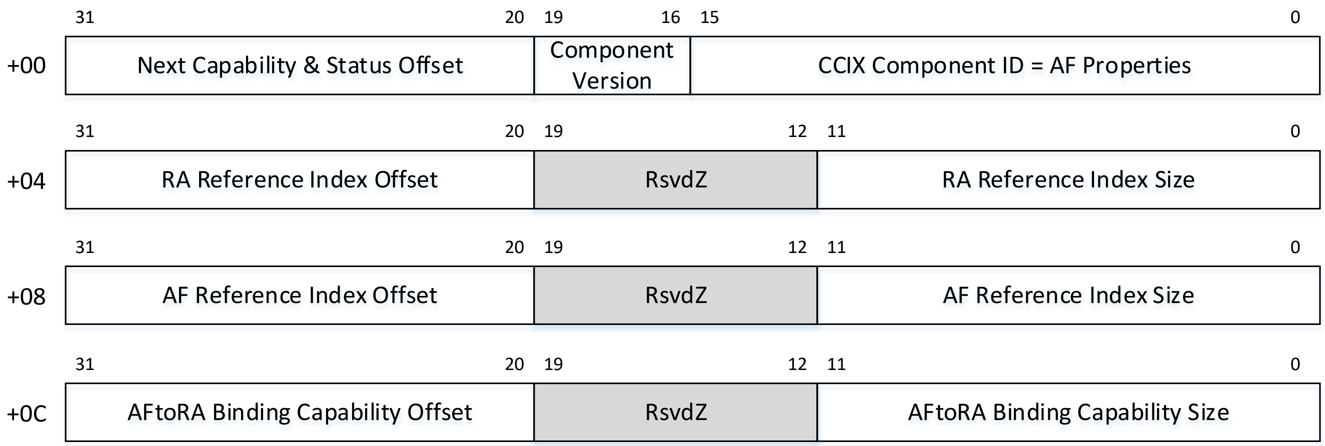

31
20
19
16
15
0
+00
Next Capability & Status Offset
Component Version
CCIX Component ID = AF Properties
31
20
19
12
11
0
+04
RA Reference Index Offset
RsvdZ
RA Reference Index Size
31
20
19
12
11
0
+08
AF Reference Index Offset
RsvdZ
AF Reference Index Size
31
20
19
12
11
0
+0C
AFtoRA Binding Capability Offset
RsvdZ
AFtoRA Binding Capability Size

Figure 6-74: AF Properties Capabilities & Status Structure

Table 6-60 describes the fields of the RA Reference Index Pointer (RARefIndexPtr) Register at Byte Offset 04h of the AF Properties Capabilities & Status Structure.

Table 6-60: RARefIndexPtr Register fields

Bit Location	Register Description	Attributes
11:0	**RARefIndexSize** This field indicates the RA Reference Index structure size in number of DW.	RO
19:12	Reserved and Zero	RsvdZ
31:20	**RARefIndexOffset** This field indicates the RA Reference Index structure Offset in number of bytes. The offset must be in integer multiples of DW. The offset + size must remain within the size of CCIX Protocol Layer DVSEC.	RO

Table 6-61 describes the fields of the AF Reference Index Pointer (AFRefIndexPtr) Register at Byte Offset 08h of the AF Properties Capabilities & Status Structure.

Table 6-61: AFRefIndexPtr Register fields

Bit Location	Register Description	Attributes
11:0	**AFRefIndexSize** This field indicates the AF Reference Index structure size in number of DW. An AFRefIndexSize value of 000h and an AFRefIndexOffset value of 000h, i.e. a null AFRefIndexPtr, indicates that this CCIX Device does not have the optional AF Reference Index data structure.	RO
19:12	Reserved and Zero	RsvdZ
31:20	**AFRefIndexOffset** This field indicates the AF Reference Index structure Offset in number of bytes. The offset must be in integer multiples of DW. The offset + size must remain within the size of CCIX Protocol Layer DVSEC. An AFRefIndexSize value of 000h and an AFRefIndexOffset value of 000h, i.e. a null AFRefIndexPtr, indicates that this CCIX Device does not have the optional AF Reference Index data structure.	RO

Table 6-62 describes the fields of the AFtoRA Binding Capability Pointer (AFtoRABindingCapPtr) Register at Byte Offset 0Ch of the AF Properties Capabilities & Status Structure.

Table 6-62: AFtoRABindingCapPtr Register fields

Bit Location	Register Description	Attributes
11:0	**AFtoRABindingCapSize** This field indicates the AFtoRA Binding Capability structure size in number of DW.	RO
19:12	Reserved and Zero	RsvdZ
31:20	**AFtoRABindingCapOffset** This field indicates the AFtoRA Binding Capability structure Offset in number of bytes. The offset must be in integer multiples of DW. The offset + size must remain within the size of CCIX Protocol Layer DVSEC.	RO

6.2.2.9.1.1 RA Reference Index Structure

Figure 6-75 shows the overall layout of the RA Reference Index Structure. The RA Reference Index Structure contains at least one RA Reference Index Entry, and the number of entries is based on the number of RAs in the CCIX Device, with a maximum of 32 RAs per CCIX Device. Each RA Reference Index Entry indicates the location of a RA DVSEC on that Device. Each RA Reference Index Entry must be unique, i.e., the location of a RA DVSEC on that Device is not repeated within the RA Reference Index Structure. The RA Reference Index Number, starting with RA Reference Index 0, is based on the position of the RA Reference Index Entry within the RA Reference Index Structure. Thus, RA Reference Index Entries 0 to n in the RA Reference Index Structure are associated with RA Reference Index Number 0 to n. The RA Reference Index Number is used as a reference pointer to the RA indicated in the RA Reference Index Structure in both the AFtoRA Binding Capability structure described in Section 6.2.2.9.1.3, as well as the AF Binding Control Entries described in Section 6.2.2.9.2.1.

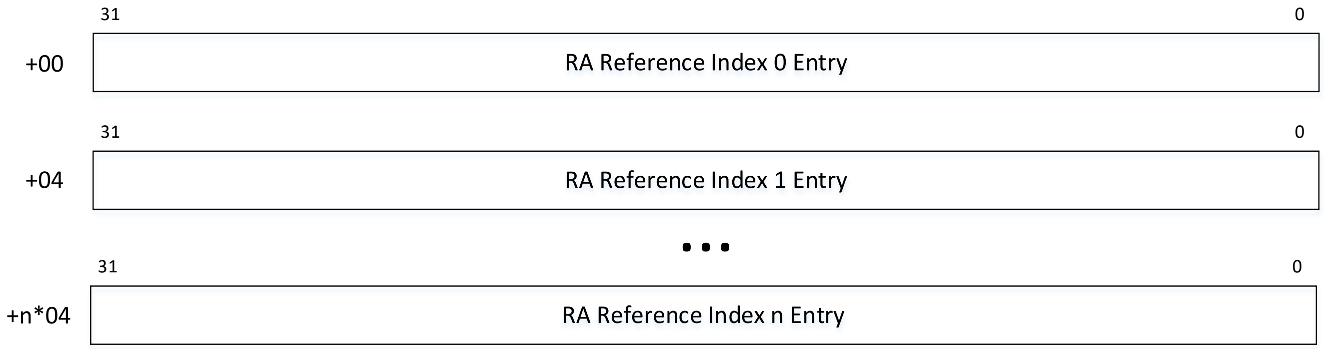

31
0
+00
RA Reference Index 0 Entry
31
0
+04
RA Reference Index 1 Entry
• • •
31
0
+n*04
RA Reference Index n Entry

Figure 6-75: RA Reference Index Structure

Figure 6-76 illustrates the layout of an RA Reference Index Entry.

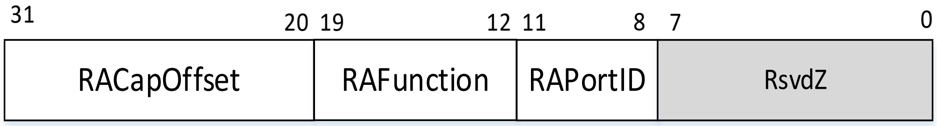


Figure 6-76: RA Reference Index Entry

Table 6-63 describes the register fields of the RA Reference Index Entry.

Table 6-63: RA Reference Index Entry Register Index fields

Bit Location	Register Description	Attributes
7:0	Reserved and Preserved	RsvdZ
11:8	**RAPortID** Indicates the CCIX PortID of the RA to which an AF to RA binding exists. 0h – Fh: Encodings for RAPortID values from 0 to 15.	RO
19:12	**RAFunction** Indicates the PCIe Function number of the RA to which an AF to RA binding exists, the function being located at the RAPortID declared in this entry. 00h – FFh: Encodings for RAFunction values from 0 to 255. To accommodate PCIe ARI, RAFunction supports 256 Functions.	RO
31:20	**RACapOffset** Indicates the DVSEC Capability Offset of the RA to which an AF to RA binding exists, the RACapOffset being relative to the RAPortID and RAFunction declared in this entry.	RO

6.2.2.9.1.2 AF Reference Index Structure

The AF Reference Index Structure is an optional structure for identifying Acceleration Functions using a PCIe framework. The AF Reference Index Structure mechanism relies on the PCIe Device and Function number (or PCIe Function number if ARI is supported), and the PCIe Port of the AF being sufficient to uniquely distinguish one Acceleration Function from another.

However, alternate frameworks that do not rely on PCIe Bus, Device, and Function number as the sole mechanism to delineate a unique Acceleration Function are not required to define the AF Reference Index Structure, and indicate that by providing a null AF Reference Index Pointer.

Figure 6-77 shows the overall layout of the AF Reference Index Structure. The number of entries in the AF Reference Index Structure is based on the number of AFs in the CCIX Device. Each AF Reference Index Entry indicates the Port and Physical Function of the AF on that Device. Each AF Reference Index Entry must be unique, i.e. the Port and Physical Function of the AF on that Device is not repeated within the AF Reference Index Structure. The AF Reference Index Number, starting with AF Reference Index 0, is based on the position of the AF Reference Index Entry within the AF Reference Index Structure. Thus, AF Reference Index Entries 0 to n in the AF Reference Index Structure result in AF Reference Index Number 0 to n. The AF Reference Index Number is subsequently used in both the AFtoRA Binding Capability structure, described in Section 6.2.2.9.1.3, as well as the AF Binding Control Entries, described in Section 6.2.2.9.2.1, as a reference pointer to the AF described in the AF Reference Index Structure.

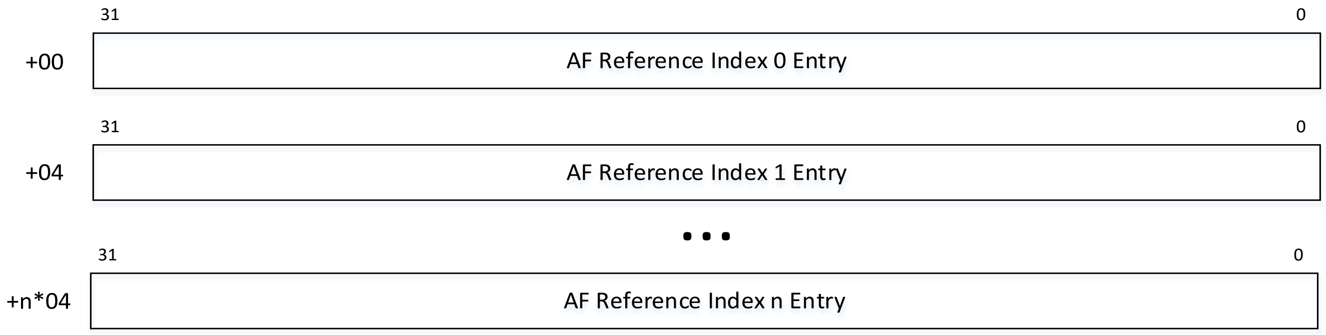

31
0
+00
AF Reference Index 0 Entry
31
0
+04
AF Reference Index 1 Entry
• • •
31
0
+n*04
AF Reference Index n Entry

Figure 6-77: AF Reference Index Structure

Figure 6-78 illustrates the layout of an AF Reference Index Entry.

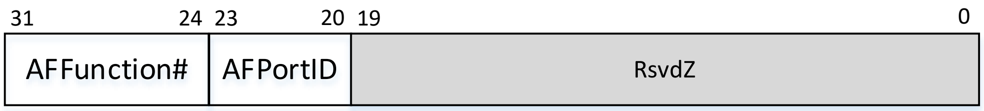


Figure 6-78: AF Reference Index Entry

Table 6-64 describes the register fields of the AF Reference Index Entry.

Table 6-64: AF Reference Index Entry Register fields

Bit Location	Register Description	Attributes
19:0	Reserved and Preserved	RsvdZ
23:20	**AFPortID** Indicates the CCIX PortID of the AF. 0h – Fh: Encodings for AFPortID values from 0 to 15.	RO
31:24	**AFFunctionNumber** Indicates the PCIe Function number of the AF. 00h – FFh: Encodings for AFFunctionNumber values from 0 to 255. To accommodate PCIe ARI, AFFunctionNumber supports 256 Functions.	RO

6.2.2.9.1.3 AF to RA Binding Capability Structure

The AF to RA Binding Capability Structure provides binding information between the AFs and RAs on a CCIX Device. An RA having binding to an AF means that RA is capable of servicing CCIX traffic on behalf of that AF. The

AF to RA Binding Capability Structure comprehends that an RA is capable of concurrently servicing CCIX traffic on behalf of multiple AFs. The AF to RA Binding Structure also comprehends that an AF may have a binding to multiple RAs.

Figure 6-79 shows the overall layout of the AF to RA Binding Capability Structure, indexed by the Acceleration Function Reference Index. The AF to RA Binding Capability Structure contains at least one AF to RA Binding Capability Entry, and the number of entries is based on the number of AFs in the CCIX Device. The AF Reference Indices in the AF to RA Binding Capability Structure are linearly enumerated with AF Reference Indices starting with an index value of 0. Thus, AF to RA Binding Capability Entries 0 to n describe the binding capabilities of AF Reference Indices 0 to n.

The AF Reference Index, used to index into the AF to RA Binding Capability Structure, uniquely identifies the CCIX AF on the CCIX Device. An AF Reference Index is either derived using the PCIe based framework described in Section 6.2.2.9.1.2, or an AF Reference Index is derived using alternative frameworks outside the scope of this specification. However, alternative frameworks must be consistent with respect to the AF associated with the AF Reference Index and the associated binding of that AF to the RA described in the AF to RA Binding Capability Structure.

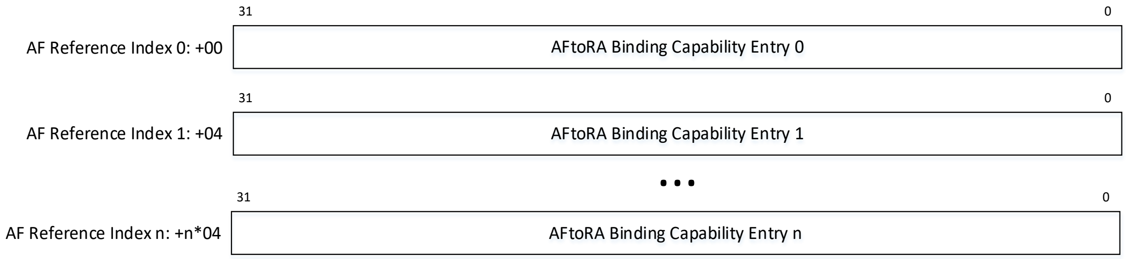

31
0
AF Reference Index 0: +00
AFtoRA Binding Capability Entry 0
31
0
AF Reference Index 1: +04
AFtoRA Binding Capability Entry 1
• • •
31
0
AF Reference Index n: +n*04
AFtoRA Binding Capability Entry n

Figure 6-79: AF to RA Binding Capability Structure

Figure 6-80 illustrates the layout of the AF to RA Binding Capability Entry.

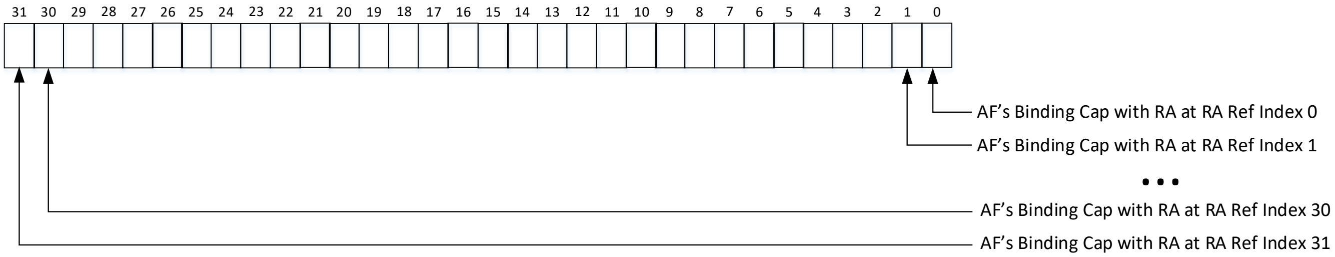

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
AF’s Binding Cap with RA at RA Ref Index 0
AF’s Binding Cap with RA at RA Ref Index 1
• • •
AF’s Binding Cap with RA at RA Ref Index 30
AF’s Binding Cap with RA at RA Ref Index 31

Figure 6-80: AF to RA Binding Capability Entry

Table 6-65 describes the register fields of the AF to RA Binding Capability Entry.

Table 6-65: AF to RA Binding Capability Entry Register fields

Bit Location	Register Description	Attributes
31:0	**AFtoRABindingCapVctr** AFtoRABindingCapVctr indicates the RAs that have a binding to the AF with the AF Reference Index associated with this entry. AFtoRABindingCapVctr[0] to AFtoRABindingCapVctr[31] provide the binding capabilities of the RAs located at RA Reference Index Entry 0 to 31 described in Section 6.2.2.9.1.1. 0b: • Indicates that the RA associated with this bit-position does not have binding to this AF, i.e. the RA cannot service CCIX traffic on behalf of this AF. 1b: • Indicates that the RA associated with this bit-position has binding to this AF, i.e. the RA can service CCIX traffic on behalf of this AF. At least one bit position of the AFtoRABindingCapVctr must be set, i.e. at least one RA must have a binding to the AF with the AF Reference Index associated with this entry. It is permitted for multiple bit positions of the AFtoRABindingCapVctr to be set, i.e. multiple RAs may have binding to the AF with the AF Reference Index associated with this entry. It is also permitted for multiple AF to RA Binding Capability entries to have the same AFtoRABindingCapVctr bit-position set, i.e. a RA can service CCIX traffic on behalf of multiple AFs.	RO

6.2.2.9.2 AF to RA Properties Control Structure

Figure 6-81 shows the overall layout of the AF Properties Control Structure. Following the AF Properties Control Structure header are the AF to RA Binding Control Entries, described further in Section 6.2.2.9.2.1.

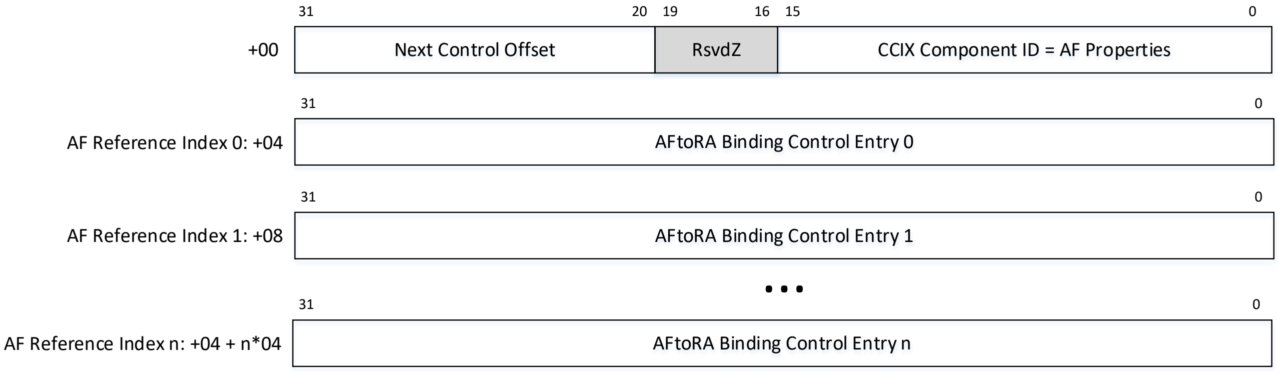

31
20
19
16
15
0
+00
Next Control Offset
RsvdZ
CCIX Component ID = AF Properties
31
0
AF Reference Index 0: +04
AFtoRA Binding Control Entry 0
31
0
AF Reference Index 1: +08
AFtoRA Binding Control Entry 1
• • •
31
0
AF Reference Index n: +04 + n*04
AFtoRA Binding Control Entry n

Figure 6-81: AF Properties Control Structure

6.2.2.9.2.1 AF to RA Binding Control Entries

AF to RA Binding Control Entries provide the optional ability to control binding between the AFs and RAs on a CCIX Device. This means that an AF to RA Binding Control Entry optionally controls whether a RA can service CCIX traffic on behalf of an AF. AF to RA Binding Control Entries allow optional control of a single RA's binding to multiple AFs. An AF to RA Binding Entry also allows optional control of an AF's binding to multiple RAs.

Figure 6-81 shows the overall layout of the AF to RA Binding Control Entries, indexed by the AF Reference Index. There must be at least one AF to RA Binding Control Entry, located at Byte Offset 04h of the AF Properties Control Structure. The number of AF to RA Binding Control Entries is based on the number of AFs in the CCIX Device. The AF to RA Binding Control Entries 0 to n control the binding of AFs associated with AF Reference Index 0 to n.

Figure 6-82 illustrates the layout of an AF to RA Binding Control Entry.

