7

7. Software Initialization and Configuration

The PCI Express Configuration model supports two Configuration Space access mechanisms:

- PCI compatible Configuration Access Mechanism (CAM)
- PCI Express Enhanced Configuration Access Mechanism (ECAM)

The PCI compatible mechanism supports 100% binary compatibility with PCI 3.0 or later aware operating systems and their corresponding bus enumeration and configuration software.

The enhanced mechanism is provided to increase the size of available Configuration Space and to optimize access mechanisms.

7.1. Configuration Topology

To maintain compatibility with PCI software configuration mechanisms, all PCI Express elements have a PCI-compatible Configuration Space. Each PCI Express Link originates from a logical PCI-PCI Bridge and is mapped into Configuration Space as the secondary bus of this Bridge. The Root Port is a PCI-PCI Bridge structure that originates a PCI Express Link from a PCI Express Root Complex (see Figure 7-1).

A PCI Express Switch is represented by multiple PCI-PCI Bridge structures connecting PCI Express Links to an internal logical PCI bus (see Figure 7-2). The Switch Upstream Port is a PCI-PCI Bridge; the secondary bus of this Bridge represents the Switch's internal routing logic. Switch Downstream Ports are PCI-PCI Bridges bridging from the internal bus to buses representing the Downstream PCI Express Links from a PCI Express Switch. Only the PCI-PCI Bridges representing the Switch Downstream Ports may appear on the internal bus. Endpoints, represented by Type 0 Configuration Space headers, are not permitted to appear on the internal bus.

A PCI Express Endpoint is mapped into Configuration Space as a single Function in a Device, which might contain multiple Functions or just that Function. PCI Express Endpoints and Legacy Endpoints are required to appear within one of the Hierarchy Domains originated by the Root Complex, meaning that they appear in Configuration Space in a tree that has a Root Port as its head. Root Complex Integrated Endpoints and Root Complex Event Collectors do not appear within one of the Hierarchy Domains originated by the Root Complex. These appear in Configuration Space as peers of the Root Ports.

Unless otherwise specified, requirements in the Configuration Space definition for a device apply to single Function devices as well as to each Function individually of a multi-Function device.