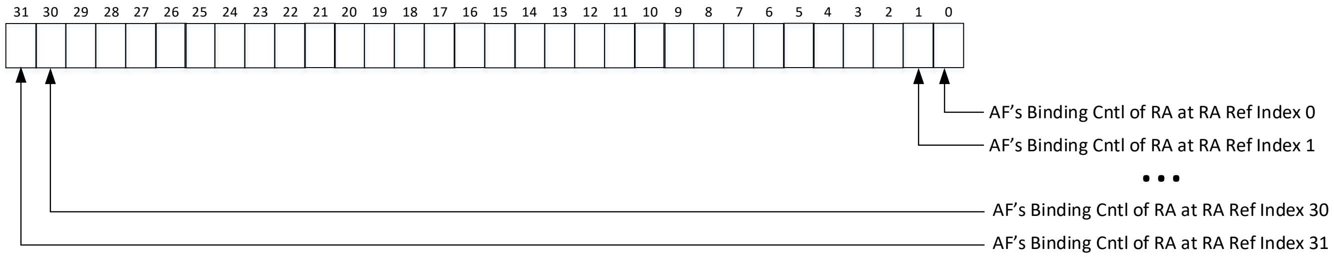

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
AF's Binding Cntl of RA at RA Ref Index 0
AF's Binding Cntl of RA at RA Ref Index 1
• • •
AF's Binding Cntl of RA at RA Ref Index 30
AF's Binding Cntl of RA at RA Ref Index 31

Figure 6-82: AF to RA Binding Control Entry

Table 6-66 describes the register fields of an AF to RA Binding Control Entry.

Table 6-66: AF to RA Binding Control Entry Register fields

Bit Location	Register Description	Attributes
31:0	**AFtoRABindingCntlVctr** AFtoRABindingCntlVctr optionally controls whether a RA has a binding to the AF with the AF Reference Index associated with this entry. AFtoRABindingCntlVctr[0] to AFtoRABindingCntlVctr[31] control the binding of the RAs associated with RA Reference Index 0 to 31 described in Section 6.2.2.9.1.1. The initial value out of reset of AFtoRABindingCntlVctr must be the same as the value of AFtoRABindingCapVctr, described in Section 6.2.2.9.1.3. This means that the initial value of the control vector reflects the AF to RA binding capabilities of the AF with the AF Reference Index associated with this entry. If a 0b value is written to a bit position in the AFtoRABindingCntlVctr, different from the 1b value of that same bit position in the AFtoRABindingCapVctr, and a subsequent read of the AFtoRABindingCntlVctr indicates that the value has not changed and remains 1b, then the CCIX Device does not support disabling of the binding of the RA associated with that bit-position. 0b: • Indicates that the binding of the RA associated with this bit-position is disabled to this AF, i.e. the RA cannot service CCIX traffic on behalf of this AF. • Acceleration Frameworks may choose to quiesce an AF with respect to its interactions with an RA prior to a 1b-to-0b transition of the bit-position of the AFtoRABindingCntlVctr associated with that RA. Acceleration Frameworks based on PCIe may choose to quiesce an AF using PCIe FLR. Other methods to queiesce an AF is outside the scope of this specification. 1b: • Indicates that the binding of the RA associated with this bit-position is enabled to this AF, i.e. the RA can service CCIX traffic on behalf of this AF.	RW

Bit Location	Register Description	Attributes
	A bit position of the AFtoRABindingCntlVctr must not be written with value 1b if the same bit position in the AFtoRABindingCapVctr is a value 0b. Setting AFtoRABindingCntlVctr to 0000h disables the servicing of all CCIX traffic on behalf of the AF with the AFID associated with this entry, if a subsequent read of the AFtoRABindingCntlVctr indicates a value of 0000h. Acceleration Frameworks may choose to reset an AF with respect to its interactions with an RA following a successful 1b-to-0b transition of the bit-position in the AFtoRABindingCntlVctr associated with that RA.	

6.2.2.10 DVM Agent Structure

Figure 6-83 shows the overall layout of the DVM Agent Status Structure.

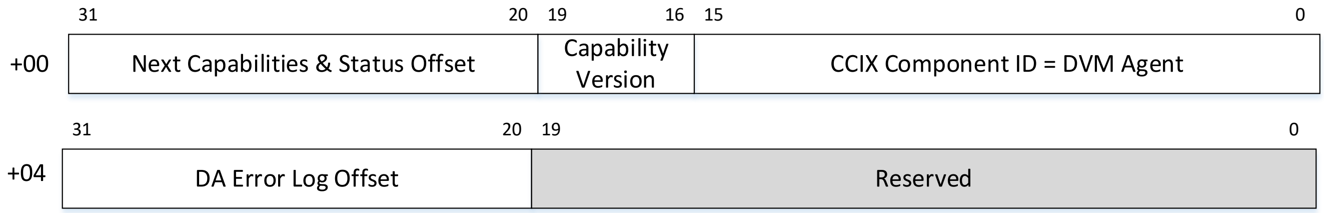
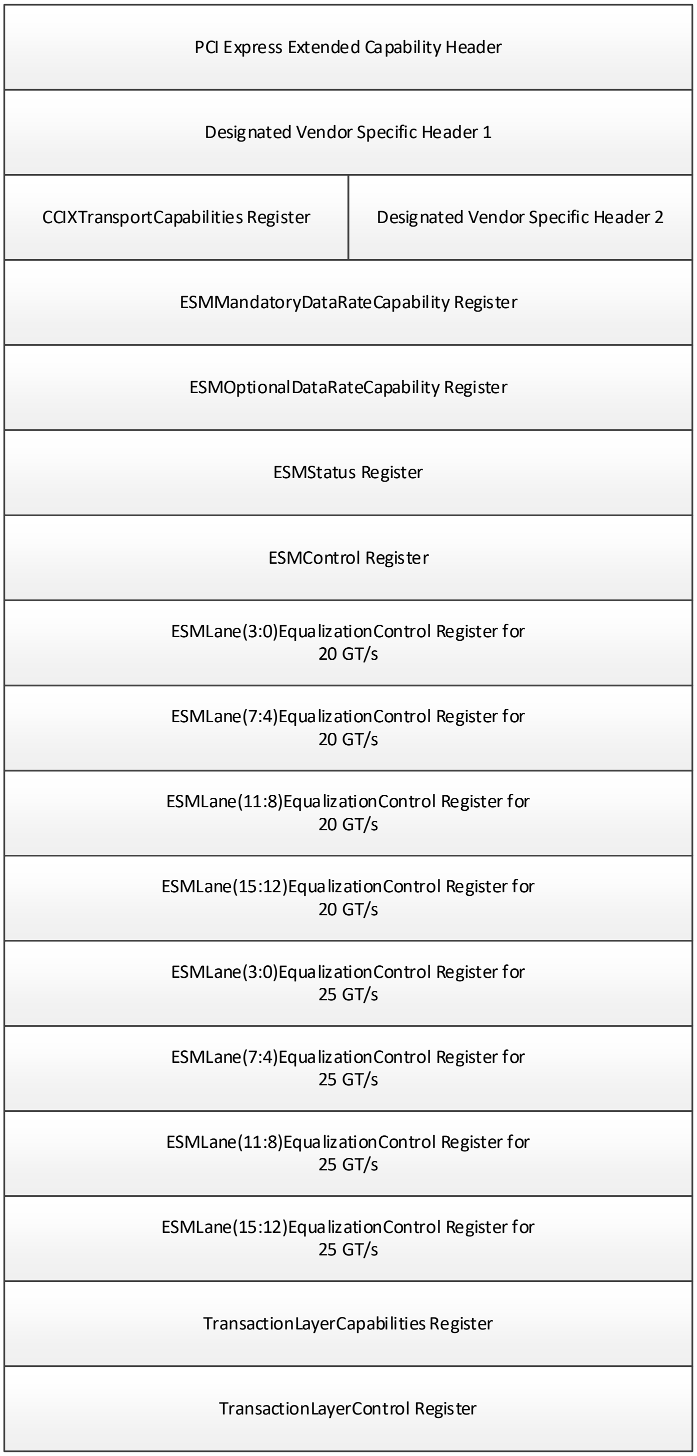


Figure 6-83: DVM Agent Status Structure

6.3 Transport DVSEC

Figure 6-84 shows the Transport DVSEC supported by CCIX devices.

The ESMMandatoryDataRateCapabilities, ESMOptionalDataRateCapabilities, ESMControl, ESMStatus, and ESMLaneEqualizationControl registers must only be implemented if the CCIXTransportCapabilities.ESMModeSupported bit is Set (see Section 4.3.2). The rest of the registers in the Transport DVSEC are at the same address offsets regardless of whether the ESM registers are implemented or not.



PCI Express Extended Capability Header

Designated Vendor Specific Header 1

CCIXTransportCapabilities Register | Designated Vendor Specific Header 2

ESMMandatoryDataRateCapability Register

ESMOptionalDataRateCapability Register

ESMStatus Register

ESMControl Register

ESMLane(3:0)EqualizationControl Register for 20 GT/s

ESMLane(7:4)EqualizationControl Register for 20 GT/s

ESMLane(11:8)EqualizationControl Register for 20 GT/s

ESMLane(15:12)EqualizationControl Register for 20 GT/s

ESMLane(3:0)EqualizationControl Register for 25 GT/s

ESMLane(7:4)EqualizationControl Register for 25 GT/s

ESMLane(11:8)EqualizationControl Register for 25 GT/s

ESMLane(15:12)EqualizationControl Register for 25 GT/s

TransactionLayerCapabilities Register

TransactionLayerControl Register

Figure 6-84: CCIX Transport DVSEC

6.3.1 CCIXTransportCapabilities Register

The CCIXTransportCapabilities register identifies ESM specific capabilities. Figure 6-85 details allocation of fields in this register and Table 6-67 provides the respective field definitions.

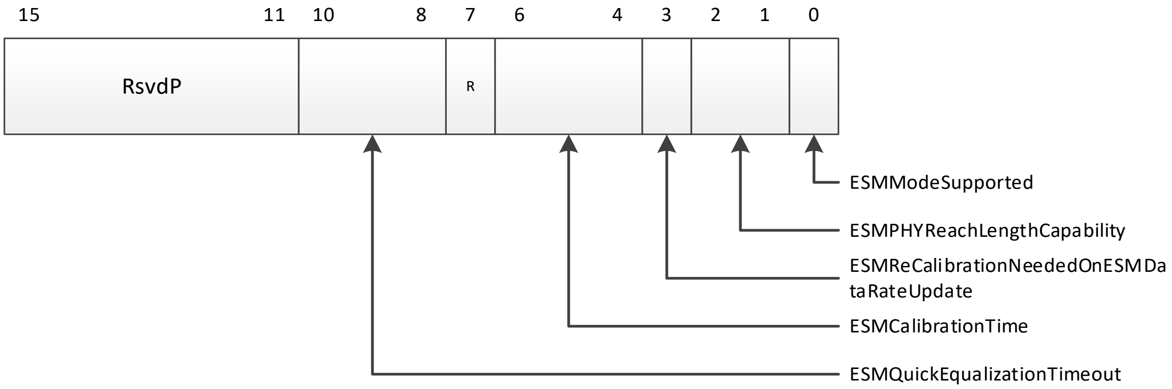

15
11
10
8
7
6
4
3
2
1
0
RsvdP
R
ESMModeSupported
ESMPHYReachLengthCapability
ESMReCalibrationNeededOnESMDataRateUpdate
ESMCalibrationTime
ESMQuickEqualizationTimeout

Figure 6-85: CCIXTransportCapabilities Register

Table 6-67: CCIXTransportCapabilities Register

Bit Location	Register Description	Attributes
0	**ESMModeSupported.** Set if Physical Layer supports ESM Mode, otherwise Clear.	HwInit
2:1	**ESMPHYReachLengthCapability.** Indicates the Reach Length Capability at ESM Data Rate1 only. Defined encodings are: 00b = PHY is Short Reach Capable 01b = PHY is Long Reach Capable 10b = PHY is Short Reach and Long Reach Capable 11b = Reserved If ESMModeSupported is Clear, this field is RsvdP. At 20.0 GT/s and 25.0 GT/s data rates, Long Reach Capable PHY must also be capable of driving short channels. 11b encoding is reserved for future use.	HwInit

Bit Location	Register Description	Attributes
3	**ESMReCalibrationNeededOnESMDataRateUpdate.** Set if the Physical Layer requires re-calibration if ESMControl.ESMDataRate0 or ESMDataRate1 fields are modified (to change Data Rate(s)), after initial or subsequent calibration and entry into LTSSM state L0 at the Data Rate(s) programmed in ESMDataRate1. Clear if the Physical Layer does not require re-calibration, after the initial calibration and if ESMControl.ESMDataRate0 or ESMDataRate1 fields are subsequently modified (to change Data Rate(s)). PHYs that do not need to be re-calibrated are permitted to hardwire this bit to 0b. If ESMModeSupported is Clear, this field is RsvdP.	HwInit
6:4	**ESMCalibrationTime.** Indicates the maximum time taken by the Physical Layer to complete the Calibration step. Defined encodings are: 000b = 10 us 001b = 50 us 010b = 100 us 011b = 500 us 100b = 1 ms 101b = 5 ms 110b = 10 ms 111b = 50 ms If ESMModeSupported is Clear, this field is RsvdP.	HwInit
7	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
10:8	**ESMQuickEqualizationTimeout.** Minimum Equalization Phase 2 and Phase 3 to Recovery Speed Timeout supported for Data Rates greater than 16.0 GTs/, when ESMControl.QuickEqualizationTimeoutSelect is programmed to a value greater than 000b. Defined encodings are: 000b = Quick Equalization is not supported by the device 001b = 8 ms / 16 ms 010b = 24 ms / 32 ms 011b = 50 ms / 58 ms 100b = 100 ms / 108 ms 101b = 200 ms / 208 ms All other encodings are Reserved. If supported, devices must advertise ESM Quick Equalization Timeout that is less than or equal to ESMControl.ESMExtendedEqualizationPhase2Timeout on the USP or ESMControl.ESMExtendedEqualizationPhase3Timeout on the DSP. If ESMModeSupported is Clear, this field is RsvdP.	HwInit
15:11	Reserved and Preserved	RsvdP

6.3.2 ESMMandatoryDataRateCapability Register

The ESMMandatoryDataRateCapability register advertises the ESM data rates supported.

Figure 6-86 details allocation of fields in this register and Table 6-68 provides the respective field definitions. This register is only implemented if ESMModeSupported is Set.

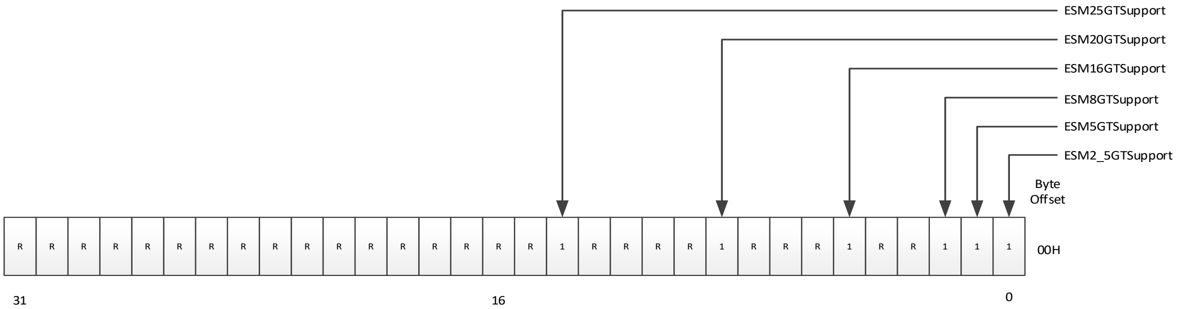


Figure 6-86: ESMMandatoryDataRateCapability Register

Table 6-68: ESMMandatoryDataRateCapability Register

Bit Location	Register Description	Attributes
0	ESM2_5GTSupport. Must be Set.	HwInit
1	ESM5GTSupport. Must be Set.	HwInit
2	ESM8GTSupport. This value is permitted to depend on the value programmed into ESMDataRate1. See Section 4.4.2.3 , Step 2e.	RO
4:3	Reserved	RsvdP
5	ESM16GTSupport. This value is permitted to depend on the value programmed into ESMDataRate1. See Section 4.4.2.3 , Step 2e.	RO
8:6	Reserved	RsvdP
9	ESM20GTSupport. This value is permitted to depend on the value programmed into ESMDataRate1. See Section 4.4.2.3 , Step 2e.	RO
13:10	Reserved	RsvdP
14	ESM25GTSupport. This value is permitted to depend on the value programmed into ESMDataRate1. See Section 4.4.2.3 , Step 2e.	RO
20:15	Reserved	RsvdP
21	ESM32GTSupport. This value is permitted to depend on the value programmed into ESMDataRate1. See Section 4.4.2.3 , Step 2e.	RO
24:22	Reserved	
25	ESM64GTSupport. This value is permitted to depend on the value programmed into ESMDataRate1. See Section 4.4.2.3 , Step 2e.	RO
31:26	Reserved	RsvdP

6.3.3 ESM OptionalDataRateCapability Register

The ESMOptionalDataRateCapability register enables a device to advertise optional ESM data rates supported. Currently, there are no optional data rates defined; future releases of the specification may define optional data rates.

Table 6-69 provides the respective field definitions. This register is only implemented if ESMModeSupported is Set.

Table 6-69: ESMOptionalDataRateCapability Register

Bit Location	Register Description	Attributes
31:0	Reserved	RsvdP

6.3.4 ESMStatus Register

The ESMStatus register provides information about ESM specific parameters. Figure 6-87 details allocation of fields in this register and Table 6-70 provides the respective field definitions. This register is only implemented if ESMModeSupported is Set.

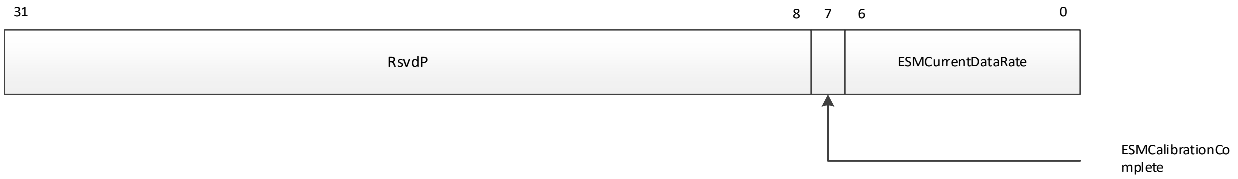


Figure 6-87: ESMStatus Register

Table 6-70: ESMStatus Register

Bit Location	Register Description	Attributes
6:0	**ESMCurrentDataRate.** Hardware indicates the negotiated ESM data rate. Defined encodings are: 000 0000b = ESM Inactive 000 0001b = 2.5 GT/s 000 0010b = 5.0 GT/s 000 0011b = 8.0 GT/s 000 0110b = 16.0 GT/s 000 1010b = 20.0 GT/s 000 1111b = 25.0 GT/s 001 0110b = 32.0 GT/s 011 0110b = 64.0 GT/s All other encodings are Reserved. Default value of this field is 00 0000b.	RO
7	**ESMCalibrationComplete.** Set after ESM physical layer calibration is complete. Physical layer calibration is initiated by Setting ESMControl.ESMPerformCalibration. Default value of this bit is 0b.	RO
31:8	Reserved and Preserved	RsvdP

6.3.5 ESMControl Register

The ESMControl register controls ESM specific parameters.

Figure 6-88 details allocation of fields in this register and Table 6-71 provides the respective field definitions. This register is only implemented if ESMModeSupported is Set.

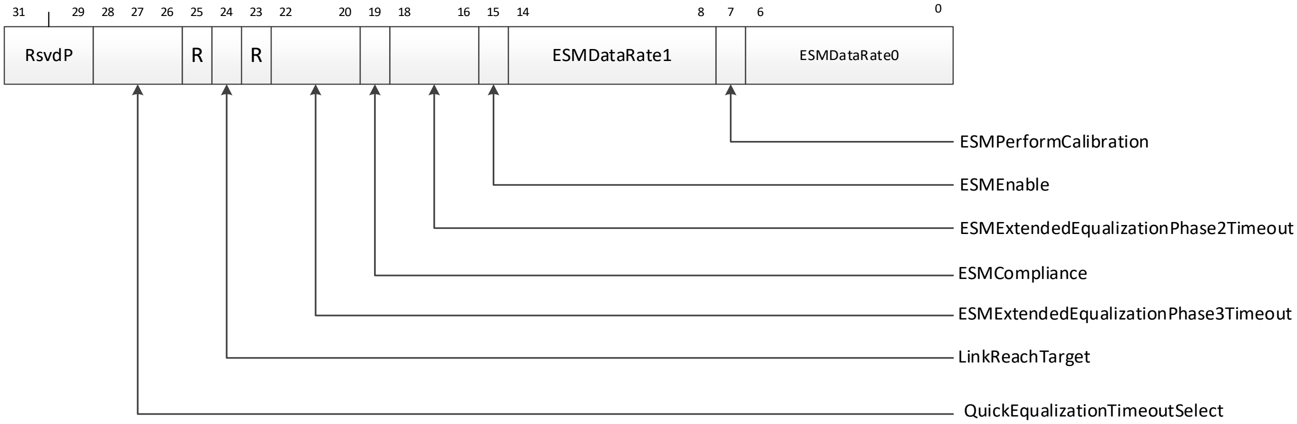


Figure 6-88: ESMControl Register

Table 6-71: ESMControl Register

Bit Location	Register Description	Attributes
6:0	**ESMDataRate0.** Sets the ESM Data Rate for ESM data rates associated with the 8.0 GT/s Data Rate Identifer. Programmed by SSW before entering ESM. The programmed data rate must be a supported data rate in either the ESMMandatoryDataRateCapability or ESMOptionalDataRateCapability registers, otherwise the result is undefined. Defined encodings are: 000 0000b = No Speed 000 0011b = 8.0 GT/s 000 0110b = 16.0 GT/s 000 1010b = 20.0 GT/s 000 1111b = 25.0 GT/s 001 0110b = 32.0 GT/s 011 0110b = 64.0 GT/s All other encodings are Reserved. Only certain settings for this field are permitted to be programmed. If any other settings are used the behavior is undefined. See Section 6.3.5.1 for more details. When ESMCompliance is Set, this field is RWS, else it is RW. Default value of this field is 000 0000b.	RWS/RW

Bit Location	Register Description	Attributes
7	**ESMPerformCalibration.** A write of 1b initiates the start of physical layer calibration before transition to ESM. Physical layer calibration is performed after the Link has transitioned to a non-D0 state through Programmed Power Management (Link enters the LTSSM L1 state when idle). Hardware sets ESMStatus.ESMCalibrationComplete to indicate that physical layer calibration is complete. When ESMCompliance is Set, this field is RWS, else it is RW.This bit always returns 0b when read.	RWS/RW
14:8	**ESMDataRate1.** Sets the ESM Data Rate for ESM data rates associated with the 16.0 GT/s Data Rate Identifer. . Programmed by SSW before entering ESM. The programmed data rate must be a supported data rate in either the ESMMandatoryDataRateCapability or ESMOptionalDataRateCapability registers, otherwise the result is undefined. Defined encodings are: 000 0000b = No Speed 000 0110b = 16.0 GT/s 000 1010b = 20.0 GT/s 000 1111b = 25.0 GT/s 001 0110b = 32.0 GT/s 011 0110b = 64.0 GT/s All other encodings are Reserved. Only certain settings for this field are permitted to be programmed. If any other settings are used the behavior is undefined. See Section 6.3.5.1 for more details. When ESMCompliance is Set, this field is RWS, else it is RW. Default value of this field is 000 0000b.	RWS/RW
15	**ESMEnable**. When the bit is 0b, a write of 1b initiates the transition to ESM, when the Link is operating at 2.5 GT/s or 5.0 GT/s, and the Link is operating in PCI Express Mode. When the bit is 0b, writing this bit to 1b when the Data Rate is not 2.5 GT/s and not 5 GT/s, or not operating in PCI Express Mode, produces undefined results. When ESMCompliance is Set, this field is RWS, else it is RW. Default value is of this bit is 0b.	RWS/RW

Bit Location	Register Description	Attributes
18:16	**ESMExtendedEqualizationPhase2Timeout.** LTSSM Equalization Phase2 Timeout For Upstream Port / Downstream Port when the current data rate is higher than 16.0 GT/s. Defined encodings are: 000b = 24 ms / 32 ms 001b = 50 ms / 58 ms 010b = 100 ms / 108 ms 011b = 200 ms / 208 ms 100b = 400 ms / 408 ms 101b = 600 ms / 608 ms All other encodings are Reserved. 1. For an USP, if CCIXTransportCapabilities.ESMPHYReachLengthCapability is a defined value other than 00b, hardware must initialize this field to indicate its Equalization Phase2 Timeout requirement for a Long Reach Link during ESM Data Rate1 link initialization. 2. For an USP, if CCIXTransportCapabilities.ESMPHYReachLengthCapability is 00b (PHY is only Short Reach Capable) hardware must initialize this field to 000b. 3. For a DSP, SSW must program this field, taking into account the value of the LinkReachTarget bit and the requested value of the ESMExtendedEqualizationPhase2Timeout field from the USP.	RW/RO
19	**ESMCompliance.** This bit must be Set when performing certain compliance testing. Requirements related to the ESMCompliance bit will be called out by individual compliance procedures. Default value is of this bit is 0b.	RWS

Bit Location	Register Description	Attributes
22:20	**ESMExtendedEqualizationPhase3Timeout.** LTSSM Equalization Phase3 Timeout For Downstream Port / Upstream Port when the current data rate is higher than 16.0 GT/s. Defined encodings are: 000b = 24 ms / 32 ms 001b = 50 ms / 58 ms 010b = 100 ms / 108 ms 011b = 200 ms / 208 ms 100b = 400 ms / 408 ms 101b = 600 ms / 608 ms All other encodings are Reserved. 1. For a DSP, if CCIXTransportCapabilities.ESMPHYReachLengthCapability is a defined value other than 00b, hardware must initialize this field to indicate its Equalization Phase3 Timeout requirement, for a Long Reach Link during ESM Data Rate1 link initialization. 2. For a DSP, if CCIXTransportCapabilities.ESMPHYReachLengthCapability is 00b (PHY is only Short Reach Capable), hardware must initialize this field to 000b 3. For an USP, SSW must program this field taking into account the value of the LinkReachTarget bit and the requested value of ESM Extended Equalization Phase3 Timeout value from the DSP.	RW/RO
23	Reserved and Preserved	RsvdP
24	**LinkReachTarget**. This bit indicates the Reach Length Target of current link. Defined encodings are: 0b = Reach Length Target of current link is Short Reach (SR) 1b = Reach Length Target of current link is Long Reach (LR) When ESMCompliance is Set, this field is RWS, else it is RW. The default value of this bit is 0b. SSW should configure this bit to indicate the target reach length of the Link before initiating the ESM physical layer calibration, based on the platform channel capability (Compliance to SR or LR) and ESMPHYReachLengthCapability of the DSP and USP.	RWS/RW
25	Reserved and Preserved	RsvdP

Bit Location	Register Description	Attributes
28:26	**QuickEqualizationTimeoutSelect.** Defined encodings are: 000b = Quick Equalization is Disabled 001b = 8 ms / 16 ms 010b = 24 ms / 32 ms 011b = 50 ms / 58 ms 100b = 100 ms / 108 ms 101b = 200 ms / 208 ms All other encodings are Reserved. The default value of this field is equal to a value in CCIXLinkTransportCapabilities.ESMQuickEqualizationTimeout. For Data Rates higher than 16.0 GT/s, if CCIXTransportCapabilities.ESMQuickEqualizationTimeout is greater than 000b, setting QuickEqualizationTimeoutSelect to value greater than 000b causes the corresponding QuickEqualizationTimeout value to be used during the Recovery Equalization Phase2/Phase3 to Recovery Speed transition in place of the Timeouts specified in the ESMControl.ESMExtendedEqualizationPhase2Timeout and ESMControl.ESMExtendedEqualizationPhase3Timeout fields. Setting QuickEqualizationTimeoutSelect to 000b disables the Quick Equalization mechanism. If the QuickEqualizationTimeoutSelect field is to be programmed to a non-zero value, it must be programmed to an encoding corresponding to a Timeout equal to or greater than that advertised in CCIXTransportCapabilities.ESMQuickEqualizationTimeout, otherwise the resulting behavior is undefined. Setting ESMControl.QuickEqualizationTimeoutSelect to a value other than 000b has no effect on behavior if CCIXTransportCapabilities.ESMQuickEqualizationTimeout is 000b. See Implementation Note “ESM Control Register Quick Equalization Timeout Select field” for usage guidelines.	RW
31:29	Reserved and Preserved	RsvdP

6.3.5.1 Rules for Programming ESM Fields

Rules for programming the ESMControl.ESMDataRate0 and ESMDataRate1 fields.

1 In the presence of Retimer(s), ESMDataRate0 and ESMDataRate1 must not be programmed to 20.0 GT/s or 25.0 GT/s .

2 Before Setting ESMControl.ESMEnable, ESMDataRate0 and ESMDataRate1 must be programmed.

3 After the initial transitions to the data rates programmed in the ESMDataRate0 and ESMDataRate1 fields, link speed transitions are performed to either ESM Data Rate0 or ESM Data Rate1 without performing link equalization, as long as LinkUp=1 and neither side requests that link equalization be redone. These link speed transitions are permitted to be entered using the Downstream Port “Target Link Speed” mechanism

defined in the *PCI Express Base Specification*, Link Control 2 register, where ESM Data Rate0 corresponds to 8.0 GT/s operation, and ESM Data Rate1 corresponds to 16.0 GT/s operation. Link speed transitions are permitted to be initiated by implementation specific mechanisms.

4 A link speed transition to 2.5 GT/s or 5.0 GT/s from the data rates programmed in ESMDataRate0 or ESMDataRate1 is permitted to be initiated by using the Downstream Port “Target Link Speed” mechanism defined in the *PCI Express Base Specification*, Link Control 2 register or an implementation specific mechanism.

5 When operating in ESM mode, if link operation is desired at a data rate that differs from those currently programmed in the ESMControl.ESMDataRate0 or ESMDataRate1 fields, then the Link speed must first be transitioned to 2.5 GT/s using the Downstream Port “Target Link Speed” mechanism defined in the *PCI Express Base Specification*, Link Control 2 register or an implementation specific mechanism. SSW Clears ESMControl.ESMEnable on both the DSP and USP. SSW must read the CCIXCapabilities.ESMReCalibrationNeededonESMDataRateUpdate bit on both the DSP and USP. If the CCIXCapabilities.ESMReCalibrationNeededonESMDataRateUpdate bit is Clear on both the DSP and USP, Section 4.4.2.3 , Step 2c), 2d), 2e) and 2m) are performed, followed by, Step 3. If the CCIXCapabilities.ESMReCalibrationNeededonESMDataRateUpdate bit is Set on the DSP or the USP, Section 4.4.2.3 Step 2c), 2d), 2e), 2f), 2g), 2h), 2i), 2j), 2k), 2l), and 2m) are performed, followed by, Step 3.

6 Since Rule 5 is a re-entry into ESM mode with different values programmed into ESMControl.ESMDataRate0/1 entry into LTSSM Equalization states (with ESMControl.ESMExtendedEqualizationPhase2/3Timeout) is required. Time taken to perform a data rate change by link partner devices with the CCIXCapabilities.ESMReCalibrationNeededonESMDataRateUpdate bit Set, or those advertising greater than 24 ms / 32 ms settings in the ESMControl.ESMExtendedEqualizationPhase2/3Timeout fields, may not meet the timeout requirements in Equalization Phases of the *PCI Express Base Specification*. It is strongly recommended that SSW ensures that the link is in a quiescent state (e.g., stop application traffic), before performing data rate changes that require equalization.

7 ESMMandatoryDataRateCapability.ESM16GTSupport must be Set before the value to program into ESMControl.ESMDataRate1 is determined (Section 4.4.2.3 , Step 2e).

8 Writing ESMDataRate1 or ESMDataRate0 when ESMControl.ESMEnable is Set results in undefined behavior.

9 Programming ESMDataRate0 to 20.0 GT/s and ESMDataRate1 to 25.0 GT/s (or vice versa) is not permitted and results in undefined behavior.

IMPLEMENTATION NOTE

ESMControl.QuickEqualizationTimeoutSelect field

Link partner devices that advertise greater than 24 ms / 32 ms settings in the ESMControl.ESMExtendedEqualizationPhase2/3Timeout fields may take longer than *PCI Express Base Specification* compliant devices to complete link training to ESM Data Rate1 after a hot reset or redo equalization event. For these events such devices may not meet requirements specified in *PCI Express Base Specification* Section 6.6.1 that could result in failure to enumerate the device or other issues on platforms running legacy operating systems (OS).

The ESMControl.QuickEqualizationTimeoutSelect field has been defined to avoid the above issue. SSW may optionally Set QuickEqualizationTimeoutSelect to a value greater than 000b during LinkUp as specified in Section 4.4.2.3 Step 2 to effectively reduce the Timeouts used in LTSSM Equalization Phase2/3 states as compared to the default, effectively achieving faster equalization. When the ESMControl.QuickEqualizationTimeoutSelect field is set to a value greater than 000b all of the following apply:

- The ESMControl.ESMExtendedEqualizationPhase2/3Timeout field programming must be ignored and ESM Equalization Phase2/3 Timeouts in the device will use values corresponding to the encoding programmed in the ESMControl.QuickEqualizationTimeoutSelect field.
- Setting the ESMControl.QuickEqualizationTimeoutSelect field to a value greater than 000b, is permitted to be used to restore previously stored implementation specific PHY electrical design parameters, including (but not limited to) transmitter FFE and receiver equalization, that will lead to faster bit/symbol lock and achieve BER better than 10E-4, at the data rate programmed in the ESMControl.ESMDataRate1 field, considering that Equalization timeout is limited to encoded values programmed in ESMControl.QuickEqualizationTimeoutSelect. Designs may use implementation specific mechanisms to save and restore PHY Electrical parameters. The number and type of bits saved and restored are implementation specific. SSW must Clear the ESMControl.QuickEqualizationTimeoutSelect field after Link-Up at ESM Data Rate1 has been achieved.

6.3.6 ESMLaneEqualizationControl Registers

Separate ESMLaneEqualizationControl registers are specified for 20.0 GT/s and 25.0 GT/s data rates. The 20.0 GT/s registers precede the 25.0 GT/s registers in the Transport DVSEC register space. ESMLaneEqualizationControl registers are specified for all implemented lanes, starting with Lane 0, and must be implemented in whole DW increments, with entries for unimplemented lanes being RsvdP. Each ESMLaneEqualizationControl register holds information for up to 4 Lanes (see Figure 6-89). These registers are only implemented if ESMModeSupported bit is 1b.

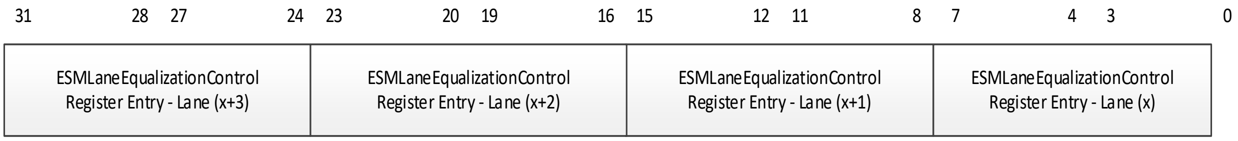

31
28
27
24
23
20
19
16
15
12
11
8
7
4
3
0
ESMLaneEqualizationControl Register Entry - Lane (x+3)
ESMLaneEqualizationControl Register Entry - Lane (x+2)
ESMLaneEqualizationControl Register Entry - Lane (x+1)
ESMLaneEqualizationControl Register Entry - Lane (x)

Figure 6-89: ESMLaneEqualizationControl Registers

Each entry contains the value for the Lane, with the corresponding Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training.

Figure 6-90 details allocation of fields in this register and Table 6-72 provides the respective field definitions.

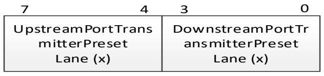

7
4
3
0
UpstreamPortTransmitterPreset Lane (x)
DownstreamPortTransmitterPreset Lane (x)

Figure 6-90: ESMLaneEqualizationControl Register Entry

Table 6-72: ESMLaneEqualizationControl Register Entry

<table>
<tr><th>Bit Location</th><th>Register Description</th><th>Attributes</th></tr>
<tr><td>3:0</td><td><b>DownstreamPortESMTransmitterPrese</b>t
Transmitter Preset used for ESM equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port. See the <i>PCI Express Base Specification</i>, Section 4.2.3.2 for encodings.
For an Upstream Port if Crosslink Supported is 0b, this field is RsvdP. Otherwise, this field is HwInit.
The default value of this field is 1111b.</td><td>HwInit/
RsvdP
(see description)</td></tr>
<tr><td>7:4</td><td><b>UpstreamPortESMTransmitterPreset</b>
Field contains the Transmit Preset value sent or received during ESM Link Equalization. Field usage varies as follows:
<table>
<tr><th></th><th>Operating Port Direction</th><th>Crosslink Supported</th><th>Usage</th></tr>
<tr><td>A</td><td>Downstream Port</td><td>Any</td><td>Field contains the value sent on the associated Lane during Link Equalization.
Field is HwInit.</td></tr>
<tr><td>B</td><td>Upstream Port</td><td>0b</td><td>Field is intended for debug and diagnostics. It contai value captured from the associated Link during Link Equalization.
Field is RO.
When crosslinks are supported, case C (below) applie this captured information is not visible to software. Vendors are encouraged to provide an alternate mechanism to obtain this information.</td></tr>
<tr><td>C-</td><td>Upstreabm Port</td><td>1b</td><td>Field is not used or affected by the current Link Equalization.
Field value will be used if a future crosslink negotiati switches the Operating Port Direction so that case A (above) applies.
Field is HwInit.</td></tr>
</table>
See <i>PCI Express Base Specification</i>, Section 4.2.3.2 for encodings.
The default value of this field is 1111b.</td><td>HwInit/RO
(see description)</td></tr>
</table>

6.3.7 TransportLayerCapabilities Register

The TransactionLayerCapabilities register identifies CCIX transaction specific capabilities. Figure 6-91 details allocation of fields in this register and Table 6-73 provides the respective field definitions.

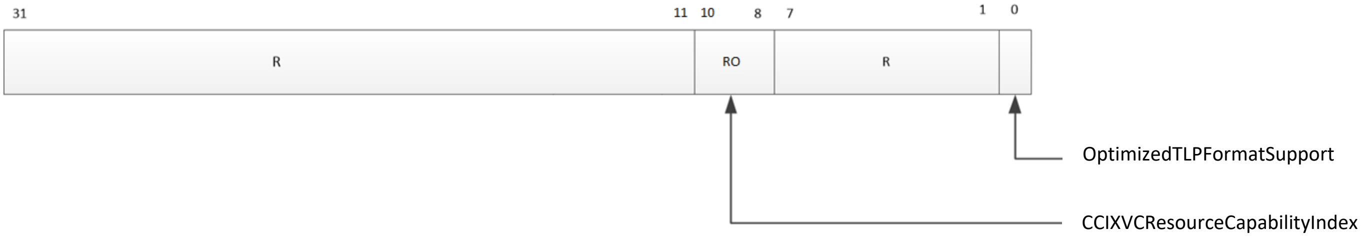


Figure 6-91: TransactionLayerCapabilities Register

Table 6-73: TransactionLayerCapabilities Register

Bit Location	Register Description	Attributes
0	**OptimizedTLPFormatSupport**. When Set - Generation, Reception, and Routing are supported. When Clear - Generation, Reception, and Routing are not supported.	HwInit
1	**ReceiverSupportsType0FormatOnly** When Set, the Receiver does not support the packing of Containers into Flits using the Type 1 format. Only the Type 0 format is supported. When Clear, both Type 0 and Type 1 formats are supported. (See Section 4.4.3) Default value of this bit is 0b.	HwInit
7:2	Reserved and Preserved	RsvdP
10:8	**CCIXVCResourceCapabilityIndex**. With reference to the PCIe "Virtual Channel Capability" this is the index used to identify the VC Resource Capability Register corresponding to the VC to be used for CCIX traffic. The valid range is 1 <= index <= n (where n is the "Extended VC Count" as defined in the latest PCIe specifications). The VC Resource Offset corresponding to "index" is "10h + index*0Ch"	HwInit
31:11	Reserved and Preserved	RsvdP

6.3.8 TransportLayerControl Register

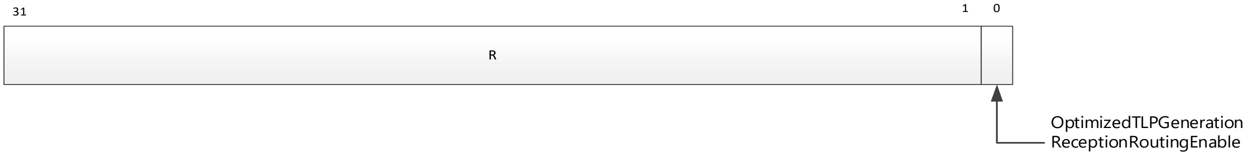


Figure 6-92: TransactionLayerControl Register

Table 6-74: TransactionLayerControl Register

Bit Location	Register Description	Attributes
0	**OptimizedTLPGenerationReceptionRoutingEnable**. See Section 4.2.4 for the behavior controlled by this bit. See Section 4.2.1 for system software requirements regarding the changing of this bit. Functions that do not generate, receive, or forward Optimized TLPs are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
1	**OptionalLengthCheckEnable.** When Set, enables the optional check of the Length field. (See Section 4.2.2.1.1 and Section 4.2.2.1.2). See Section 4.2.1 for system software requirements regarding the changing of this bit. Functions that do not implement this check are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
2	**Type0OnlyPackingEnable** When Set, the Transmitter is not permitted to use the Type 1 format and must pack Containers into Flits using the Type 0 format only. When Clear, the transmitter is permitted to pack with both Type 0 and Type 1 formats (see Section 4.4.3). Functions that support only the Type 0 format are permitted to hardwire this bit to 1b. Default value of this bit is 0b.	RW
31:3	Reserved and Preserved	RsvdP

6.4 DVSEC Discovery and Configuration

This section provides an overview of the CCIX Protocol Layer DVSEC Discovery and Configuration Sequence. The detailed sequence is documented in the CCIX Software Guide.

1 CCIX Device Power Up.

2 PCIe Discovery and Enumeration.

3 PCIe topology or PCIe Device Tree is handed to the CCIX Configuration Software.

4 CCIX Configuration Software then performs CCIX Device Discovery - revisiting PCIe Device Tree and identifying CCIX Components via CCIX ID identifiers within the PCIe Device's DVSEC Configuration space.

The Discovery Sequence follows the order in which the CCIX DVSEC Register Structures are arranged as shown in Figure 6-1.

a Discovery of Transport Layer DVSEC attributes, including whether Extended Speed Mode (ESM) and Optimized TLP Headers are supported by each CCIX Device.

b Discovery of Protocol Layer DVSEC attributes:

i. If a CCIX Component indicates not-Ready status, the discovery is postponed until past the Readiness Time Reported by the CCIX Device. However, CCIX Configuration Software is permitted to consider Capabilities & Status declared within the not-Ready state as transient Capabilities & Status and compare them against the final Capabilities & Status declared when the CCIX Device is ready.

ii. Software must not set CCIX Component Enable until it gets a CCIX Component Ready Status indication.

5 Create and Configure G-HSAM and G-HSAM allocation across CCIX Agents.

a Recommend coalescing G-SAM to hierarchy of HA to allow for Address Range based routing. For example, one RA accessing multiple HAs via one RSAM entry that spans addresses allocated to the multiple HAs.

b Recommend coalescing G-HSAM to hierarchy of SA to allow for Address Range based routing. For example, one HA accessing multiple SAs via one HSAM entry that spans addresses allocated to the multiple SAs.

6 Create and Configure CCIX AgentID Map.

a Recommend coalescing CCIX AgentID to hierarchy of SA/HA to allow CCIX AgentID range based routing if needed.

7 Determine CCIX PortID Map.

8 Configure HBAT/HA IDM and SBAT/SA IDM attributes.

9 Configure HSAM and RA IDM attributes.

10 If a CCIX Switch is present, setup its SAM/IDM and other Common, Port and Link data structures such that the data structures reflect the topology of CCIX Devices that are around that CCIX switch (see Section 6.5).