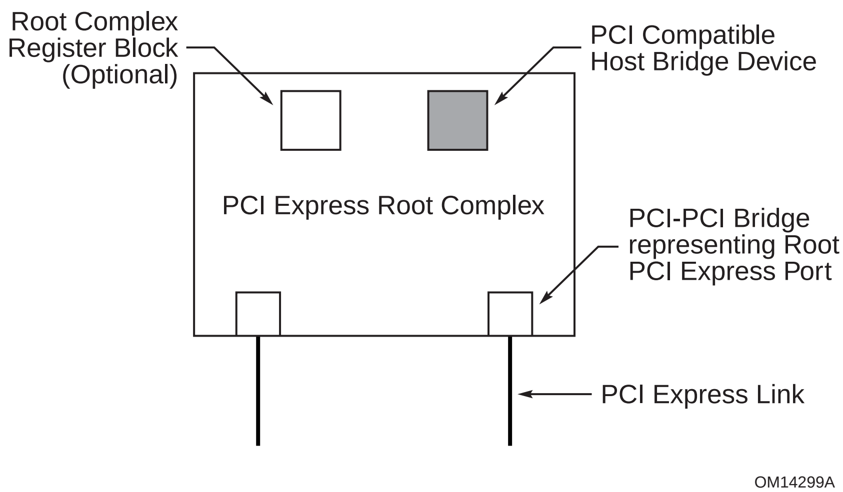

Figure 7-1: PCI Express Root Complex Device Mapping

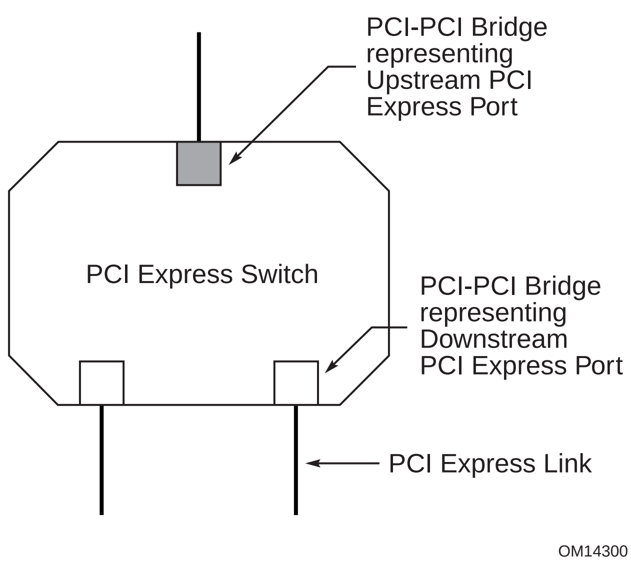

Figure 7-2: PCI Express Switch Device Mapping [91]

7.2. PCI Express Configuration Mechanisms

PCI Express extends the Configuration Space to 4096 bytes per Function as compared to 256 bytes allowed by *PCI Local Bus Specification, Revision 3.0*. PCI Express Configuration Space is divided into a PCI 3.0 compatible region, which consists of the first 256 bytes of a Function's Configuration Space, and a PCI Express Extended Configuration Space which consists of the remaining Configuration Space (see Figure 7-3). The PCI 3.0 compatible Configuration Space can be accessed using either the mechanism defined in the *PCI Local Bus Specification, Revision 3.0* or the PCI Express

[91] Future PCI Express Switches may be implemented as a single Switch device component (without the PCI-PCI Bridges) that is not limited by legacy compatibility requirements imposed by existing PCI software.

Enhanced Configuration Access Mechanism (ECAM) described later in this section. Accesses made using either access mechanism are equivalent. The PCI Express Extended Configuration Space can only be accessed by using the ECAM.[92]

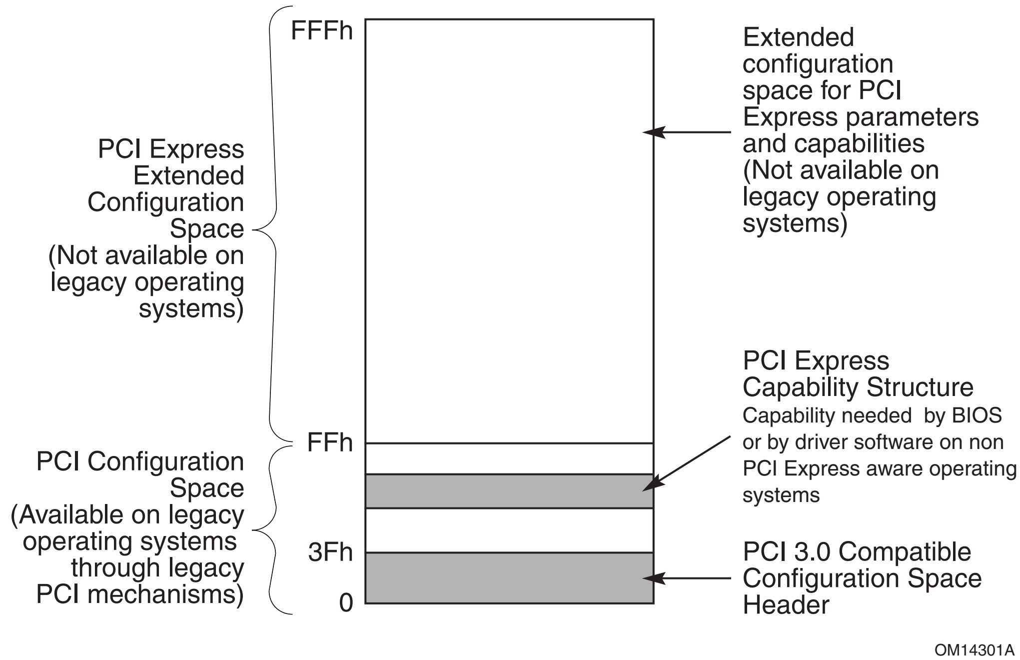

Figure 7-3: PCI Express Configuration Space Layout

7.2.1. PCI 3.0 Compatible Configuration Mechanism

The PCI 3.0 compatible PCI Express Configuration Mechanism supports the PCI Configuration Space programming model defined in the *PCI Local Bus Specification, Revision 3.0*. By adhering to this model, systems incorporating PCI Express interfaces remain compliant with conventional PCI bus enumeration and configuration software.