11 Enable SA.

12 Enable HA.

13 Enable RA.

6.5 CCIX Switch Referenced Data Structures

A CCIX Switch is a multiport CCIX Device with CCIX Port-to-Port forwarding capabilities on all CCIX Ports, and without necessarily having CCIX Agent Capabilities. As a result, CCIX Switch DVSEC data structures include all the Transport Layer, and relevant Protocol Layer DVSEC data structures. The Protocol Layer DVSEC data structures describing the forwarding properties and configuration of the CCIX Switch are the Common, CCIX Port and CCIX Link data structures described in Section 6.2.

Examples to Illustrate Protocol Layer DVSEC Usage

This section contains examples that illustrate the usage of CCIX Protocol Layer Component data structures to achieve CCIX Device-to-Device and CCIX Agent-to-Agent Packet flow.

CCIX Protocol Layer Component data structures are part of PCIe DVSEC space. CCIX Protocol Layer DVSEC data structures, or fields or field encodings within those data structures, described in this section are in italics.

Each example focuses on particular aspects of data structure usage and as such, may not include all the data structures necessary, or an exhaustive description of the usage of all fields within a highlighted data structure. The description of the usage of a data structure or data structure field in a latter part of the section is inferred, and therefore not repeated, if the data structure or data structure field usage is in the same manner as in a former part of this section.

6.5.1 Simple CCIX Topology and Relevant Data Structures

Figure 6-93 describes a simple CCIX topology connecting three CCIX Devices, each with a unique CCIX Agent type. Figure 6-93 also illustrates how the SAM is mapped to a CCIX Agent's view, some of the data structures for a CCIX Agent's capabilities and controls, and the setup of those structures to enable a CCIX Agent's accessibility and routing attributes as part of the CCIX system. A PCIe Switch is shown as the transport switch between the CCIX Devices but it's transparent with respect to the *CCIX Protocol Layer DVSEC structures* because the *CCIX Link Destination TransportIDs* (*LinkTransportIDMapEntry.DestTransportID*) are programmed with the PCIe BDFs of the three CCIX Devices.

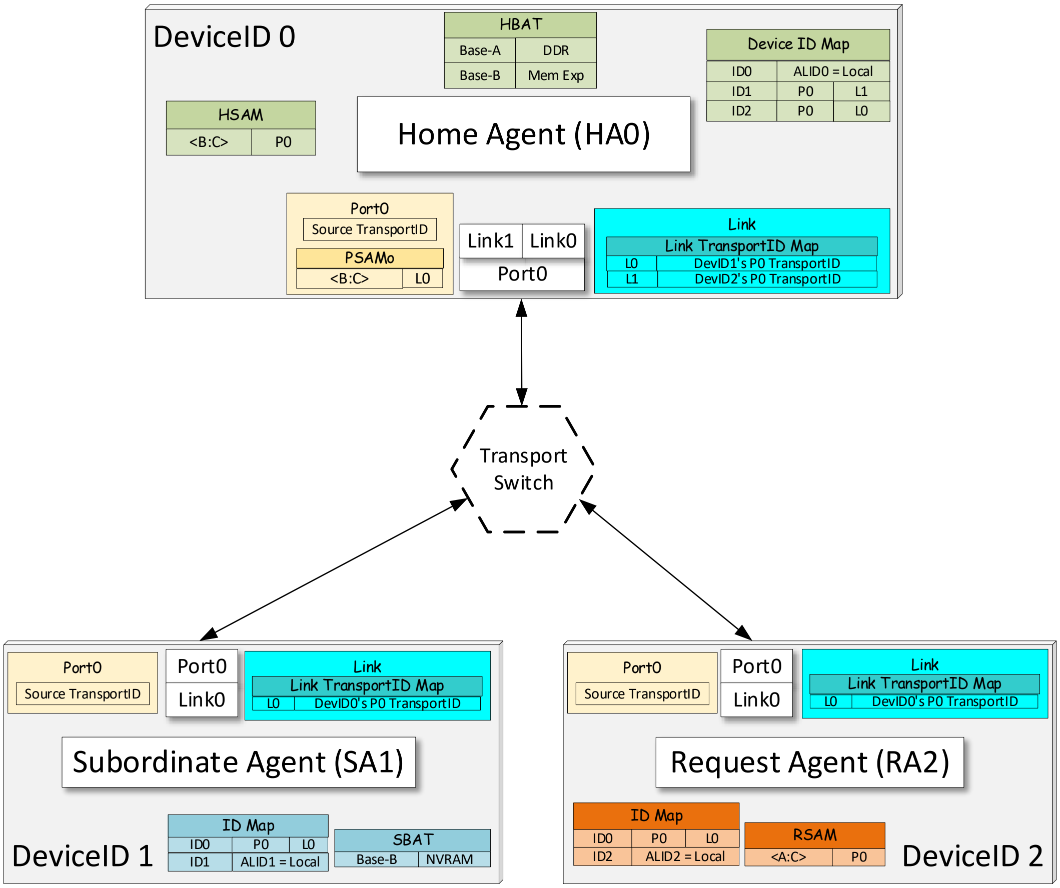


Figure 6-93: Example simple CCIX Topology with relevant data structures

- **Request Agent (RA):**
  - The Request Agent is on a CCIX Device programmed with a *ComnCntl1.DevIDCntl* value of 2 in its *Common Control structure*. The CCIX Device is thus enumerated as CCIX DeviceID2.
  - The Request Agent is programmed with a *RAID* value of 2 in its *RA Control structure (RACntl.RAID field)*. The RA is thus enumerated as RAID2. There is no requirement that the CCIX Device ID and CCIX AgentID match, and a common value of 2 is for illustration purposes only.
  - Address Routed CCIX Packets from the RA:

- – The routing properties of the G-RSAM, for address routed CCIX Packets from Request Agents, are described in CCIX DeviceID2's *Common Control structure*, the *RSAM Table*.

- – The *RSAM* Table contains the routing properties of SAM windows. The SAM windows are described by the *RSAM Entry Start/End Registers* (SAMEntryAddr0.StartAddr/SAMEntryAddr1.EndAddr). The *RSAM Entry* also contain the *CCIX PortID* (*SAMEntryAttr.PortID*) destination for those SAM Windows.

  - ▪ The Request Agent accesses the routing properties of the G-RSAM <A:C>, with the RSAM Entry *SAMEntryAddr0.StartAddr* field programmed with 4GB aligned Start Address-A, and the RSAM Entry *SAMEntryAddr1.EndAddr* field programmed with 4GB aligned End Address-C.

  - ▪ G-SAM <A:C> programmed in the RSAM Entry has been mapped to CCIX Port0, indicated by the *SAMEntryAttr.PortID* field being programmed with a value of 0.

- – The number of RSAM Table entries is a per-CCIX-Device capability advertised in the RSAM Max Table Size (RSAMPtr.RSAMMaxTblSize) field of the Primary CCIX Port Extended Common Capabilities & Status structure. The number of RSAM Table entries indicates the number of unique SAM Windows the RA is capable of accessing.

- o ID Map data structures:

  - – The AgentID Map or IDM structure maps the destination CCIX PortID (IDMEntry.AgentIDnPortID) and CCIX LinkID (IDMEntry.AgentIDnLinkID) for ID routed packets from the RA:

    - ▪ The IDM entry for AgentID0 is mapped to CCIX Port0, CCIX Link0

- o CCIX Packets from RA2 are sent with the value in the Source TransportID field (PortSrcIDCntl.SrcTransportID) of CCIX Port0's CCIX Port Control structure, and the value in the Destination TransportID field (LinkTransportIDMapEntry.DestTransportID) in the CCIX Link0 Entry of the CCIX Link TransportID Map which is part of the CCIX Link Control structure.

- **Home Agent (HA) on CCIX Device 0:**

  - o The Home Agent's capabilities and controls are contained within the *Home Agent data structure*:

    - – The *Home Agent Capabilities & Status structure* on CCIX Device 0 has declared a *HACapStat.NumHAID field* value of 1, indicating the Home Agent only requires one AgentID value programmed.

    - – The Home Agent's *AgentID* has been programmed with ID0 in the *AgentIDforHA0* field of *HAIDTblEntry0*, which is part of the *Home Agent Control structure*.

    - – The *Home Agent Capabilities & Status structure* declares the number of memory pools (*HACapStat.HAMemPoolCap*), size and type of those memory pools of the HA (*MemPoolEntryCapStat0 Register*).

    - – The HA has also indicated memory expansion capability via the HACapStat.HAMemExpnCap field and memory expansion has been enabled for this HA via the HACntl.HAMemExpnEnable field of the Home Agent Control structure.

  - o Address data structures:

- There are two types of Tables describing the HA's routing properties to G-SAM Windows, the *HSAM* or *Home Agent System Address Map Table*, and the *HBAT*, or *Home Agent Base Address Table*.
- The properties of the G-RSAM windows mapped to this HA are described in the *HBAT data structure*:
    - The *HA Memory Pool Capability (HACapStat.HAMemPoolCap)* has declared capabilities of two *Memory Pool Entries*. As a result, the HA has also implicitly declared capability of two *HBAT Entries*, with each *HBAT Entry* conforming to the structure declared in the *Memory Pool General Memory Type (MemPoolEntryCapStat0.MemPoolGenMemTypeCap) field* of its corresponding *Memory Pool Entry*.
    - The HA has also declared Memory Expansion Capability (HACapStat.HAMemExpnCap). Since the HA only has two Memory Pool Entries, if the first Memory Pool Entry does not declare Memory Expansion Capability, it means the second Memory Pool Entry must declare Memory Expansion Capability in its Memory Pool General Memory Type (MemPoolEntryCapStat0.MemPoolGenMemTypeCap) field.
    - HA0's *Memory Pool Entry* 0 has declared local, direct-attached DDR Memory in the *MemPoolEntryCapStat0.MemPoolSpcificMemTypeCap field*, and the corresponding *BAT Entry 0* has been mapped to 4GB aligned Base Address-A of the G-RSAM. HA0's *Memory Pool Entry* 0, in the *MemPoolEntryCapStat0.MemPoolAddrCap field,* has indicated addressing capability to 4GB aligned Base Addresses.
    - HA0's Memory Expansion *BAT Entry 1* has been mapped to 4GB aligned Base Address-B of the G-RSAM. Since this HA0's *Memory Pool Entry* has declared *Memory Expansion Capability*, the *MemPoolEntryCapStat0.MemPoolAddrCap field* must indicate addressing capability to 4GB aligned Base Addresses.
- The HSAM Table describes the routing properties of the G-HSAM Windows mapped to Subordinate Agents, for address routed CCIX Packets from this Home Agent. This is optional capability; a Home Agent is not required to have an HSAM structure if the HA has not declared Memory Expansion Capability (HACapStat.HAMemExpnCap).
    - The Home Agent has been provided memory expansion to G-HSAM Window <B:C> mapped to a destination Subordinate Agent, with the *HSAM Entry0 SAMEntryAddr0.StartAddr field* programmed with 4GB aligned Start Address-B, and the *HSAM Entry SAMEntryAddr1.EndAddr field* programmed with 4GB aligned End Address-C.
- G-HSAM Window <B:C> programmed in *HSAM Entry0* has been routed to CCIX Port0 with the *SAMEntryAttr.PortID field* set to 0h.

- ID Map data structure:
    - The AgentID Map or IDM structure maps the destination CCIX PortID and CCIX LinkID for ID routed packets from the Home Agent.

        - ▪ The IDM entry for AgentID1 has SA1 destined packets routed to CCIX Port0, CCIX Link1.
        - ▪ The IDM entry for AgentID2 has RA2 destined packets routed to CCIX Port0 CCIX Link0.

- **Subordinate Agent:**
    - The Subordinate Agent's capabilities and controls are contained within the *Subordinate Agent data structure*.
    - General SA capabilities and control attributes:
        - The Subordinate Agent's *AgentID* has been programmed with ID1 in the *SACntl.SAID field* of the *Subordinate Agent Control structure*.
        - The Subordinate Agent Capabilities & Status structure declares the number of memory pools, size and type of those memory pools, of the SA.
    - Address data structures:
        - The properties of the G-HSAM window mapped to this SA are described in the SBAT data structure:
            - ▪ SA1's NVRAM Memory Pool has been mapped in BAT Entry 0 to 4GB aligned Base Address-B of the G-HSAM.
        - The *SBAT* entries are programmed in the *SA Control structure*.
    - ID Map data structure:
        - The AgentID Map or IDM structure maps the destination CCIX PortID and CCIX LinkID for ID routed packets from the Subordinate Agent.
            - ▪ The IDM entry for AgentID0 has HA0 destined packets routed to CCIX Port0 CCIX Link0.

- **CCIX Link**:
    - A CCIX Device can have one or more CCIX Links for a given CCIX Port on that CCIX Device. The CCIX Link's capabilities and controls are contained within the *CCIX Link data structures*.
    - General CCIX Link capabilities and control attributes:
        - The capabilities and control attributes determine the CCIX LinkID, max credit capability (LinkSendCap and LinkRcvCap Registers) and max and min credit control (LinkMaxCreditCntl and LinkMinCreditCntl Registers) of the CCIX Link.
            - ▪ Request Agent CCIX Link:
                - The single *Link Control structures* have been programmed for CCIX Link ID0 with the *Link Control Attributes* for the single CCIX Port-pair transport between RA2 and HA0.

- Link0 Attribute Control structure contains the values for the Min and Max Send Credit as well as the Max CCIX Packet Send Control for the single CCIX Port-pair transport between RA2 and HA0.

- Link0 Map Entry Control structure contains the value of CCIX Device ID0's Port0 Destination TransportID (LinkTransportIDMapEntry.DestTransportID), which in the case of the PCIe transport, contains the value of the PCIe BDF of CCIX Device 0, Port0.

- Home Agent CCIX Link:
  - The two *CCIX Link Control structures* have been programmed with the *Link Control Attributes* for CCIX Link ID0 and CCIX Link ID1, for the two CCIX Port-pair transports, one CCIX Port-pair between HA0 and RA2 and the other CCIX Port-pair between HA0 and SA1.

- Subordinate Agent CCIX Link:
  - The single *CCIX Link Control structures* have been programmed for CCIX Link ID0 with the *Link Control Attributes* for the single CCIX Port-pair transport between SA1 and HA0.

- **CCIX Port:**
  - A CCIX Device can have one or more CCIX Ports on that device. A CCIX Port's capabilities and controls are contained within the *CCIX Port data structure*.
  - General CCIX Port capabilities and control attributes:
    - The capabilities and control fields in CCIX Port structures determine the CCIX PortID (*PortCapStat1.PortID)*, Optimized Headers (*PortCapStat1.PktHdrTypeCap and PortCntl.PktHdrTypeEnable*), number of CCIX Links (*PortCapStat1.NumLinksCap and PortCntl.NumLinksEnable*), and PSAM entries (*PortCapStat1.NumPSAMEntryCap and PortCntl.NumPSAMEntryEnable*), amongst others:
      - Request, Home and Subordinate Agent CCIX Port:
        - CCIX PortID0's Control structure, i.e. the Primary CCIX Port Control structure, has the Source TransportID (*PortSrcIDCntl.SrcTransportID*) programmed with the corresponding PCIe physical transport's BDF.
  - CCIX Port Outbound characteristics:
    - *PSAM* is referenced to determine the G-SAM window to CCIX Link Map for address routed CCIX Packets.
    - Because CCIX Device 1 and CCIX Device2 each have a single CCIX Port with a single CCIX Link, the two CCIX Devices are not required to have per-*CCIX Port PSAM structures*. All CCIX Packets from RA2 are issued on CCIX Device2's CCIX Port0 CCIX Link0. Similarly, all CCIX Packets from SA1 are issued on CCIX Device1's CCIX Port0 CCIX Link0.

  - – Because CCIX Device 0 has a single CCIX Port0 with two CCIX Links, and only HA0 to SA1 CCIX Packets are address routed, a single *Port PSAM Entry* is referenced to determine the CCIX Link:
    - ▪ *Port PSAM Entry0*: SAM Window <B:C> Mapped to CCIX Link1 with the *PSAMEntryAttr.LinkNum field* programmed with value 01h, the *PSAMEntryAddr0.StartAddr field* programmed with 4GB aligned Start Address-B, and the *PSAMEntryAddr1.EndAddr field* programmed with 4GB aligned End Address-C. The *PSAMEntryAttr.PSAMEntryAddrType field* is set to 1b to indicate this PSAM Entry contains HA-to-SA address routing information.
- o CCIX Port Inbound characteristics:
  - – Either the *RSAM* or *HSAM structure* is referenced to determine the G-SAM window to Local CCIX Agent or CCIX Port Map for the case of CCIX Devices with *Multi-Port Device Capability (*ComnCapStat1.MultiPortDevCap*)*, and *CCIX Port-to-Port Forwarding Capability (*PortCapStat1.PortToPortFwdingCap*)*. For the simple topology example in Figure 6-93, all inbound address routed CCIX Packets from a CCIX Port, are routed to local destinations with the addresses falling within a SAM Window programmed in the *BAT Entry* of the CCIX Agent on that CCIX Device, hence the *RSAM* or *HSAM structure* is not referenced:
    - ▪ Device0: SAM Window <A:C> Mapped to HBAT Entries
    - ▪ Device1: SAM Window <B:C> Mapped to SBAT Entry.
- o The *ID Map structure* is referenced to determine the AgentID to CCIX Port Map for ID routed CCIX Packets from a CCIX Agent:
  - – Due to the simple topology in Figure 6-93, there is only one destination which is the CCIX Agent on that CCIX Device and as such, all *IDM Tables* in Figure 6-93 have an *IDM Entry* pointing to an on-chip Local destination, with the *IDMEntry.AgentIDnDestType* field set to 0b for that IDM Entry:
    - ▪ Device0: IDM entry for AgentID0 has *IDMEntry.AgentIDnDestType* set to 0b.
    - ▪ Device1: IDM entry for AgentID1 has IDMEntry.AgentIDnDestType set to 0b.
    - ▪ Device2: IDM entry for AgentID2 has *IDMEntry.AgentIDnDestType* set to 0b.

- CCIX Common Structure:
  - o Although not illustrated in Figure 6-93, a CCIX Device can have one or more CCIX Agents as well as one or more CCIX Ports. Therefore, properties that span all CCIX HA/SA/RA/Ports resident within that CCIX Device are located in the *CCIX Protocol Layer Common structure*. Attributes such as *CCIX Device Enable, CCIX Device Error* and *CCIX Port Aggregation Support* are described within that structure.
  - o The IDM and SAM Tables referenced throughout Sections 6.2.2.1.2.3 and 6.2.2.3 are located in the Primary CCIX Port Common Control Structure.

6.5.1.1 Simple CCIX Topology and Associated SAM and BAT Data Structures

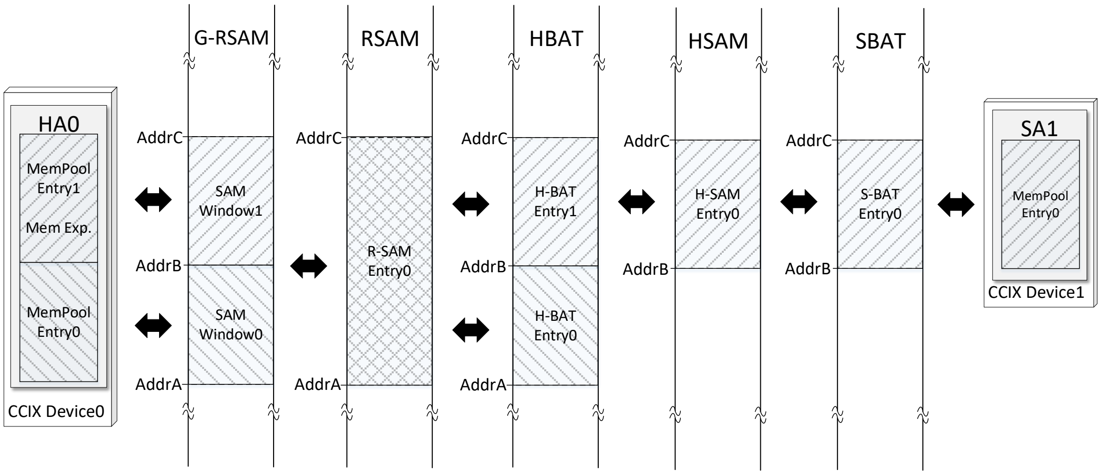
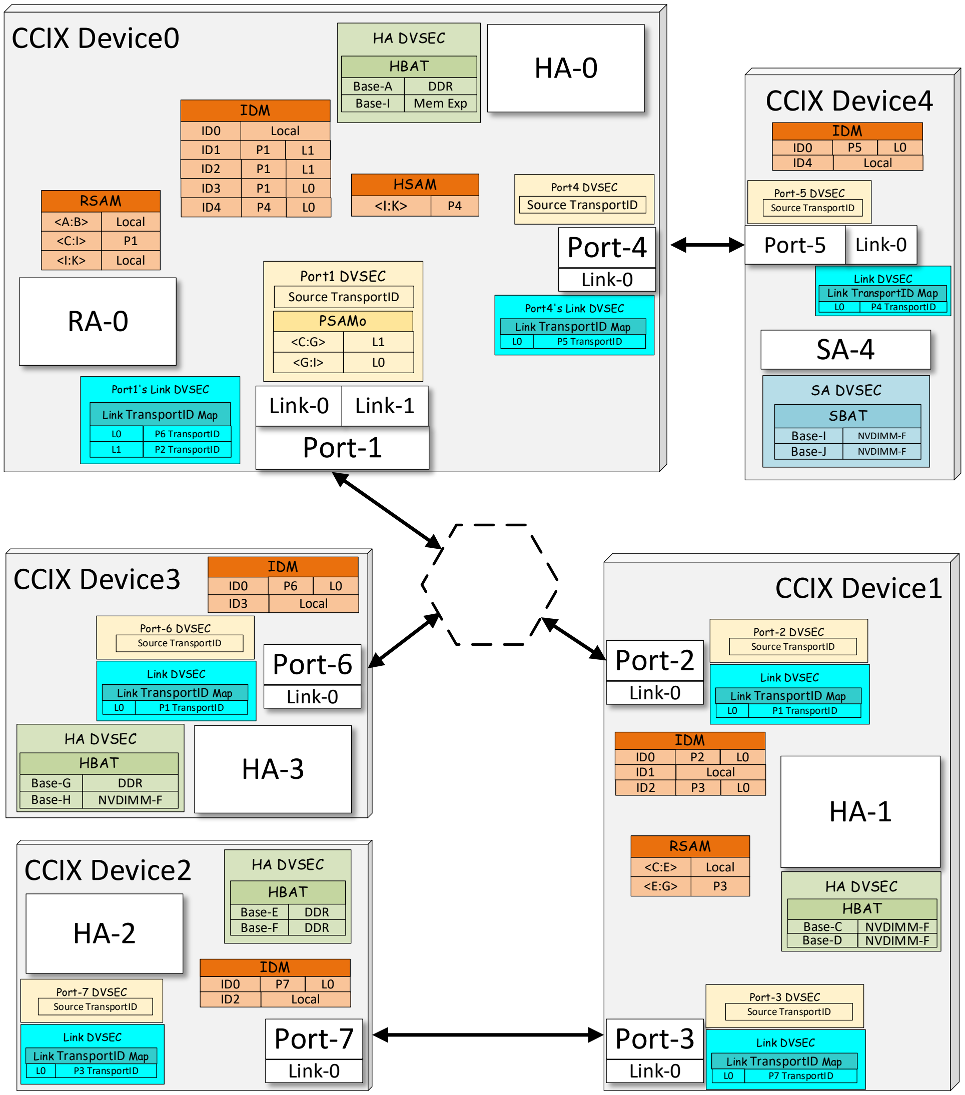


Figure 6-94: SAM Windows and associated data structures for simple CCIX Topology example

Figure 6-94 is an illustration of the G-RSAM for the simple CCIX topology example and their association with CCIX Device and CCIX Agent data structures. CCIX Configuration Software has enabled the HA to access the Subordinate Agent’s Memory Pool mapped in the SBAT for the SA, and CCIX Device0’s HSAM for the HA. The sole Request Agent has access to the aggregate of the Subordinate Agent’s and Home Agent’s Memory Pools that are mapped in CCIX Device2’s RSAM, with the aggregate region, i.e. the G-RSAM, also being mapped in the Home Agent’s HBAT.