In the same manner as PCI 3.0 device Functions, PCI Express device Functions are required to provide a Configuration Space for software-driven initialization and configuration. Except for the differences described in this chapter, the PCI Express Configuration Space headers are organized to correspond with the format and behavior defined in the *PCI Local Bus Specification, Revision 3.0* (Section 6.1).

The PCI 3.0 compatible Configuration Access Mechanism uses the same Request format as the ECAM. For PCI compatible Configuration Requests, the Extended Register Address field must be all zeros.

[92] The ECAM operates independently from the mechanism defined in the *PCI Local Bus Specification, Revision 3.0* for generation of configuration transactions; there is no implied ordering between the two.

7.2.2. PCI Express Enhanced Configuration Access Mechanism (ECAM)

For systems that are PC-compatible, or that do not implement a processor-architecture-specific firmware interface standard that allows access to the Configuration Space, the ECAM is required as defined in this section.

For systems that implement a processor-architecture-specific firmware interface standard that allows access to the Configuration Space, the operating system uses the standard firmware interface, and the hardware access mechanism defined in this section is not required. For example, for systems that are compliant with *Developer's Interface Guide for 64-bit Intel Architecture-based Servers (DIG64), Version 2.1*,[93] the operating system uses the SAL firmware service to access the Configuration Space.

In all systems, device drivers are encouraged to use the application programming interface (API) provided by the operating system to access the Configuration Space of its device and not directly use the hardware mechanism.

The ECAM utilizes a flat memory-mapped address space to access device configuration registers. In this case, the memory address determines the configuration register accessed and the memory data updates (for a write) or returns the contents of (for a read) the addressed register. The mapping from memory address space to PCI Express Configuration Space address is defined in Table 7-1.

The size and base address for the range of memory addresses mapped to the Configuration Space are determined by the design of the host bridge and the firmware. They are reported by the firmware to the operating system in an implementation-specific manner. The size of the range is determined by the number of bits that the host bridge maps to the Bus Number field in the configuration address. In Table 7-1, this number of bits is represented as n, where $1 \le n \le 8$. A host bridge that maps n memory address bits to the Bus Number field supports Bus Numbers from 0 to $2^n - 1$, inclusive, and the base address of the range is aligned to a $2^{(n+20)}$-byte memory address boundary. Any bits in the Bus Number field that are not mapped from memory address bits must be Clear.

For example, if a system maps three memory address bits to the Bus Number field, the following are all true:

- $n = 3$.
- Address bits A[63:23] are used for the base address, which is aligned to a 2^{23}-byte (8-MB) boundary.
- Address bits A[22:20] are mapped to bits [2:0] in the Bus Number field.
- Bits [7:3] in the Bus Number field are set to Clear.
- The system is capable of addressing Bus Numbers between 0 and 7, inclusive.

A minimum of one memory address bit ($n = 1$) must be mapped to the Bus Number field. Systems are encouraged to map additional memory address bits to the Bus Number field as needed to support a larger number of buses. Systems that support more than 4 GB of memory addresses are

[93] *Developer's Interface Guide for 64-bit Intel Architecture-based Servers (DIG64), Version 2.1*, January 2002, www.dig64.org

encouraged to map eight bits of memory address ($n = 8$) to the Bus Number field. Note that in systems that include multiple host bridges with different ranges of Bus Numbers assigned to each host bridge, the highest Bus Number for the system is limited by the number of bits mapped by the host bridge to which the highest bus number is assigned. In such a system, the highest Bus Number assigned to a particular host bridge would be greater, in most cases, than the number of buses assigned to that host bridge. In other words, for each host bridge, the number of bits mapped to the Bus Number field, n, must be large enough that the highest Bus Number assigned to each particular bridge must be less than or equal to $2^n - 1$ for that bridge.