6.5.2 Complex CCIX Topology and Relevant Data Structures

Figure 6-95 describes a complex CCIX topology, including CCIX Devices with multiple CCIX Agents.



Note
CCIX Ports are uniquely enumerated across CCIX Devices only for illustration purposes
i.e. Port0 can simultaneously be on CCIX DeviceID-X and DeviceID-Y
Also, CCIX Ports must be sequentially enumerated on the same CCIX Device, starting with CCIX Port0.

Figure 6-95: Example complex CCIX Topology with relevant data structures

Request Agent 0:

- RA0 routing information to the G-RSAM is described in CCIX Device0's *RSAM Table,* for address routed CCIX Packets:
  - Accesses to RSAM Window <A:B> routed to HA0 since the *SAMEntryAttr.DestType* field of the *RSAM Entry* indicates *Local Destination*.
  - Accesses to RSAM Window <C:I> routed to CCIX Port1.

  - Accesses to RSAM Window <I:K> also routed to HA0, with the *SAMEntryAttr.DestType* field of the *RSAM Entry* indicating *Local Destination*. This also illustrates a case where other considerations, such as *HA Memory Expansion capability*, could result in accesses to non-contiguous SAM Windows being routed to the same HA.
- The *IDM Table* is referenced for the CCIX Port destination for ID routed CCIX Packets from the RA:
  - ID0's *IDM Entry* has the *Local field* set, indicating a local CCIX Device destination.
  - ID1, ID2, ID3 *IDM Entries* indicate a route-destination of CCIX Port1, indicating remote CCIX Device destinations.

CCIX Port1:

- The *PSAM Table* highlights how CCIX relies on hierarchical decode for routing information. Even though one *RSAM Entry* in the *RSAM Table* has SAM Window <C:I> routed to CCIX Port1, CCIX Port1's *PSAM Table* provides further routing information at a CCIX Port-level. These address routed CCIX Requests are routed to different CCIX Links representing different off-chip CCIX Port destinations:
  - Accesses to RSAM Window <C:G> routed to CCIX Link1.
  - Accesses to RSAM Window <G:I> routed to CCIX Link0.

Home Agent 0:

- HA0 has Direct Attached Memory and its *HBAT Entry0* maps those resources:
  - Base Address-A is mapped to Memory Pool0 where Memory Pool Entry0 has declared DDR Memory Type in its Specific Memory Type Capability (MemPoolSpcificMemTypeCap) field.
- HA0 has also declared *Memory Expansion Capability* in *Memory Pool Entry1*. This capability is enabled in the corresponding *HBAT Entry1* for address routed CCIX Packets from RAs to HA0:
  - 4GB aligned Base Address-I is programmed in *HBAT Entry1.*
- Address routed CCIX Packets initiated by HA0 to SAs, reference the *HSAM Table* in the *Common Control structure* of CCIX Device0, in order to resolve the destination CCIX Port:
  - Accesses to HSAM Window <I:K> routed to Port4.

Subordinate Agent 4:

- SA4 has its *SBAT Entries* configured to match the Memory Expansion SAM Window configured in HA0:
  - 4GB aligned Base Address-I is mapped in SBAT Entry0 to Memory Pool0 where Memory Pool Entry0 has declared NVDIMM-F Memory Type in its Specific Memory Type Capability (MemPoolSpcificMemTypeCap) field. Since this is SA4's Memory Pool Entry 0, the MemPoolEntryCapStat0.MemPoolAddrCap field must indicate addressing capability to 4GB aligned Base Addresses.
  - 4GB aligned Base Address-J is mapped in SBAT Entry1 to Memory Pool1 where Memory Pool Entry1 has declared NVDIMM-F Memory Type in its Specific Memory Type Capability (MemPoolSpcificMemTypeCap) field. SA4's Memory Pool Entry 1's MEMPOOLADDRCAP field has indicated addressing capability to 4GB aligned Base Addresses.

CCIX Device1:

- CCIX Device 1 declared Multi-Port CCIX Device Capability (MPDC) in the Common Capabilities & Status structure, discovered on CCIX Port2 and Port3.
- CCIX Port2 and Port3 were both determined to be part of CCIX Device1 when the DIDC field value programmed in the CCIX Device1's Primary Port Common Control structure was reflected in the DIDS field of the CCIX Port2's and Port3's Common Capabilities & Status structure.
- CCIX Port2 and Port3 also declared CCIX Port-to-Port Forwarding Capability (PortCapStat1.PortToPortFwdingCap) in the Port Capabilities & Status structure. Since CCIX Device 1 has only two CCIX Ports, and PortCapStat1.PortToPortFwdingCap is a bi-directional capability, both CCIX Port2 and Port3 must declare CCIX Port-to-Port Forwarding Capability.
- Since CCIX Device 1 has only two CCIX Ports, and PortCapStat1.PortToPortFwdingCap is a bi-directional capability, the Port Forwarding Vector (PortCapStat3.PortFwdingVctr) field, in each Port Capabilities & Status structure, must reference the other port, i.e. PortCapStat3.PortFwdingVctr in CCIX Port2's Port Capabilities & Status structure must have the bit position for CCIX Port3 set, and PortCapStat3.PortFwdingVctr in CCIX Port3's Port Capabilities & Status structure must have the bit position for CCIX Port2 set.
    - As indicated in Figure 6-95, CCIX Device 1 has been enumerated with CCIX Port2 and Port3 designations for illustration purposes only. A CCIX Device with two CCIX Ports, must have those CCIX Ports linearly enumerated, and must have one *Primary Port Capabilities & Status structure* with PortID0 declared in the *PID field*. Since CCIX Device 1 has only two CCIX Ports, the CCIX Ports would be required to be enumerated as CCIX Port0 and Port1, and the *PortCapStat3.PortFwdingVctr field* bit position descriptions above must be adjusted accordingly.
- RA0 address routed CCIX Packets, that ingress on CCIX Port 2, have their routing information in CCIX Device1's *RSAM Table.* The *RSAM Entries* have routing information for two RSAM Windows, where packets addressed to one RSAM Window are routed to a Home Agent local to the CCIX Device, and packets addressed to the other RSAM Window are routed to an egress Port destined to a Home Agent on a remote CCIX device:
    - Accesses to RSAM Window <C:E> routed to HA1 since the *SAMEntryAttr.DestType* field of the *RSAM Entry* indicates *Local Destination*.
    - Accesses to RSAM Window <E:G> routed to CCIX Port3.

Configuration and usage of data structures in Figure 6-95, that are not specifically covered in the descriptions above, can be inferred from other equivalent data structures described thus far.

6.5.2.1 Complex Topology and Associated Number of SAM Entries

In Figure 6-95, the number of RSAM Entries that are enabled in CCIX Device 0 and Device 1 differs from the (P + 1 + Local) minimum number of SAM Entry Capability requirement, as stated in Table 6-10.

As noted in Table 6-10, particular CCIX tree topologies may achieve their necessary SAM Window routing attributes by utilizing less than (P + 1 + Local) SAM Entries. CCIX Device 0 and Device 1 in Figure 6-95 illustrates such a utilization:

- RSAM Entries utilized:
  - CCIX Device 0:
    - Because only Port1 has RA-to-HA traffic and Port4 does not, Device 0 requires only 2 RSAM Entries for Port routing
    - For Local RSAM Windows, it is possible to optimize the G-RSAM Window allocation such that the two Local RSAM Windows can be optimized to utilize only one RSAM entry for routing to a Local destination. However, the RSAM on CCIX Device 0 illustrates unoptimized G-RSAM allocation such that two RSAM Entries are utilized for routing to a Local destination.
    - Thus, it's possible for G-RSAM Window allocation such that the RSAM Table on CCIX Device 0 utilizes a total of 2 Entries, even though the minimum requirement of (P + 1 + Local) SAM Entries for CCIX Device 0 is 3.
  - CCIX Device 1:
    - RA-to-HA traffic arriving on Port2 gets forward to only one branch of the tree, i.e. Port3. So only one RSAM entry is utilized for routing to CCIX Port 3.
    - One entry is utilized for routing to a Local destination.
    - Thus, it's sufficient for the RSAM Table on CCIX Device 1 to have utilized 2 Entries, even though the minimum requirement of (P + 1 + Local) SAM Entries for CCIX Device 1 is 3.
- HSAM Entries utilized:
  - CCIX Device 0:
    - Because only Port4 has HA-to-SA traffic and Port0 does not, it's sufficient for the HSAM Table on Device 0 to utilize only 1 entry.
    - Because there is an HA on CCIX Device 0, an SA cannot be on the same CCIX Device. Therefore, there is no requirement on CCIX Device 0 for an HSAM Entry to have a Local routing destination.
    - Thus, it's sufficient for HSAM Table on Device 0 to have utilized 1 Entry, even though the minimum requirement of (P + 1 + Local) SAM Entries for CCIX Device 0 is 3.

6.5.3 CCIX Topology with Port Aggregation and SA Device Aggregation

Figure 6-89 describes a CCIX topology, with the first HA Device with Port Aggregated connections to a second HA Device, with both HAs having SA Device aggregated connections to a third and fourth SA Device.

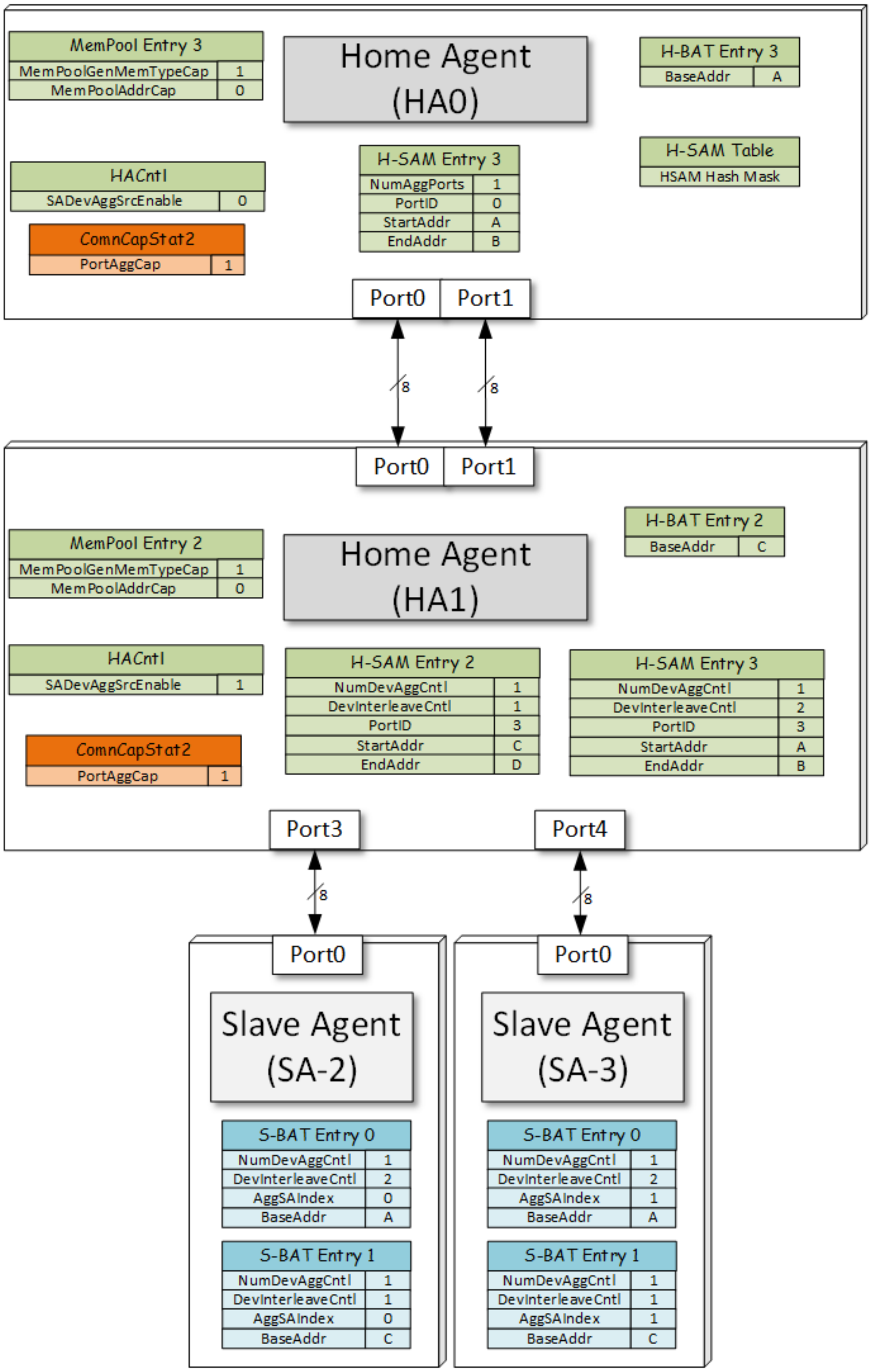


Figure 6-96: Example CCIX Topology with Port and SA Device Aggregation

Possible considerations that can result in the particular topology illustrated in Figure 6-89 include:

- The CCIX Device with HA0 is a CCIX Device with Port Aggregation Capability and HA0 with CCIX Memory Expansion Capability, but not CCIX SA Device Aggregation Capability.
- The memory expansion capabilities of both HA0 and HA1 can be satisfied by aggregating the memory pool sizes available in SA2 and SA3.
- The CCIX Device with HA1 has four CCIX Ports and HA0 with Memory Expansion Capability. The CCIX Device has Port Aggregation Capability and SA Device Aggregation Capability.
- As a result, the CCIX Device with HA1 is configured as an intermediate CCIX Device with both CCIX Port Aggregation, and SA Device Aggregation enabled.
- SA2 and SA3 are together as an aggregated SA Device pair due to their matching capabilities (memory pool sizes, connectivity/bandwidth, etc.).
- The total bandwidth provided by the CCIX ports of the aggregated SA Device pair matches the total bandwidth of CCIX ports on each of the CCIX devices with HA0 and HA1.

CCIX Device with Home Agent 0:

- HA0 has declared *Memory Expansion Capability* via a value of 1b in the *MemPoolGenMemTypeCap* field in its *Memory Pool Entry3*. This capability is enabled in the corresponding *HBAT Entry3* for address routed CCIX Packets from HA0 to the corresponding SAs:
  - ○ 4GB aligned Base Address-A is programmed in *HBAT Entry3.*
- The CCIX Device with HA0 has indicated *Port Aggregation Capability* via a value of 1b in the *PortAggCap* field of the *ComnCapStat2* Register. This capability is enabled via a value of 1b in the *NumAggPorts* field of *HSAM Entry3*:
- ○ Base Address-A from *HBAT Entry3* is programmed as the *StartAddr* in *HSAM Entry3* with the EndAddr-B corresponding to the *MemPoolSizeCap* indicated in *Memory Pool Entry3.*
- ○ Port0 of the CCIX Device is programmed as the BasePortID in the *PortID* field in *HSAM Entry3.*
- ○ Determination of a Port0 or Port1 desitnation for address routed HA-to-SA CCIX packets to <A:B> follow the Aggregated Port Selection Function (APSF) which references the *HSAM Hash Mask* programmed in the *HSAM Table*.
- For completeness, Figure 6-89 also illustrates that the CCIX Device with HA0 does not have:
- ○ SA Device Aggregation enabled, indicated via a value of 0b in the *SADevAggSrcEnable* field of the *HACntl* Register.

CCIX Device with Home Agent 1:

- HA1 has declared *Memory Expansion Capability* via a value of 1b in the *MemPoolGenMemTypeCap* field in its *Memory Pool Entry2*. This capability is enabled in the corresponding *HBAT Entry2* for address routed CCIX Packets from HA1 to the corresponding SAs:
  - ○ 4GB aligned Base Address-C is programmed in *HBAT Entry2.*
- The CCIX Device with HA1 has indicated *SA Device Aggregation Source Capability* via a value of 1b in the *SADevAggSrcCap* field of the *HACapStat* Register. This *SA Device Aggregation Source Capability* is enabled via a value of 1b in the *SADevAggSrcEnable* field of of the *HACntl* Register and further in *HSAM Entry2*:

- The *NumAggDevCntl* field is programmed with a value of 1h to indicate the address range indicated in *HSAM Entry2* has an *SA Device Aggregated Destination* across two SA Devices.
- The *DeviceInterleaveCntl* field is programmed with a value of 1h to indicate a 128B address interleave size between the two SA Devices for the address range indicated in *HSAM Entry2*.
- Base Address-C from *HBAT Entry2* is programmed as the *StartAddr* in *HSAM Entry2* with the EndAddr-D corresponding to the *MemPoolSizeCap* indicated in *Memory Pool Entry2*.
- Port3 of the CCIX Device is programmed as the BasePortID in the *PortID* field in *HSAM Entry2*.
- Determination of a Port3 or Port4 destination for address routed HA-to-SA CCIX packets to <C:D> follow the Aggregated SA Selection Function (ASSF).

- The CCIX Device with HA1 is configured such that address routed HA-to-SA CCIX packets to <A:B> that ingressed on their first hop to this CCIX Device on Port0 and Port1 now egress to their next hop SA Device Aggregated Destinations via fields programmed in *HSAM Entry3*:
  - The *NumAggDevCntl* field is programmed with a value of 1h to indicate the address range indicated in *HSAM Entry3* has an SA Device Aggregated Destination across two SA Devices.
  - The *DeviceInterleaveCntl* field is programmed with a value of 2h to indicate a 256B address interleave size between the two SA Devices.
  - Base Address-A is programmed as the *StartAddr* in *HSAM Entry3* with EndAddr-B.
  - Port3 of the CCIX Device is programmed as the BasePortID in the *PortID* field in *HSAM Entry3*.
  - Determination of a Port3 or Port4 destination for address routed HA-to-SA CCIX packets to <A:B> follow the Aggregated SA Selection Function (ASSF).
- For a sequential address sequence of HA-to-SA CCIX Memory Request packets that fall under address range <A:B> or address range <C:D>, the Aggregated SA Selection Function (ASSF) will select for egress on Port3 and Port4 addresses where the LSB bits of the 64B aligned addresses follow a pattern:
  (in this example, while the LSB bits are used to illustrate address distribution between the CCIX Ports, the MSB bits of the address are not shown, but it's the MSB bits that determine whether the address falls under address range <A:B> or address range <C:D>)
  - Port3:
  - Address Range <A:B>: 0B, 64B, 128B, 192B, > > > 512B, 576B, 640B, 704B,…
  - Address Range <C:D>: 0B, 64B, > > > 256B, 320B, > > > 512B, 578B,…
  - Port4:
  - Address Range <A:B>: 256B, 320B, 384B, 448B, > > > 768B, 832B, 896B, 960B,…
  - Address Range < C:D>: 128B, 192B, > > > 384B, 448B, > > > 640B, 704B,…
    - Addresses in Blue: CCIX Memory Requests eligible for the ReqChain OpCode depending on the address of the previous CCIX Memory Request and the capabilities of the Link partners.
    - Addresses in Green (with >>> arrows before it indicating a gap in 64B address increments): CCIX Memory Requests eligible for the ReqDevChain OpCode depending on the address of the previous CCIX Memory Request and the capabilities of the Link partners.

CCIX Device with Subordinate Agent 3:

- The CCIX Device with SA3 has indicated *SA Device Aggregation Destination Capability* via a value of 1b in the *SADevAggDestCap* field of the *SACapStat* Register. This *SA Device Aggregation Destination Capability* is enabled via a non-zero value in the *NumAggDevCntl* field *of* one or more *SBAT Entries*:
    - The *NumAggDevCntl* fields are programmed with a value of 1h to indicate the address ranges indicated in *SBAT Entry0* and *SBAT Entry1* are one of two *SA Device Aggregated Destinations*.
    - *SBAT Entry0* has its *DeviceInterleaveCntl* field programmed with a value of 2h to indicate a 256B address interleave size for the address range starting at Base Address-A programmed as the *StartAddr* with the EndAddr-B corresponding to the *MemPoolSizeCap* indicated in *Memory Pool Entry0*. Furthermore, the *AggSAIndex* field of *SBAT Entry0* is programmed with a value of 0h indicating SA3 is in the first of the two index positions of the *SA Device Aggregated Destinations*.
    - *SBAT Entry1* has its *DeviceInterleaveCntl* field programmed with a value of 1h to indicate a 128B address interleave size for the address range starting at Base Address-C programmed as the *StartAddr* with the EndAddr-D corresponding to the *MemPoolSizeCap* indicated in *Memory Pool Entry1*. Furthermore, the *AggSAIndex* field of *SBAT Entry1* is programmed with a value of 0h indicating SA3 is in the first of the two index positions of the *SA Device Aggregated Destinations*.

CCIX Device with Subordinate Agent 4:

- The CCIX Device with SA4 has its data structures programmed in a manner identical to those of the CCIX Device with SA3 described above, for the address ranges <A:B> and <C:D>, with the following two exceptions:
    - The *AggSAIndex* fields of both *SBAT Entry0 and Entry1* are programmed with a value of 1h indicating SA3 is in the second of the two index positions of the *SA Device Aggregated Destinations*.

Chapter 7. CCIX RAS Overview

This section introduces CCIX® Reliability Availability Serviceability (RAS) framework overview.

7.1 Classification of Hardware Faults

There are three categories of hardware faults (from the perspective of the error consumer) that host system software typically deals with:

1 Synchronous and Precise CPU Exceptions:

- These exceptions may be due to Data Abort, Instruction Prefetch Abort, or some form of bus error due to a hardware fault on an access to some faulting address. May be triggered by CPU load synchronous encountering an uncorrected error (UE) and/or external bus errors.
- For example:
    - Synchronous External Abort (SEA) on ARMv8 Systems.
    - Machine Check Exception (MCE) on x86 Systems.

2 Asynchronous and Imprecise CPU Exceptions:

- Triggers an exception to a dedicated CPU vector. May be triggered by CPU internal UEs, corrected errors (CEs), and/or external bus errors.
- For example:
    - System Error Interrupt (SEI) on ARMv8 Systems.
    - Machine Check Exception (MCE) and/or Corrected Machine Check (CMC) on x86 Systems.
    - Some systems may rely on normal interrupts for asynchronous hardware error notification.

3 Asynchronous and Imprecise System Exceptions or Interrupts:

- These are exceptions that happen outside of the CPU subsystem, typically reported to the CPU via wired interrupt and/or message signaled interrupt (MSI), which would notify the CPU that a hardware fault has occurred. May be triggered by UEs/CEs from system cache, memory, PCIe (e.g. AER or ATS) errors.

The handling of CCIX errors will be the focus of this document. CCIX errors may fall into any of the above categories depending on the specifics of the error manifests. CCIX errors that are consumed inside the CPU subsystem will typically fall into either the first or the second categories. CCIX errors that are consumed outside of the CPU subsystem will typically fall into the third category.

7.2 Hardware Error Propagation

The host system and the CCIX device can have similar sets of components:

- Various Types of Memory.
- Various Types of Caches and Register Files.

- Processing Elements.
- Coherent Interconnects.

Each of the above components may consume, produce, or simply detect a hardware error. These hardware errors may be related to ECC, command failure, transport errors, decode errors, logic errors, timeouts/watchdog, etc.

The following example illustrates how poisoned data may flow through a host system that is attached to a CCIX device. When data is poisoned, an error may propagate throughout the system from one entity to another, which may be triggered by a transaction initiated by a processing element and/or cache/snoop logic. The expectation would be that while the error propagates, it is still contained to the particular granule size (e.g. cache line).

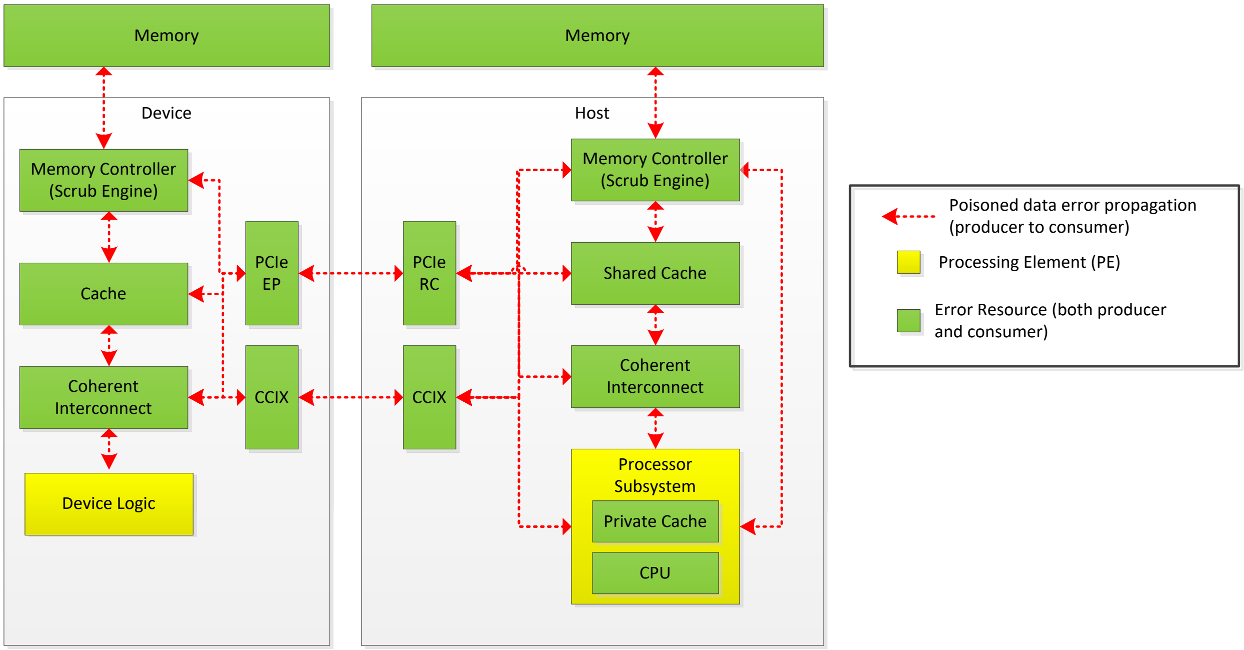


Figure 7-1: Example Error Propagation Flow on CCIX devices

The specifics of how errors are handled on the host side are either in an implementation defined manner or via a standardized mechanism per the CPU host architecture. These details could be silicon vendor-specific or compliant with some architecture defined outside of the scope of this specification.

The specifics of how CCIX errors are reported and handled will be described in this document.

7.3 CCIX Protocol Error Reporting (PER)

There are two categories of hardware faults that can cause CCIX hardware errors:

- PCIe Transport Errors:
  - These are reported via standard PCIe mechanisms (e.g. Advanced Error Reporting (AER)).
- CCIX Protocol Errors (PER):

    - On the device side, protocol errors are reported via CCIX PER Message (Section 7.3.1) and logged to CCIX DVSEC space (Section 6.2).
    - On the host side, these errors are reported to the error agent and the way the error agent reports the error to the CPU is either in an implementation defined manner or via standardized mechanism per the CPU host architecture, which is outside the scope of this specification.

CCIX devices must comply with the following requirements when it comes to reporting and logging protocol errors:

- An **error producer or consumer** of a CCIX protocol error must:
    - Always log the error in the appropriate DVSEC structure (depending on the source agent/component).
    - Send a PER message to the "Error Agent" (routed by Error AgentID described in Chapter 6).
        - The Error AgentID provides the path to the host and is described in more detail in Chapter 3.
- In addition to the above rules, an error producer of a CCIX protocol error (e.g. SA, CCIX port, CCIX Link, RA, or HA) must attempt to poison data on uncorrected data errors whenever possible. This applies to both ECC errors as well as general bus or address decode errors that may result in loss or corruption of data.
    - If poisoning is not possible and CCIX communication becomes unsafe, the error severity bit SevNocomm must be set (see Table 7-1).
- If the error producer is also the consumer of that error (e.g. RA was the first to detect an ECC error on data that it is about to operate on), it must also follow the error consumer rules described below.

IMPLEMENTATION NOTE

It is an implementation choice for the device to either provide two separate PER messages for detection and consumption, or whether to consolidate that error syndrome in a single PER message. If two PER messages are sent, this will result in a multiple error scenario as described in Section 7.3.2.2 . The actual consumption of an error is considered to be of higher severity, and so it would be expected that the error log reflects the details around the error consumption, overriding any log of the error production/detection.

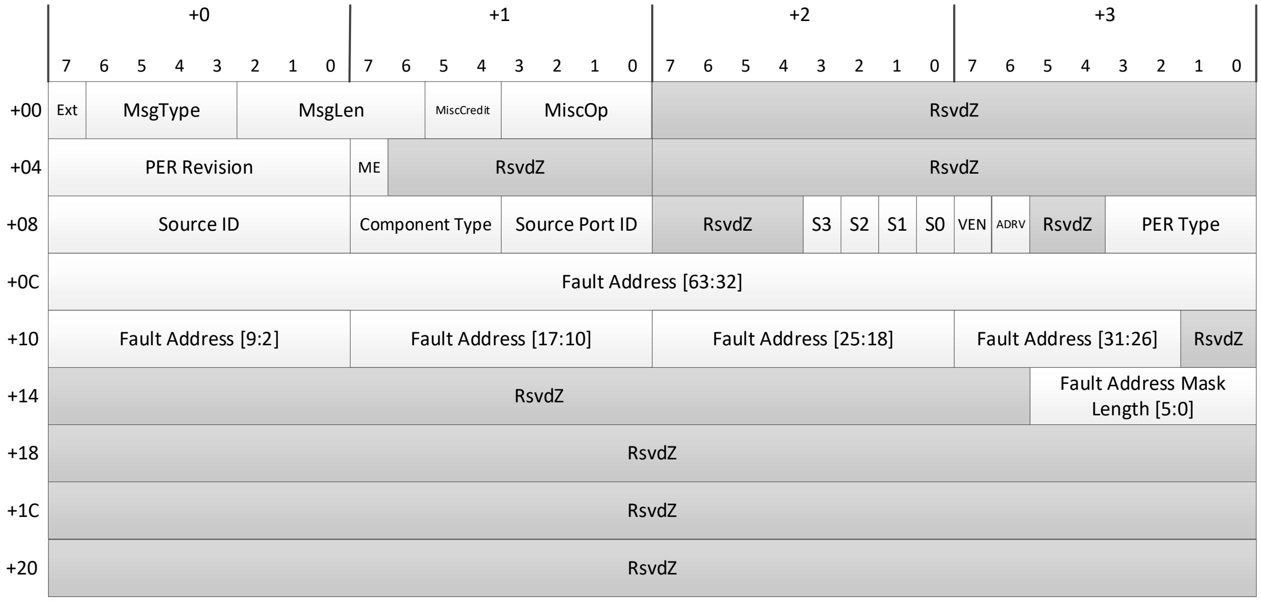
- In addition to the above rules, an error consumer of a CCIX protocol error (e.g. RA, HA, switch) must not operate on the erroneous data:
    - If possible, make an attempt to avoid further actions/operations that will create new errors.
- PER message may be masked by the host:
    - Section 7.4 describes the mechanism for masking CCIX protocol errors for a particular device.
- PER message is always routed from CCIX device to the host via the Error AgentID as specified in the CCIX DVSEC specification.

CCIX RAS introduces a new CCIX message for reporting CCIX protocol errors to the host. This section also describes examples of potential software actions when the host software becomes notified of a CCIX protocol error.

7.3.1 CCIX PER Message Format

This section describes the CCIX PER message details. The CCIX PER message is an ID routed message type (as described in Chapter 3 – it is a “Miscellaneous Message Type”) that carries minimal mandated error information needed for the host to handle the error. PER message is sent over the credit-less message classes of CCIX protocol link. At PCIe transport level, this message, like all CCIX protocol messages, will be sent over PCIe CCIX VC.

The PER message is sent by the error reporting device to the programmed CCIX Port (determined by the Error AgentID per Chapter 3). Figure 7-2 describes the format of the CCIX PER message. The field definitions of this message will have the same field definitions as the CCIX PER Log Common Header Structure, which is described in more detail in Section 7.3.2.1



<table>
<tr><th></th><th colspan="8">+0</th><th colspan="8">+1</th><th colspan="8">+2</th><th colspan="8">+3</th></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>+00</td><td colspan="1">Ext</td><td colspan="4">MsgType</td><td colspan="5">MsgLen</td><td colspan="2">MiscCredit</td><td colspan="4">MiscOp</td><td colspan="16">RsvdZ</td></tr>
<tr><td>+04</td><td colspan="8">PER Revision</td><td colspan="1">ME</td><td colspan="7">RsvdZ</td><td colspan="16">RsvdZ</td></tr>
<tr><td>+08</td><td colspan="8">Source ID</td><td colspan="4">Component Type</td><td colspan="4">Source Port ID</td><td colspan="4">RsvdZ</td><td>S3</td><td>S2</td><td>S1</td><td>S0</td><td>VEN</td><td>ADRV</td><td colspan="2">RsvdZ</td><td colspan="4">PER Type</td></tr>
<tr><td>+0C</td><td colspan="32">Fault Address [63:32]</td></tr>
<tr><td>+10</td><td colspan="8">Fault Address [9:2]</td><td colspan="8">Fault Address [17:10]</td><td colspan="8">Fault Address [25:18]</td><td colspan="6">Fault Address [31:26]</td><td colspan="2">RsvdZ</td></tr>
<tr><td>+14</td><td colspan="26">RsvdZ</td><td colspan="6">Fault Address Mask Length [5:0]</td></tr>
<tr><td>+18</td><td colspan="32">RsvdZ</td></tr>
<tr><td>+1C</td><td colspan="32">RsvdZ</td></tr>
<tr><td>+20</td><td colspan="32">RsvdZ</td></tr>
</table>

Figure 7-2: CCIX Protocol Error Reporting (PER) Message Format

The details of the fields in DW0 are described in Chapter 3. The details of the fields in DW1 and DW2 are described in Table 7-1 and Table 7-2, respectively. The fault address is provided in DW3 and DW4.

The fault address mask length is provided in byte 3 of DW5. When the fault address is valid, the fault address mask length indicates the number of contiguous mask bits, used to specify the granularity of an error. For example: an error that is detected on a 64-byte cache line granularity would report a value of 6 in byte 3 of DW5 of the PER message.

The remaining DW6, DW7, and DW8 are reserved for future use.

Table 7-1: CCIX PER Message DW 1

Bit Location	Field Description	Attibutes
7:0	**PER Revision (PerRev)** This field indicates the revision of the PER message. 0x01: Revision 1 (the current revision). All other values are reserved.	RO
14:8	Reserved and Zero	RsvdZ
15	**Multiple Errors (MultiErr)** This field indicates that multiple errors were detected at that CCIX agent or component. 0: Single error • A single error was detected and logged. The PER log will hold the syndrome information of this error. 1: Multiple errors • Multiple errors were detected, the only error that is logged was the first error detected at this severity (i.e. CE vs. UE per SevUe bit described in the next DW). Subsequent errors of the same or lower severity will not be reported or logged.	RO
31:16	Reserved and Zero	RsvdZ

Table 7-2: CCIX PER Message DW 2

Bit Location	Field Description	Attributes
7:0	**Error Source ID (ErrSrcId)** If the agent type is an DA, HA, SA, or RA: This field indicates the CCIX AgentID of the component that reported this error. In this case bits 7:6 must be zero, since AgentID is only 6 bits. The AgentID should be used to derive the BDF. Otherwise: This specifies the CCIX Device ID (i.e. in the case of Port, CCIX Link, or device errors).	RO
11:8	**Error Source Port ID (ErrPortId)** If the CCIX protocol component type is a Port or CCIX Link: This field indicates the CCIX Port ID that reported this error. The combination ErrSrcId and ErrPortId should be used to derive the BDF. Otherwise: This field is not valid.	RO

Bit Location	Field Description	Attributes
15:12	**Error Component Type (ErrCompType)** This field indicates the type of CCIX protocol component that reported this error. 0x0: RA 0x1: HA 0x2: SA 0x3: Port 0x4: CCIX Link 0x5: DA All other values are reserved.	RO
16	**Uncorrected Error (SevUe)** When this bit is set, hardware is indicating that it was unable to correct or recover from the error. 0: Corrected / Informational • Recommend System Action: Refer to SevDegraded to determine if a CE threshold has been reached. Refer to SevDeferred to determine if this is an unconsumed error that may have triggered poison creation. Refer to MultiErr to determine if multiple errors have occurred. Log the error. 1: Uncorrected • Recommend System Action**:** Refer to SevNocomm, SevDegraded, and SevDeferred to determine action.	RO

Bit Location	Field Description	Attributes
17	**CCIX Protocol Communication Broken (SevNocomm)** This field is invalid/ignored if SevUe is zero. When this bit is set, hardware indicates that CCIX protocol communication is broken, and is unsafe to proceed with CCIX protocol communication. 0: CCIX protocol Communication is safe • Recommend System Action: Refer to remaining syndrome info for details (e.g. address), MultiErr, and SevDegraded. • Action depends on system/software configuration details when it comes to that particular CCIX endpoint. The system action here will be platform specific. • For example: Uncorrected memory errors may be attributed to an address or address mask (i.e. address range). There may be software knowledge about that particular CCIX device to determine whether the error may be isolated or contained. o If Address available (AddrValid is 1): Software may attempt to recover without restarting (depending on the criticality of the data that was lost/corrupted) via page offlining mechanisms. o If Address is not available (AddrValid is 0): Depending on how isolated the CCIX endpoint is, it may involve resetting the CCIX end point and restarting applications/VMs associated with that endpoint, or if the CCIX endpoint is critical to system operation, this may require a full system reboot. 1: CCIX Protocol Communication is unsafe • In cases where an uncorrected error is lost, this bit must be set. • Recommend System Action: Action is implementation-dependent, which may depend on how isolated the CCIX endpoint is. • For example: the action may involve resetting the CCIX agent, restarting applications/VMs associated with that agent, or (if the CCIX agent is critical to system operation) may require a full system reboot. • Parameters to make these decisions could vary between platforms due to differences in system RAS policies. • In case of multiple error scenarios, loss of UEs must result in broken CCIX protocol communication. If so, this must be reflected by setting this bit.	RO

Bit Location	Field Description	Attributes
18	**Degraded / Threshold (SevDegraded)** When this bit is set, the CCIX device is operating in a degraded reliability mode and device failure is more likely. Hardware may set this bit when it has reached some type of error threshold or critical error condition (e.g. CE threshold or unlikely/unexpected UEs). The setting of thresholds are device/vendor-specific. It may be hard coded by hardware or programmable via device driver or firmware. The specifics of setting thresholds is outside the scope of this specification.	RO
19	**Deferrable (SevDeferred)** When this bit is set, the error was detected by the agent or device, but has not yet been consumed. In the case of a data error, this bit may be used to indicate poison creation. Action is implementation-dependent. This field is only valid if SevUe=0. The act of poison creation is typically informational and must be reported from the data source that detected the error. If the error is eventually consumed by the target/destination agent, that agent will report the error as an uncorrected error (i.e. SevUe=1). It is possible for the poisoned data to be consumed by multiple agents on a data path and therefore reported multiple times. It is implementation-dependent whether or not intermediate agents/components that are passing along poisoned data send a PER message or log the poison. Some system/device implementations may choose to refrain from reporting the error until the poisoned data arrives at its final destination (if possible), and therefore limit reporting of the error to the source and destination.	RO
23:20	Reserved and Zero	RsvdZ

Bit Location	Field Description	Attributes
27:24	**PER Type (PerType)** This field indicates the PER type. When VEN=0, this field selects one of the following architectural PER types: 0x0: Memory Error Type Structure (valid only if ErrCompType is HA or SA) – described in Section 7.3.3. 0x1: Cache Error Type Structure (valid only if ErrCompType is RA, HA, or SA) – described in Section 7.3.4. 0x2: ATC Error Type Structure (valid only if ErrCompType is RA) – described in Section 7.3.5. 0x3: Port Error Type Structure (valid only if ErrCompType is Port) – described in Section 7.3.6. 0x4: CCIX Link Error Type Structure (valid only if ErrCompType is CCIX Link) – described in Section 7.3.7. 0x5: Agent Internal (valid for all) – described in Section 7.3.8. 0x6: DVM Error Type Structure (valid only if ErrCompType is DA) All other values are reserved.	RO
29:28	Reserved and Zero	RsvdZ
30	**Address Valid Flag (AddrValid)** This field is set when the fault base address fields in the CCIX PER message is valid. 0: Fault Address is unknown • For Error Agent types of HA or SA: Software may assume that the Fault Address Mask is based on the BAT. • For Error Agent Types of RA: Software may assume that the Fault Address Mask is all of memory (any data in memory could be corrupted). 1: Fault Address is known • The Fault Address fields (DW3 and DW4) in the PER message provides a valid address. • The Fault Address Mask Length is only valid if this bit is set to one. • It is assumed that the host hardware may only take action on the cache line granularity due to lack of address mask in the PER message. Errors with larger granularity will depend on software action, since software will have access to the PER log.	RO
31	**Vendor-Specific Error Type Flag (VenErrTypeFlag)** This field is set for CCIX protocol errors that cannot be described by PER Types. The PerType field will be implementation defined when this field is set. All other fields in the PER message must still be set per this specification.	RO

7.3.2 CCIX PER Log Structures

The CCIX PER Log is intended for reporting hardware faults that will result in protocol errors. The protocol errors will normally be associated with a particular CCIX protocol component (i.e., CCIX Link, Port, DA, HA, SA, or RA). The offset, in number of bytes, to the start of the PER log structure is indicated by the Error Log Offset of the various capability structures, which are shown in Section 6.2.2. The Error Log Offset must be in integer multiples of DW.

IMPLEMENTATION NOTE

The Error Log location, indicated for each component in the Error Log Offset, may be unique per CCIX protocol component, or could also be shared across multiple CCIX protocol components (within the same PCIe BDF).