In some processor architectures, it is possible to generate memory accesses that cannot be expressed in a single Configuration Request, for example due to crossing a DW aligned boundary, or because a locked access is used. A Root Complex implementation is not required to support the translation to Configuration Requests of such accesses.

Table 7-1: Enhanced Configuration Address Mapping

Memory Address[94]	PCI Express Configuration Space
A[(20 + n – 1):20]	Bus Number $1 \le n \le 8$
A[19:15]	Device Number
A[14:12]	Function Number
A[11:8]	Extended Register Number
A[7:2]	Register Number
A[1:0]	Along with size of the access, used to generate Byte Enables

Note: for Requests targeting Extended Functions in an ARI Device, A[19:12] represents the (8-bit) Function Number, which replaces the (5-bit) Device Number and (3-bit) Function Number fields above.

The system hardware must provide a method for the system software to guarantee that a write transaction using the ECAM is completed by the completer before system software execution continues.

[94] This address refers to the byte-level address from a software point of view.

IMPLEMENTATION NOTE

Ordering Considerations for the Enhanced Configuration Access Mechanism

The ECAM converts memory transactions from the host CPU into Configuration Requests on the PCI Express fabric. This conversion potentially creates ordering problems for the software, because writes to memory addresses are typically posted transactions but writes to Configuration Space are not posted on the PCI Express fabric.

Generally, software does not know when a posted transaction is completed by the completer. In those cases in which the software must know that a posted transaction is completed by the completer, one technique commonly used by the software is to read the location that was just written. For systems that follow the PCI ordering rules throughout, the read transaction will not complete until the posted write is complete. However, since the PCI ordering rules allow non-posted write and read transactions to be reordered with respect to each other, the CPU must wait for a non-posted write to complete on the PCI Express fabric to be guaranteed that the transaction is completed by the completer.

As an example, software may wish to configure a device Function's Base Address register by writing to the device using the ECAM, and then read a location in the memory-mapped range described by this Base Address register. If the software were to issue the memory-mapped read before the ECAM write was completed, it would be possible for the memory-mapped read to be re-ordered and arrive at the device before the Configuration Write Request, thus causing unpredictable results.

To avoid this problem, processor and host bridge implementations must ensure that a method exists for the software to determine when the write using the ECAM is completed by the completer.

This method may simply be that the processor itself recognizes a memory range dedicated for mapping ECAM accesses as unique, and treats accesses to this range in the same manner that it would treat other accesses that generate non-posted writes on the PCI Express fabric, i.e., that the transaction is not posted from the processor's viewpoint. An alternative mechanism is for the host bridge (rather than the processor) to recognize the memory-mapped Configuration Space accesses and not to indicate to the processor that this write has been accepted until the non-posted Configuration Transaction has completed on the PCI Express fabric. A third alternative would be for the processor and host bridge to post the memory-mapped write to the ECAM and for the host bridge to provide a separate register that the software can read to determine when the Configuration Write Request has completed on the PCI Express fabric. Other alternatives are also possible.

IMPLEMENTATION NOTE

Generating Configuration Requests

Because Root Complex implementations are not required to support the generation of Configuration Requests from accesses that cross DW boundaries, or that use locked semantics, software should take care not to cause the generation of such accesses when using the memory-mapped ECAM unless it is known that the Root Complex implementation being used will support the translation.

7.2.2.1. Host Bridge Requirements

For those systems that implement the ECAM, the PCI Express Host Bridge is required to translate the memory-mapped PCI Express Configuration Space accesses from the host processor to PCI Express configuration transactions. The use of Host Bridge PCI class code is reserved for backwards compatibility; host Bridge Configuration Space is opaque to standard PCI Express software and may be implemented in an implementation specific manner that is compatible with PCI Host Bridge Type 0 Configuration Space. A PCI Express Host Bridge is not required to signal errors through a Root Complex Event Collector. This support is optional for PCI Express Host Bridges.