Devices with a common Error Log location shared across ALL CCIX protocol components on that Device can indicate that common location in the Device Error Log Offset field.

Error log from a CCIX device is made available through a RO space in Device’s DVSEC. This means that once a log is captured, it cannot be overwritten by future errors with the same or lesser severity (see Section 7.3.2.2). The error log must be sticky and the host must retain the error log even in scenarios where the link goes down.

Figure 7-3 describes the CCIX PER Log Format. The first eight DW of the PER log structure is a log header.. Some of these fields share a common definition and format with PER message. The PER Log Header provides the fault address and fault address mask length (if available, ADRV=1). The remaining structure field definitions depend on the “PER Type”.

The following subsections describe the error log field details. Each error log field may be classified as Mandatory (M) or Optional (O). The classification is indicated by the “M/O” column in the tables provided in this section.

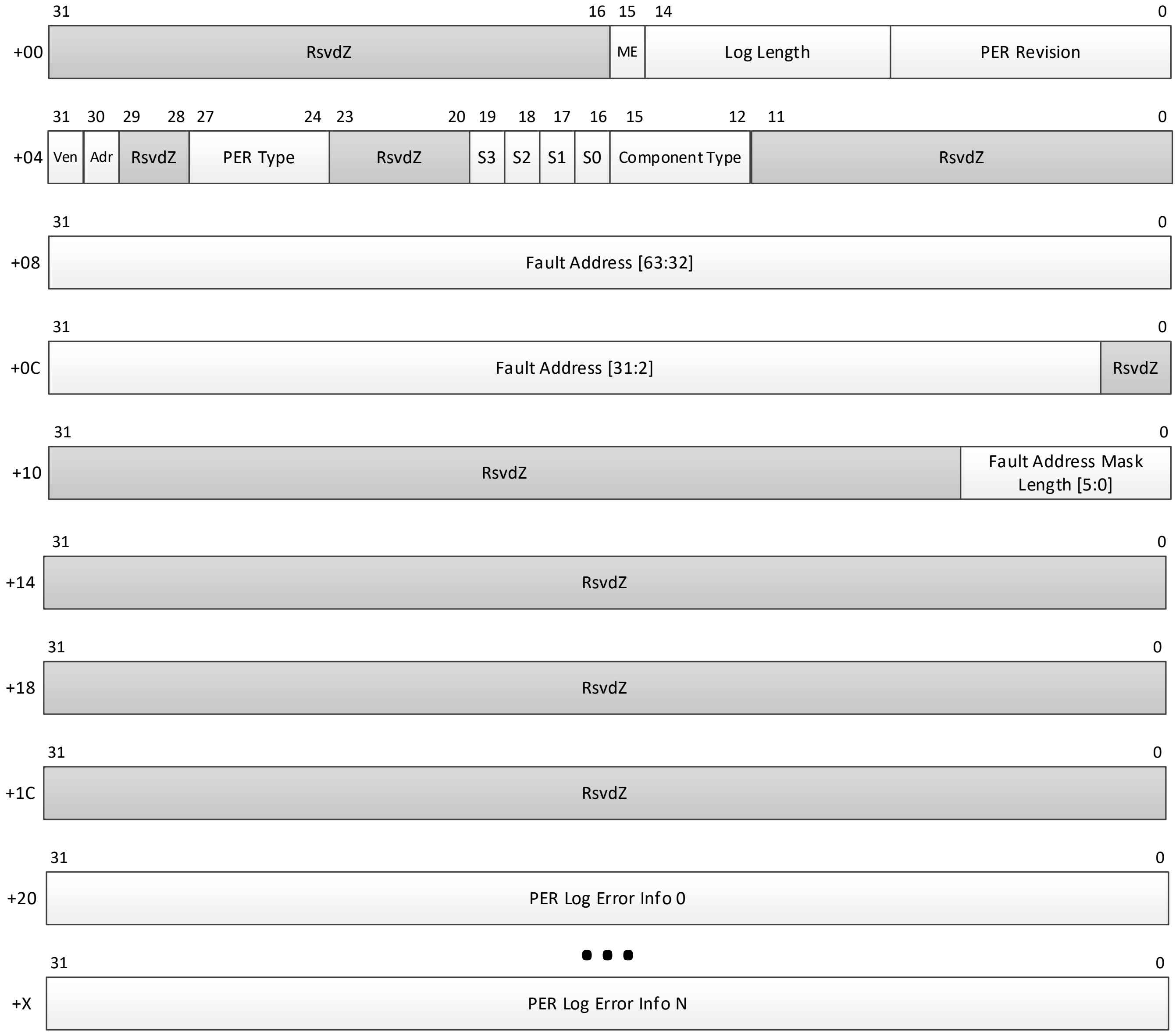


Figure 7-3: CCIX PER Log Structure Format

7.3.2.1 CCIX PER Log Header

The PER Log Header consists of 256 bits. DW0 and DW1 represent the error log attribute fields, which are described in Table 7-3 and Table 7-4. The third and fourth DWs provide the fault address of the error, as described in Section 7.3.1. The fifth DW provide the fault address mask length of the error, as described in Section 7.3.1. The sixth, seventh, and eighth DW are reserved for future use.

Table 7-3: CCIX PER Log Header DW 0

Bit Location	Field Description	Attribute	M/O
7:0	See Table 7-1 for definition of this field, substituting "PER Log header" for "PER message".	RO	M
14:8	**Log Length (LogLen)** This field describes the number of DWs in the log (including the log header). For errors that do not have any additional error info, this field must be the size of the log header (0x20).	RO	M
15	See Table 7-1 for definition of this field.	RO	M
31:16	Reserved and Zero	RsvdZ	M

Table 7-4: CCIX PER Log Header DW 1

Bit Location	Field Description	Attribute	M/O
11:0	Reserved and Zero	RsvdZ	M
15:12	See Table 7-2 for definition of this field.	RO	M
16	See Table 7-2 for definition of this field.	RO	M
17	See Table 7-2 for definition of this field.	RO	M
18	See Table 7-2 for definition of this field.	RO	M
19	See Table 7-2 for definition of this field.	RO	M
23:20	Reserved and Zero	RsvdZ	M
27:24	See Table 7-2 for definition of this field.	RO	M
29:28	Reserved and Zero	RsvdZ	M
30	See Table 7-2 for definition of this field, substituting "PER Log header" for "PER message".	RO	M
31	See Table 7-2 or definition of this field.	RO	M

7.3.2.2 Overwrite/Overflow Rules in Multiple Error Scenarios

This subsection describes how a CCIX device/agent must handle multiple error scenarios. When an error occurs while the PER log is still populated, due to not yet being acknowledged and cleared by software, the device will send a PER message with the MultiErr bit set and update the PER log if the following actions apply:

- The MultiErr bit is not already set in the PER log, update the PER log to set the MultiErr bit.
  - If the error is of the same or lower severity, PER log is maintained except for setting of the MultiErr bit.

  - If the error is of higher severity (e.g. first error was a CE, second error was a UE), then PER log will be updated with the new syndrome information in addition to setting of the MultiErr bit.
- The MultiErr bit is already set, but the new error is of higher severity. In this case, the PER log will be updated with the new syndrome information.

The severity priority from lowest to highest is enumerated in Table 7-5.

Table 7-5: CCIX Error Severity Priorities (lowest to highest)

Severity Priority	SEV0 (SevUe)	SEV1 (SevNocom)	SEV2 (SevDegraded)	SEV3 (SevDeferred)
1	0	0	0	0
2	0	0	1	0
3	0	0	0	1
4	0	0	1	1
5	1	0	X	0
6	1	1	X	0

Each time the PER log is altered due to one of the above scenarios, this must also result in a PER message reflecting this change in the log. In the below example, the host may have received four PER messages, but the PER log only retains the final state.

For example:

- A multi-error scenario on the same agent will result in multiple PER messages.
- An initial single corrected error is logged and triggers a PER message to be sent with MultiErr=0 and SevUe=0.
- On a second corrected error, the log is updated to MultiErr=1, maintaining the existing syndrome information of the first error, followed by a PER message to reflect the new state of the PER log.
- An uncorrected error is then detected (SevUe=1), the log is updated to capture the syndrome of the uncorrected error (since the new error is of higher severity), followed by a PER message to reflect the new state of the PER log.
- On a second uncorrected error, the log should be updated to upgrade SevNocomm=1, followed by a PER message to reflect the new state of the PER log. Due to the loss of the syndrome of the second UE, the address of the data that has been corrupted is unknown, rendering further communication unsafe.

The loss of UEs is typically considered more severe than loss of CEs.

For example:

- In cases where SevUe=1 and MultiErr=1, the host must also refer to SevNocomm to determine whether a CE or UE is lost.
- If CE is lost, the device may set SevNocomm=0: Software may just handle the UE in the PER log, and log that multiple CEs have occurred (not having syndrome of the CE that was lost).

- If UE is lost, the device must set SevNocomm=1 (due to not having the syndrome of the UE that was lost): See recommended actions described in SevNocomm.

IMPLEMENTATION NOTE

Host side hardware may implement PER queues and hardware counters to handle and store multiple errors. How hardware advertises these features is implementation dependent.

7.3.3 Memory Error Type Structure

Table 7-6 describes the CCIX PER Memory Error Type Structure. This structure is variable length and starts at sixth DW of the PER log.

Some field definitions in this structure are based on existing structures/fields defined in the *UEFI Specification* Version 2.7 Appendix N, Common Platform Error Record (CPER) (see Reference Documents).

Table 7-6: CCIX PER Memory Error Type Structure

Byte Location	Size (Bytes)	Register Description	Attribute	M/O
0	4	**Validation Bits** All subsequent fields will have a validation bit to indicate whether that field has valid data. Bit 0: Generic Memory Type Bit 1: Operation Bit 2: Memory Error Type Bit 3: Card (Channel) Bit 4: Bank Bit 5: Device Bit 6: Row Bit 7: Column Bit 8: Rank Bit 9: Bit Position Bit 10: Chip Identification Bit 11: Vendor-Specific Log Info Bit 11 is intended to indicate that there is vendor-specific log info provided with this error log. Bit 12: Module Bit 13: Specific Memory Type All other bits are reserved. Note: The order of the Validation Bits in Table 7-6 does not, in all cases, follow the order of	RO	M

Byte Location	Size (Bytes)	Register Description	Attribute	M/O
		fields in that same table. This is unlike Table 7-7 and Table 7-8 for example, where the order of the Validation Bits follow the order of fields in that same table.		
4	1	**FRU ID** This field indicates a generic instance ID for identifying the location of the field replaceable unit (FRU). The value of 0xFF is reserved for errors that are detected in "on-chip memory" that is exposed to software. In this case, it is assumed the FRU is the actual CCIX device itself. For example: On a device that supports up to 4 channels and 2 DIMMs per channel, that device supports up to 8 DIMMs populated. The value of this field will indicate which of the 8 possible DIMM slots raised the error condition.	RO	M
5	1	Reserved and Zero	RsvdZ	M
6	2	**Length** This field indicates the length of this memory error type structure in bytes (including the Vendor-Specific Log Info).	RO	M
8	1	**Memory Pool Generic Memory Type Capability** This field indicates one of the following memory types. 0x00: Other, Non-Specified 0x01: ROM 0x02: Volatile Memory 0x03: Non-Volatile Memory 0x04: Device/Register Memory 0x80-0xFF: Values are vendor-specific All other values are reserved.	RO	O
9	1	**Operation Type** This field indicates one of the following operation types. 0: Generic 1: Read 2: Write 4: Scrub All other values are reserved.	RO	O
10	1	**Memory Error Type**	RO	O

Byte Location	Size (Bytes)	Register Description	Attribute	M/O
		This field indicates the type of error that occurred. 0: Unknown 1: No error 2: Single-bit ECC 3: Multi-bit ECC 4: Single-symbol ChipKill ECC 5: Multi-symbol ChipKill ECC 6: Master abort 7: Target abort 8: Parity Error 9: Watchdog timeout 10: Invalid address 11: Mirror Broken 12: Memory Sparing 13: Scrub 14: Physical Memory Map-out event All other values are reserved.		
11	1	**Card or Channel Number (Chan)** This field indicates the card or channel number (in hex) of the error location. For example: On a device that supports up to 4 channels, this field indicates which channel raised the error.	RO	O
12	2	**Module (Mod)** This field indicates the module number (in hex) of the memory error location. (Channel Number and Module should provide the information necessary to identify the failing FRU). For example: On a device that supports up to 4 channels and 2 DIMMs per channel, the value of this field may be between zero or one, to indicate which on the two DIMMs raised an error.	RO	O
14	2	**Bank** This field indicates the bank number (in hex) of the memory associated with the error. When Bank is addressed via group/address (e.g., DDR4). Bit 7:0 – Bank Address Bit 15:8 – Bank Group Device	RO	O

Byte Location	Size (Bytes)	Register Description	Attribute	M/O
16	4	**Device** This field indicates the device number (in hex) of the memory associated with the error.	RO	O
20	4	**Row** This field indicates the row number (in hex) of the memory error location.	RO	O
24	4	**Column** This field indicates the column number (in hex) of the memory error location.	RO	O
28	4	**Rank** This field indicates the rank number (in hex) of the memory error location.	RO	O
32	1	**Bit Position** This field indicates the bit position (in hex) at which the memory error occurred.	RO	O
33	1	**Chip Identification** This is an encoded field used to address the die in 3DS packages.	RO	O
34	1	**Memory Pool Specific Memory Type Capability** This field indicates one of the following memory types. 0x00: Other, Non-Specified 0x01: SRAM 0x02: DDR 0x03: NVDIMM-F 0x04: NVDIMM-N 0x05: HBM 0x06: Flash 0x80-0xFF: Values are vendor-specific All other values are reserved.	RO	O
35	(indicated by VenLen)	**Vendor-Specific Log Info** This is a variable length field that may be used by vendors for additional error information. This structure is described in Section 7.3.9.	RO	O

7.3.4 Cache Error Type Structure

Table 7-7 describes the CCIX PER Cache Error Type Structure. This structure is variable length and starts at sixth DW of the PER log.

Some field definitions in this structure are based on existing structures/fields defined in the UEFI 2.7 Appendix N Common Platform Error Record (CPER) (see Reference Documents).

Table 7-7: CCIX PER Cache Error Type Structure

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
0	4	**Validation Bits** All subsequent fields will have a validation bit to indicate whether the field has valid data. Bit 0: Cache Type Bit 1: Operation Bit 2: Cache Error Type Bit 3: Cache Level Bit 4: Set Bit 5: Way Bit 6: Cache Instance ID Bit 7: Vendor-Specific Log Info Bit 7 is intended to indicate that there is vendor-specific log info provided with this error log. All other bits are reserved.	RO	M
4	2	**Length** This field indicates the length of this cache error type structure in bytes (including the Vendor-Specific Log Info).	RO	M
6	1	**Cache Type** This field indicates one of the following cache types. 0x0: Instruction Cache 0x1: Data Cache 0x2: Generic / Unified Cache 0x3: Snoop Filter Directory All other values are reserved.	RO	O

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
7	1	**Operation Type** This field indicates one of the following operation types. 0: Generic Error (type of error cannot be determined) 1: Generic Read (type of instruction or data request cannot be determined) 2: Generic Erite (type of instruction or data request cannot be determined) 3: Data Read 4: Data Write 5: Instruction Fetch 6: Prefetch 7: Eviction 8: Snooping (the request agent described in this log initiated a cache snoop that resulted in an error) 9: Snooped (the request agent described in this log raised a cache error caused by another processor or device snooping into its cache) 10: Management or Command Error All other values are reserved.	RO	O
8	1	**Cache Error Type** This field indicates the type of error that occurred. 0: Data 1: Tag 2: Timeout 3: Hang 4: Data Lost 5: Invalid Address All other values are reserved.	RO	O
9	1	**Cache Level** This field indicates the level of the cache where the error was detected (relative to the agent).	RO	O
10	4	**Set** This field indicates the set number (in hex) where the error was detected.	RO	O
14	4	**Way** This field indicates the way number (in hex) where the error was detected.	RO	O
18	1	**Cache Instance ID** This field indicates the instance number (in hex) or ID of the cache where the error was detected (relative to the agent).	RO	O

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
19	1	Reserved and Zero	RsvdZ	O
20	(indicated by VenLen)	**Vendor-Specific Log Info** This is a variable length field that may be used by vendors for additional error information. This structure is described in Section 7.3.9.	RO	O

7.3.5 ATC Error Type Structure

Table 7-8 describes the CCIX PER ATC Error Type Structure. This structure is variable length and starts at sixth DW of the PER log.

Table 7-8: CCIX PER ATC Error Type Structure

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
0	4	**Validation Bits** All subsequent fields will have a validation bit to indicate whether the field has valid data. Bit 0: Operation Type Bit 1: ATC Instance ID Bit 2: Vendor-Specific Log Info Bit 2 is intended to indicate that there is vendor-specific log info provided with this error log. All other bits are reserved.	RO	M
4	2	**Length** This field indicates the length of this ATC error type structure in bytes (including the Vendor-Specific Log Info).	RO	M

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
6	1	**Operation Type** This field indicates one of the following operation types. 0: generic error (type of error cannot be determined) 1: Generic Read (type of instruction or data request cannot be determined) 2: Generic Write (type of instruction or data request cannot be determined) 3: Data Read 4: Data Write 5: Instruction Fetch 6: Prefetch 7: Eviction 8: Snooping (the request agent described in this log initiated a cache snoop that resulted in an error) 9: Snooped (The request agent described in this log raised a cache error caused by another processor or device snooping into its cache) 10: Management or Command All other values are reserved.	RO	O
7	1	**ATC Instance ID** This field indicates the instance number (in hex) or ID of the cache where the error was detected (relative to the agent).	RO	O
8	4	Reserved and Zero	RsvdZ	O
12	(indicated by VenLen)	**Vendor-Specific Log Info** This is a variable length field that may be used by vendors for additional error information. This structure is described in Section 7.3.9.	RO	O

7.3.6 Port Error Type Structure

Table 7-9 describes the CCIX PER Port Error Type Structure. This structure is variable length and starts at sixth DW of the PER log.

Table 7-9: CCIX PER Port Error Type Structure

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
0	4	**Validation Bits** All subsequent fields will have a validation bit to indicate whether the field has valid data. Bit 0: Operation Bit 1: Port Error Type Bit 2: CCIX Message Bit 3: Vendor-Specific Log Info Bit 3 is intended to indicate that there is vendor-specific log info provided with this error log. All other bits are reserved.	RO	M
4	2	**Length** This field indicates the length of this port error type structure in bytes (including the Vendor-Specific Log Info).	RO	M
6	1	**Operation Type** This field indicates one of the following operation types. 0: Command Error 1: Read 2: Write All other values are reserved.	RO	O
7	1	**Port Error Type** This field indicates the type of error that occurred. 0: Generic Bus / Subordinate Error 1: Bus Parity / ECC error (in transit to agent) 2: Decode Error (BDF not present) 3: Decode Error (Invalid Address) 4: Decode Error (Invalid AgentID) 5: Bus Timeout 6: Hang 7: Egress Blocked All other values are reserved.	RO	O
8	32	**CCIX Message** This field indicates the raw CCIX message that resulted in an error.	RO	O

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
40	(indicated by VenLen)	**Vendor-Specific Log Info** This is a variable length field that may be used by vendors for additional error information. This structure is described in Section 7.3.9.	RO	O

7.3.7 CCIX Link Error Type Structure

Table 7-10 describes the CCIX Link Error Type Structure. This structure is variable length and starts at sixth DW of the PER log.

Table 7-10: CCIX Link Error Type Structure

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
0	4	**Validation Bits** All subsequent fields will have a validation bit to indicate whether the field has valid data. Bit 0: Operation Bit 1: Link Error Type Bit 2: Link ID Bit 3: Credit Type Bit 4: CCIX Message Bit 5: Vendor-Specific Log Info Bit 5 is intended to indicate that there is vendor-specific log info provided with this error log. All other bits are reserved.	RO	M
4	2	**Length** This field indicates the length of this port error type structure in bytes (including the Vendor-Specific Log Info).	RO	M
6	1	**Operation Type** This field indicates one of the following operation types. 0: Command Error 1: Read 2: Write All other values are reserved.	RO	O

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
7	1	**Link Error Type** This field indicates the type of error that occurred. 0: Generic Link Error 1: Link Credit Underflow 2: Link Credit Overflow 3: Unusable Credit Received 4: Link Credit Timeout All other values are reserved.	RO	O
8	1	**Link ID** This field indicates ID of the link associated with this error.	RO	O
9	1	**Link Error Credit Type** This field indicates the Credit Type associated with this error. 0: Memory Credit 1: Snoop Credit 2: Data Credit 3: Misc Credit All other values are reserved.	RO	O
10	2	Reserved and Zero	RsvdZ	M
12	32	**CCIX Message** This field indicates the raw CCIX message that resulted in an error.	RO	O
44	(indicated by VenLen)	**Vendor-Specific Log Info** This is a variable length field that may be used by vendors for additional error information. This structure is described in Section 7.3.9.	RO	O

7.3.8 Agent Internal Error Type Structure

Table 7-11 describes the CCIX PER Agent Internal Error Type Structure. This structure is variable length and starts at sixth DW of the PER log.

Table 7-11: CCIX PER Agent Internal Error Type Structure

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
0	4	**Validation Bits** All subsequent fields will have a validation bit to indicate whether the field has valid data. Bit 0: Vendor-Specific Log Info Bit 0 is intended to indicate that there is vendor-specific log info provided with this error log. All other bits are reserved.	RO	M
4	2	**Length** This field indicates the length of this agent internal error type structure in bytes (including the Vendor-Specific Log Info).	RO	M
6	2	Reserved and Zero	RsvdZ	O
8	(indicated by VenLen)	**Vendor-Specific Log Info** This is a variable length field that may be used by vendors for additional error information. This structure is described in Section 7.3.9.	RO	O

7.3.9 Vendor-Specific Log Info

Table 7-12 describes the format of the Vendor-Specific Log Info. This structure is variable length.