7.2.2.2. PCI Express Device Requirements

Devices must support an additional 4 bits for decoding configuration register access, i.e., they must decode the Extended Register Address[3:0] field of the Configuration Request header.

IMPLEMENTATION NOTE

Device-Specific Registers in Configuration Space

It is strongly recommended that PCI Express devices place no registers in Configuration Space other than those in headers or Capability structures architected by applicable PCI specifications.

Device-specific registers that have legitimate reasons to be placed in Configuration Space (e.g., they need to be accessible before Memory Space is allocated) should be placed in a Vendor-Specific Capability structure (in PCI Compatible Configuration Space) or a Vendor-Specific Extended Capability structure (in PCI Express Extended Configuration Space).

Device-specific registers accessed in the run-time environment by drivers should be placed in Memory Space that is allocated by one or more Base Address registers. Even though PCI Compatible or PCI Express Extended Configuration Space may have adequate room for run-time device-specific registers, placing them there is highly discouraged for the following reasons:

- Not all Operating Systems permit drivers to access Configuration Space directly.
- Some platforms provide access to Configuration Space only via firmware calls, which typically have substantially lower performance compared to mechanisms for accessing Memory Space.
- Even on platforms that provide direct access to a memory-mapped PCI Express Enhanced Configuration Mechanism, performance for accessing Configuration Space will typically be significantly lower than for accessing Memory Space since:
    - Configuration Reads and Writes must usually be DWORD or smaller in size,
    - Configuration Writes are usually not posted by the platform, and
    - Some platforms support only one outstanding Configuration Write at a time.

7.2.3. Root Complex Register Block

A Root Port or Root Complex Integrated Endpoint may be associated with an optional 4096-byte block of memory mapped registers referred to as the Root Complex Register Block (RCRB). These registers are used in a manner similar to Configuration Space and can include PCI Express Extended Capabilities and other implementation specific registers that apply to the Root Complex. The structure of the RCRB is described in Section 7.9.2.

Multiple Root Ports or internal devices are permitted to be associated with the same RCRB. The RCRB memory-mapped registers must not reside in the same address space as the memory-mapped Configuration Space or Memory Space.

A Root Complex implementation is not required to support accesses to an RCRB that cross DWORD aligned boundaries or accesses that use locked semantics.

IMPLEMENTATION NOTE

Accessing Root Complex Register Block

Because Root Complex implementations are not required to support accesses to a RCRB that cross DW boundaries, or that use locked semantics, software should take care not to cause the generation of such accesses when accessing a RCRB unless the Root Complex will support the access.

7.3. Configuration Transaction Rules

7.3.1. Device Number

With non-ARI Devices, PCI Express components are restricted to implementing a single Device Number on their primary interface (Upstream Port), but are permitted to implement up to eight independent Functions within that Device Number. Each internal Function is selected based on decoded address information that is provided as part of the address portion of Configuration Request packets.

Downstream Ports that do not have ARI Forwarding enabled must associate only Device 0 with the device attached to the Logical Bus representing the Link from the Port. Configuration Requests targeting the Bus Number associated with a Link specifying Device Number 0 are delivered to the device attached to the Link; Configuration Requests specifying all other Device Numbers (1-31) must be terminated by the Switch Downstream Port or the Root Port with an Unsupported Request Completion Status (equivalent to Master Abort in PCI). Non-ARI Devices must not assume that Device Number 0 is associated with their Upstream Port, but must capture their assigned Device Number as discussed in Section 2.2.6.2. Non-ARI Devices must respond to all Type 0 Configuration Read Requests, regardless of the Device Number specified in the Request.

Switches, and components wishing to incorporate more than eight Functions at their Upstream Port, are permitted to implement one or more "virtual switches" represented by multiple Type 1 (PCI-PCI Bridge) Configuration Space headers as illustrated in Figure 7-2. These virtual switches serve to allow fan-out beyond eight Functions. Since Switch Downstream Ports are permitted to appear on any Device Number, in this case all address information fields (Bus, Device, and Function Numbers) must be completely decoded to access the correct register. Any Configuration Request targeting an unimplemented Bus, Device, or Function must return a Completion with Unsupported Request Completion Status.