Table 7-12: Vendor-Specific Log Info

Byte Location	Size (Bytes)	Register Description	Attributes	M/O
0	2	**Vendor Info Length (VenLen)** This field indicates the length of this structure in bytes.	RO	M
2	6	Reserved and Zero	RsvdZ	M
8	(indicated by VenLenVen Len - 4)	**Vendor-Specific Info Structure** This is a variable length structure. This structure is implementation defined.	RO	O

7.4 CCIX Error Control & Status Structures

CCIX provides error control mechanisms at two levels:

- Level 1: CCIX Device Error Control & Status (per CCIX device)
  - Common device error control is provided via the “Primary Port Common Control Structure” (see Section 6.2.2.1.2). The “Device Error Control & Status Register" provides:
    - Bit to enable PER reporting from the CCIX device.
  - The layout of this register is described in more detail in Section 7.4.1 and Table 7-13.
- Level 2: Component Error Control & Status (per CCIX protocol component)
  - Each CCIX protocol component will have “Component Error Control & Status Registers” (*ErrCntlStat0 and *ErrCntlStat1) for local error status. The following registers include:
    - Port Error Control & Status Registers (PortErrCntlStat0 and PortErrCntlStat1).
    - CCIX Link Error Control & Status Registers (LinkErrCntlStat0 and LinkErrCntlStat1).
    - HA Error Control & Status Registers (HAErrCntlStat0 and HAErrCntlStat1).
    - RA Error Control & Status Registers (RAErrCntlStat0 and RAErrCntlStat1).
    - SA Error Control & Status Registers (SAErrCntlStat0 and SAErrCntlStat1).
    - DA Error Control & Status Registers (DAErrCntlStat0 and DAErrCntlStat1).
  - The layout of each of these registers is described in Section 7.4.1, Table 7-14, Table 7-15, and Section 6.2.2.

The reset value of all error control and status registers in a CCIX device must be disabled. Host software must explicitly enable error reporting of the CCIX device and each CCIX protocol component. Once enabled, errors are unmasked by default unless mask bits are set by software.

7.4.1 Error Control Register Definitions

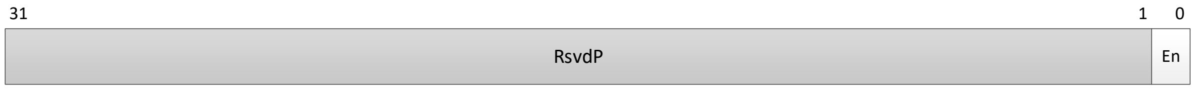


Figure 7-4: Device Error Control & Status Register (DevErrCntlStat)

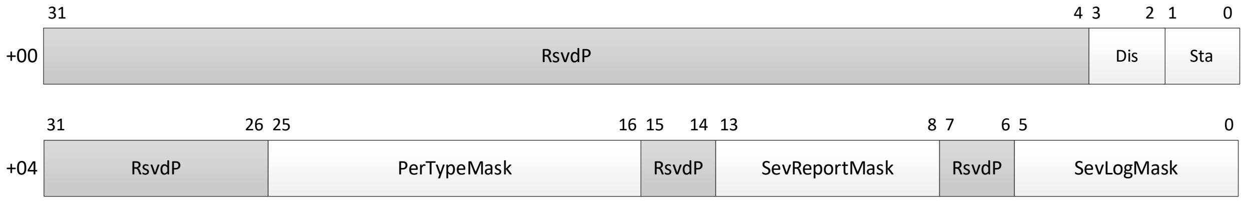


Figure 7-5: Component Error Control & Status Registers (*ErrCntlStat0 and *ErrCntlStat1)

Table 7-13 describes the bit definitions of the Device Error Control & Status Register.

Table 7-13: Device Error Control & Status Register (DevErrCntlStat) Fields

Bit Location	Field Description	Attributes	M/O
0	**Error Reporting Enable (En)** This field enables/disables both PER messages and logging on this device. 0b: PER messages will not be reported from this device. Protocol errors will not be reported once this field is cleared (if previously set). Any pending component protocol errors that have already been logged for a particular component will remain pending in the Sta field and in the PER log area (i.e. clearing this field does not clear pending errors). 1b: PER messages will be reported from this device. In the event of an unmasked protocol error, a PER log (if enabled and unmasked) will be captured and a PER message (if enabled and unmasked) will be reported to the host. Reset value of this field must be zero.	RW	M
31:1	Reserved and Preserved	RsvdP	M

Table 7-14 describes the bit definitions of the Component Error Control & Status Register 0 (*ErrCntlStat0).

Table 7-14: Component Error Control & Status Register 0 (*ErrCntlStat0) Fields

Bit Location	Field Description	Attributes	M/O
1:0	**Error Status (Sta)** This field indicates the error status of this CCIX protocol component. 00b: PER log has been cleared and there is no pending error. 01b: PER log is valid and an error is pending. A PER message was not reported. 10b: Invalid. 11b: PER log is valid, a PER message has been reported, and an error is pending. Writing these bits with all ones will clear the status back to zero and thereby clearing/invalidating the PER log (i.e. once status is cleared, the device is permitted to overwrite the log on a new error event). Writing these bits with all zeros will have no effect. Writing any other value to these bits is undefined.	RW1C	M

Bit Location	Field Description	Attributes	M/O
	(Error Status description continued) The following scenarios describe how the Sta bits would be updated (or maintained) for an upgraded PER log (as described in Section 7.3.2.2) without sending a PER message for a CCIX protocol component: • 00b -> 01b : The PER log has been updated with a new error without sending a PER message. • 00b -> 11b : The PER log has been updated with a new error and a PER message has been raised. • 01b -> 00b : Software has cleared this error status and the PER log for the CCIX protocol component is no longer valid. • 01b -> 01b : The PER log has been upgraded (as described in Section 7.3.2.2) without sending a PER message. • 01b -> 11b : A higher priority error was detected after software had enabled PER message reporting (*ErrCntlStat0.Dis cleared to zero when it was previously set to 10b) or if PER message reporting was masked on the previously logged error and unmasked on this higher priority error. The PER log has been upgraded (as described in Section 7.3.2.2) and a PER message has been raised. • 11b -> 00b : Software has cleared this error status and the PER log for the CCIX protocol component is no longer valid. • 11b -> 01b : A higher priority error was received after software had disabled PER message reporting (*ErrCntlStat0.Dis written to 10b when it was previously zero), or if PER message reporting was unmasked on the previously logged error and masked on this higher priority error. PER log has been upgraded (as described in Section 7.3.2.2) without sending a PER message • 11b -> 11b : The PER log has been upgraded (as described in Section 7.3.2.2) and a PER message has been raised. This field will only log errors if PER is enabled for the device (i.e. DevErrCntlStat.En is also set to one). Reset value of this field must be zero.		

Bit Location	Field Description	Attributes	M/O
3:2	**PER Disable (LogDis and Dis)** This field enables/disables PER logging and PER message reporting for this CCIX protocol component. 00b: PER logging and PER message reporting for this component is enabled. In the event of a protocol error, this component will capture a log and set the Sta field of this register to reflect that an error is pending. 01b: Invalid. 10b: PER message will not be reported from this component. This component will still capture a log and set the Sta field of this register to reflect that an error is pending. Subsequent protocol errors will continue to be logged, but not reported. 11b: PER logging and message reporting for this component is disabled. Protocol errors will not be logged once this field is set. Any pending error previously logged by this component will remain pending in the Sta field and in the PER log area (i.e. setting this field does not clear pending errors). Clearing of DevErrCntlStat.En will result in the device no longer reporting subsequent protocol errors. Reset value of this field must be 11b (i.e. once PER is enabled at the device, error reporting and logging remains disabled at the component by default, unless this bit is explicitly cleared prior to enabling PER at the device).	RW	M
31:4	Reserved and Preserved	RsvdP	M

Table 7-15 describes the bit definitions of the Component Error Control & Status Register 1 (*ErrCntlStat1).

Table 7-15: Component Error Control & Status Register 1 (*ErrCntlStat1) Fields

Bit Location	Field Description	Attributes	M/O
5:0	**Severity Logging Mask (SevLogMask)** This field enables masking of error logging to PER log. Bit 0: If set, logging of severity priority 1 in Table 7-5 will be masked Bit 1: If set, logging of severity priority 2 in Table 7-5 will be masked Bit 2: If set, logging of severity priority 3 in Table 7-5 will be masked Bit 3: If set, logging of severity priority 4 in Table 7-5will be masked Bit 4: If set, logging of severity priority 5 in Table 7-5 will be masked Bit 5: If set, logging of severity priority 6 in Table 7-5 will be masked Reset value of this field must be zero (i.e. no errors are masked unless explicitly set to be masked) The rules and guidelines on how/when to set mask bits are described in Section 7.4.2.	RW	M
7:6	Reserved and Preserved	RsvdP	M
13:8	**Severity Reporting Mask (SevReportMask)** This field enables masking of error reporting via PER message. Bit 0: If set, reporting of severity priority 1 in Table 7-5 will be masked Bit 1: If set, reporting of severity priority 2 in Table 7-5 will be masked Bit 2: If set, reporting of severity priority 3 in Table 7-5 will be masked Bit 3: If set, reporting of severity priority 4 in Table 7-5 will be masked Bit 4: If set, reporting of severity priority 5 in Table 7-5 will be masked Bit 5: If set, reporting of severity priority 6 Table 7-5 will be masked Reset value of this field must be zero (i.e. no errors are masked unless explicitly set to be masked). The rules and guidelines on how/when to set mask bits are described in Section 7.4.2.	RW	M
15:14	Reserved and Preserved	RsvdP	M

Bit Location	Field Description	Attributes	M/O
25:16	**PER Type Mask (PerTypeMask)** This field enables masking of errors by PER Type. Bit 0: If set, reporting of Memory errors will be masked Bit 1: If set, logging of Memory errors will be masked Bit 2: If set, reporting of Cache errors will be masked Bit 3: If set, logging of Cache errors will be masked Bit 4: If set, reporting of ATC errors will be masked Bit 5: If set, logging of ATC errors will be masked Bit 6: If set, reporting of Link and Port errors will be masked Bit 7: If set, logging of Link and Port errors will be masked Bit 8: If set, reporting of Agent Internal errors will be masked Bit 9: If set, logging of Agent Internal errors will be masked Reset value of this field must be zero (i.e., no errors are masked unless explicitly set to be masked).	RW	M
31:26	Reserved and Preserved	RsvdP	M

7.4.2 Device Error Control Flows

The following subsections describe the legal and recommended flows for enabling and disabling PER capabilities of a CCIX device. CCIX compliant devices and systems must follow the rules and flows described below.

7.4.2.1 Error Masking Rules

If software sets a mask prior to enabling of error reporting/logging at the component or device level (i.e. DevErrCntlStat.En=0 or DevErrCntlStat.En=1, *ErrCntlStat0.Dis=1, *ErrCntlStat0.LogDis=1), and all pending errors have been cleared (*ErrCntlStat0.Sta=0):

Rule: Hardware must apply the mask by the time the enable takes effect (i.e. setting of DevErrCntlStat.En=1, *ErrCntlStat0.Dis=0, *ErrCntlStat0.LogDis=0), ensuring that no errors of the masked severity and/or PER type are logged/reported (depending on what is being masked).

For example: Software may set bit 0 of *ErrCntlStat1.SevReportMask and *ErrCntlStat1.SevLogMask (to mask all CEs at the component). Hardware must guarantee that no subsequent error with severity priority 1 (per Section 7.3.2.2 is reported or logged in the future (or until this mask is cleared).

If software sets a mask after error reporting/logging at the component or device level has already been enabled (e.g. DevErrCntlStat.En=1, *ErrCntlStat0.Dis=0, *ErrCntlStat0.LogDis=0), and all pending errors have been cleared (*ErrCntlStat0.Sta=0):

Rule: Hardware must apply the mask in a timely manner, ensuring that no errors of the masked severity and/or PER type are logged/reported (depending on what is being masked). This mask must take effect prior to an immediate subsequent read of the same *ErrCntlStat0 and *ErrCntlStat1 Registers.

> **For example**: Software may set bit 0 of *ErrCntlStat1.SevReportMask and *ErrCntlStat1.SevLogMask (to mask all CEs at the component), do a subsequent read of the *ErrCntlStat0 and *ErrCntlStat1 Registers. Software then reads *ErrCntlStat0 and *ErrCntlStat1 Registers and finds that no errors are pending. Hardware must guarantee that no subsequent error with severity priority 1 (per Section 7.3.2.2) is reported or logged in the future (or until this mask is cleared).

If software sets a mask and the immediate subsequent read of *ErrCntlStat0.Sta finds that there is an error pending, software may clear the status and follow up with another read. Since hardware must guarantee that the mask was in effect by the first read, no masked error should be left pending by the second read.

7.4.2.2 Recommended Flow for Enabling Errors at Boot

On UEFI compliant systems, it is recommended that UEFI must ensure that PER reporting and logging is disabled at UEFI Core Exit Boot Services. It is required that boot firmware puts the CCIX device into a known and stable state prior to transitioning to the OS or Hypervisor control. This will be necessary because the OS/Hypervisor will not be able to handle protocol errors until the CCIX PER driver has loaded and initialized PER capabilities of the device (per the appropriate platform RAS policy).

CCIX PER driver will take the following steps for each CCIX device discovered. The driver will first ensure that DevErrCntlStat.En is cleared. For each CCIX component on that device, it will perform the following sequence.

1 Fetch user RAS policy for each device component

2 If RAS policy exists, and indicates to disable PER logging

  a. Set *ErrCntlStat0.LogDis=1

  b. Move on to next CCIX component

3 If RAS policy exists, and indicates to disable PER message

  a. Set *ErrCntlStat0.Dis=1

4 If RAS policy exists, and indicates severity mask

  a. Set *ErrCntlStat1.SevReportMask, *ErrCntlStat1.SevLogMask, and *ErrCntlStat1.PerTypeMask based on device/component policy

After all components have been configured, set DevErrCntlStat.En=1.

7.4.2.3 Guidelines for Dynamic Error Mask Updates

It is recommended that CCIX protocol errors for a component are disabled during updates to the *ErrCntlStat1 mask fields. However, if dynamic mask updates are necessary while the component error logging is enabled (i.e., DevErrCntlStat.En=1, *ErrCntlStat0.LogDis=0), software is required to query the *ErrCntlStat0.Sta immediately after updating the mask.

If *ErrCntlStat0.Sta=1, software must check if the error severity or PER type is intended to be masked. If it is an error intended to be masked, software may clear the error and proceed assuming the mask has taken effect from this point on. Otherwise, software must handle the error that has been logged.

If *ErrCntlStat0.Sta=0, no action is needed and software may proceed assuming the mask has taken effect from this point on.

Chapter 8. CCIX ATS Specification

8.1 Introduction

Memory requests issued by a CCIX® Request Agent (RA) use physical addresses. Accelerator Functions (AFs) associated with an RA must present a Physical address to the RA. AFs handed a virtual address by software must perform address translation prior to using the address. Request Agents must not issue memory requests with untranslated virtual addresses onto a CCIX link as this may violate page table based security or virtualization which is hidden from the RA and its controlling software.

There are two methods an AF may use to translate a virtual address. The first is to obtain address translations from the host system using a variant of PCIe-defined Address Translation Services before performing translated memory requests. The second method for obtaining translated addresses is for the AF to contain an MMU that supports the native page table formats of the host system. The specific requirements for a native MMU are specific to the host platform and outside the scope of this specification.

8.2 Address Translation Services

An AF may use a variant of PCIe-defined ATS to obtain translated address and manage the state of translations stored in a local ATC. The ATC is logically located in the pipeline between the accelerator’s request generation logic and the CCIX request agent. The CCIX cache may only store data tagged using translated addresses.

PCIe formatted ATS translation requests, translation completions, invalidation requests, page requests and page responses are sent over VC0. Invalidation responses may be sent on any VC except for the CCIX VC.

CCIX compatible root ports and their associated Translation Agents must support PASID prefix in conjunction with PCIe ATS. CCIX Request Agents may optionally support PASID prefix.

8.3 Invalidation Semantics

The ATS invalidation completions semantics are modified when interacting with the CCIX VC. An Invalidation Completion from an Acceleration Function may not be returned until the following conditions are met:

- Normal PCIe ATS invalidation semantics are satisfied within the non-CCIX VCs.
- All translations derived from the invalidated address range have been removed from the Request Agent's ATC.
- An AF will not generate new requests using the invalidated translations.
- All outstanding cacheable requests using invalidated translations have been completed to the cache. For example, a cacheable write is completed after it is written into the cache. A cacheable read is completed after the target data is loaded into the cache.
- All outstanding device and non-cacheable requests using invalidated translations have been completed.

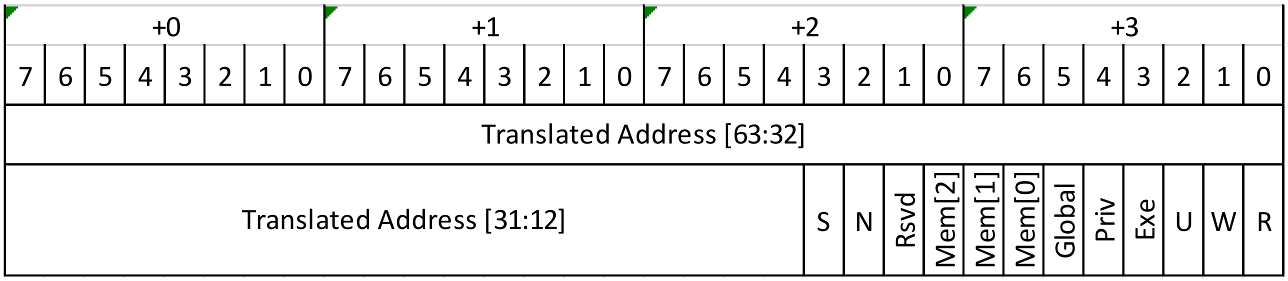
ATS invalidations do not cause data within the targeted Request Agent's cache to be invalidated or evicted. The cache may contain lines in any state associated with translated addresses that were derived from the previously valid translations.

8.4 Memory Type Information

In addition to obtaining translated page addresses, a CCIX Request Agent requires memory type information to properly interact with the memory at the translated address. The memory type information determines the types of operations that are supported on the CCIX link and whether data may be locally cached or not.

8.4.1 Memory Type

CCIX packets require the use of the ATS Memory Attributes (AMAs) returned in ATS Translation Completions. The format and definition of these ATS Translation Completions with AMA[2:0] labeled as Mem[2:0] is illustrated in Figure 8-1.



<table>
<tr><td colspan="8">+0</td><td colspan="8">+1</td><td colspan="8">+2</td><td colspan="8">+3</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Translated Address [63:32]</td></tr>
<tr><td colspan="20">Translated Address [31:12]</td><td>S</td><td>N</td><td>Rsvd</td><td>Mem[2]</td><td>Mem[1]</td><td>Mem[0]</td><td>Global</td><td>Priv</td><td>Exe</td><td>U</td><td>W</td><td>R</td></tr>
</table>

Figure 8-1: ATS Translation Completion with Memory Attributes Format

Table 8-1 illustrates the default mappings of the CCIX Memory Attributes, Mem[2:0] contained in the ATS Translation Completion AMA[2:0] field, illustrated in Figure 8-1.

Table 8-1: Memory Attributes

Mem[2:0]	Memory attribute
000	Device-Non-cacheable-No reordering-No Early write acknowledgement (Device-nRnE)
001	Device-Non-cacheable-No reordering-Early write acknowledgement permitted (Device-nRE)
010	Device-Non-cacheable-Reording permitted-Early write acknowledgement permitted (Device-RE)
011	Normal-Non-cacheable (Normal-NC)
100	Normal, Cacheable, Write-back, No allocate (Normal-WB-nRA-nWA)
101	Normal, Cacheable, Write-back, Read and Write Allocate (Normal-WB-RA-WA)
110	Normal, Cacheable, Write-back, No Read Allocate, Write Allocate (Normal-WB-nRA-WA)
111	Normal, Cacheable, Write-back, Read Allocate, No Write Allocate (Normal-WB-RA-nWA)

Acknowledgements

The CCIX® Consortium acknowledges the following authors of the specification:

Gord Caruk
Harb Abdulhamid
Hong Ahn
Jaideep Dastidar
Kiran Puranik
Millind Mittal
Phanindra Mannava

The CCIX® Consortium also acknowledges the following contributors to the specification:

Alan Wong
Ariel Almog
Ariel Shahar
Alexander Umansky
Bill Holland
Bruce Mathewson
Bryan Hornung
Carrie Cox
Charles Garcia-Tobin
Chris Bergen
Chris Borrelli
Chris Browy
Christopher Juenemann
Cyprian Wronka
David Herring
David Woolf
Dean Gonzales
Derek Rohde
Diego Crupnicoff
Dimitry Pavlovsky
Dong Wei
Dor Altshuler
Eddie Andrews
Eric Wehage
Fazil Osman
Frank Kavanagh
Fred Stivers
Haoqinfen
Gary Dick
Gabriele Paolini
Gaurav Singh
Geoff Zhang
Gerry Talbot
Gilad Shainer
Gustavo Pimentel
Jason Lawley
Jason Protchard
Jeff Defilippi
Jerrold Wheeler
Jim Panian
Joao Pinto
Joe Allen
Joe Breher
John Bainbridge
John Lofflink
John Moondanos
Jon Masters
Jonathan Cameron
John Stonick
Ken Chang
Kent Othner
Kangkang Shen
Kenneth Ma
Lana Chan
Lavi Koch
Leo Duran
Liutao
Liyongyao
Luis Rodriguez
Loren Jones
Makoto Ono
Manoj Roge
Mathieu Gagnon
Matt Evans
Maurice Steinman
Myron Slota
Nat Barbiero
Nathan Kalyanasundaram
Oded Lempel
Paul Cassidy
Philip Ng
Robbin Roger
Rick Eads
Robert Safranek
Sachin Dingra
Thanu Rangarajan
Tal Horowitz
Todd Farrell
Veronique Guerre
Vikram Sethi
Vilas Sridharan
Xin Chen
Yifan Huang
Ze'ev Rogachevsky
Zhujian