With an ARI Device, its Device Number is implied to be 0 rather than specified by a field within an ID. The traditional 5-bit Device Number and 3-bit Function Number fields in its associated Routing IDs, Requester IDs, and Completer IDs are interpreted as a single 8-bit Function Number. See Section 6.13. Any Type 0 Configuration Request targeting an unimplemented Function in an ARI Device must be handled as an Unsupported Request.

If an ARI Downstream Port has ARI Forwarding enabled, the logic that determines when to turn a Type 1 Configuration Request into a Type 0 Configuration Request no longer enforces a restriction on the traditional Device Number field being 0.

The following section provides details of the Configuration Space addressing mechanism.

7.3.2. Configuration Transaction Addressing

PCI Express Configuration Requests use the following addressing fields:

- Bus Number – PCI Express maps logical PCI Bus Numbers onto PCI Express Links such that PCI 3.0 compatible configuration software views the Configuration Space of a PCI Express Hierarchy as a PCI hierarchy including multiple bus segments.
- Device Number – Device Number association is discussed in Section 7.3.1. When an ARI Device is targeted and the Downstream Port immediately above it is enabled for ARI Forwarding, the Device Number is implied to be 0, and the traditional Device Number field is used instead as part of an 8-bit Function Number field. See Section 6.13.
- Function Number – PCI Express also supports multi-Function devices using the same discovery mechanism as PCI 3.0. With ARI Devices, discovery and enumeration of Extended Functions require ARI-aware software. See Section 6.13.
- Extended Register Number and Register Number – Specify the Configuration Space address of the register being accessed (concatenated such that the Extended Register Number forms the more significant bits).

7.3.3. Configuration Request Routing Rules

For Endpoints, the following rules apply:

- If Configuration Request Type is 1,
  - Follow the rules for handling Unsupported Requests
- If Configuration Request Type is 0,
  - Determine if the Request addresses a valid local Configuration Space of an implemented Function
    - If so, process the Request
    - If not, follow rules for handling Unsupported Requests

For Root Ports, Switches, and PCI Express-PCI Bridges, the following rules apply:

- Propagation of Configuration Requests from Downstream to Upstream as well as peer-to-peer are not supported
  - Configuration Requests are initiated only by the Host Bridge
- If Configuration Request Type is 0,
  - Determine if the Request addresses a valid local Configuration Space of an implemented Function
    - If so, process the Request
    - If not, follow rules for handling Unsupported Requests

- If Configuration Request Type is 1, apply the following tests, in sequence, to the Bus Number and Device Number fields:
    - If in the case of a PCI Express-PCI Bridge, equal to the Bus Number assigned to secondary PCI bus or, in the case of a Switch or Root Complex, equal to the Bus Number and decoded Device Numbers assigned to one of the Root (Root Complex) or Downstream Ports (Switch),
        - Transform the Request to Type 0 by changing the value in the Type[4:0] field of the Request (see Table 2-3) – all other fields of the Request remain unchanged
        - Forward the Request to that Downstream Port (or PCI bus, in the case of a PCI Express-PCI Bridge)
    - If not equal to the Bus Number of any of Downstream Ports or secondary PCI bus, but in the range of Bus Numbers assigned to either a Downstream Port or a secondary PCI bus,
        - Forward the Request to that Downstream Port interface without modification
    - Else (none of the above)
        - The Request is invalid – follow the rules for handling Unsupported Requests
- PCI Express-PCI Bridges must terminate as Unsupported Requests any Configuration Requests for which the Extended Register Address field is non-zero that are directed towards a PCI bus that does not support Extended Configuration Space.

  Note: This type of access is a consequence of a programming error.

Additional rule specific to Root Complexes:

- Configuration Requests addressing Bus Numbers assigned to devices within the Root Complex are processed by the Root Complex
    - The assignment of Bus Numbers to the devices within a Root Complex may be done in an implementation specific way.

For all types of devices:

All other Configuration Space addressing fields are decoded according to the *PCI Local Bus Specification, Revision 3.0*.

7.3.4. PCI Special Cycles

PCI Special Cycles (see the *PCI Local Bus Specification, Revision 3.0* for details) are not directly supported by PCI Express. PCI Special Cycles may be directed to PCI bus segments behind PCI Express-PCI Bridges using Type 1 Configuration Cycles as described in *PCI Local Bus Specification, Revision 3.0*.

7.4. Configuration Register Types

Configuration register fields are assigned one of the attributes described in Table 7-2. All PCI Express components, with the exception of the Root Complex and system-integrated devices, initialize register fields to specified default values. Root Complexes and system-integrated devices initialize register fields as required by the firmware for a particular system implementation.

Table 7-2: Register and Register Bit-Field Types

Register Attribute	Description
HwInit	**Hardware Initialized** – Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. (System firmware hardware initialization is only allowed for system integrated devices.) Bits are read-only after initialization and can only be reset (for write-once by firmware) with Fundamental Reset (see Section 6.6.1). HwInit register bits are not modified by an FLR.
RO	**Read-only** – Register bits are read-only and cannot be altered by software. Register bits are permitted to be initialized by hardware mechanisms such as pin strapping or serial EEPROM.
RW	**Read-Write** – Register bits are read-write and are permitted to be either Set or Cleared by software to the desired state.
RW1C	**Write-1-to-clear status** – Register bits indicate status when read. A Set bit indicates a status event which is Cleared by writing a 1b. Writing a 0b to RW1C bits has no effect.
ROS	**Sticky - Read-only** – Register bits are read-only and cannot be altered by software. Bits are neither initialized nor modified by hot reset or FLR. Where noted, devices that consume AUX power must preserve sticky register values when AUX power consumption (via either AUX power or PME Enable) is enabled. In these cases, registers are neither initialized nor modified by hot, warm, or cold reset (see Section 6.6).
RWS	**Sticky - Read-Write** – Register bits are read-write and are Set or Cleared by software to the desired state. Bits are neither initialized nor modified by hot reset or FLR. Where noted, devices that consume AUX power must preserve sticky register values when AUX power consumption (via either AUX power or PME Enable) is enabled. In these cases, registers are neither initialized nor modified by hot, warm, or cold reset (see Section 6.6).

Register Attribute	Description
RW1CS	**Sticky - Write-1-to-clear status** – Registers indicate status when read. A Set bit indicates a status event which is Cleared by writing a 1b. Writing a 0b to RW1CS bits has no effect. Bits are neither initialized nor modified by hot reset or FLR. Where noted, devices that consume AUX power must preserve sticky register values when AUX power consumption (via either AUX power or PME Enable) is enabled. In these cases, registers are neither initialized nor modified by hot, warm, or cold reset (see Section 6.6).
RsvdP	**Reserved and Preserved** – Reserved for future RW implementations. Registers are read-only and must return zero when read. Software must preserve the value read for writes to bits.
RsvdZ	**Reserved and Zero** – Reserved for future RW1C implementations. Registers are read-only and must return zero when read. Software must use 0b for writes to bits.

7.5. PCI-Compatible Configuration Registers

The first 256 bytes of the PCI Express Configuration Space form the PCI 3.0 compatibility region. This region completely aliases the PCI 3.0 Configuration Space of the Function. Legacy PCI devices can also be accessed with the ECAM without requiring any modifications to the device hardware or device driver software. This section establishes the mapping between PCI 3.0 and PCI Express for format and behavior of PCI 3.0 compatible registers.

All registers and fields not described in this section have the same definition as in the *PCI Local Bus Specification, Revision 3.0*. Layout of the Configuration Space and format of individual configuration registers are depicted following the little-endian convention used in the *PCI Local Bus Specification, Revision 3.0*.

7.5.1. Type 0/1 Common Configuration Space

Figure 7-4 details allocation for common register fields of PCI 3.0 Type 0 and Type 1 Configuration Space headers for PCI Express device Functions.

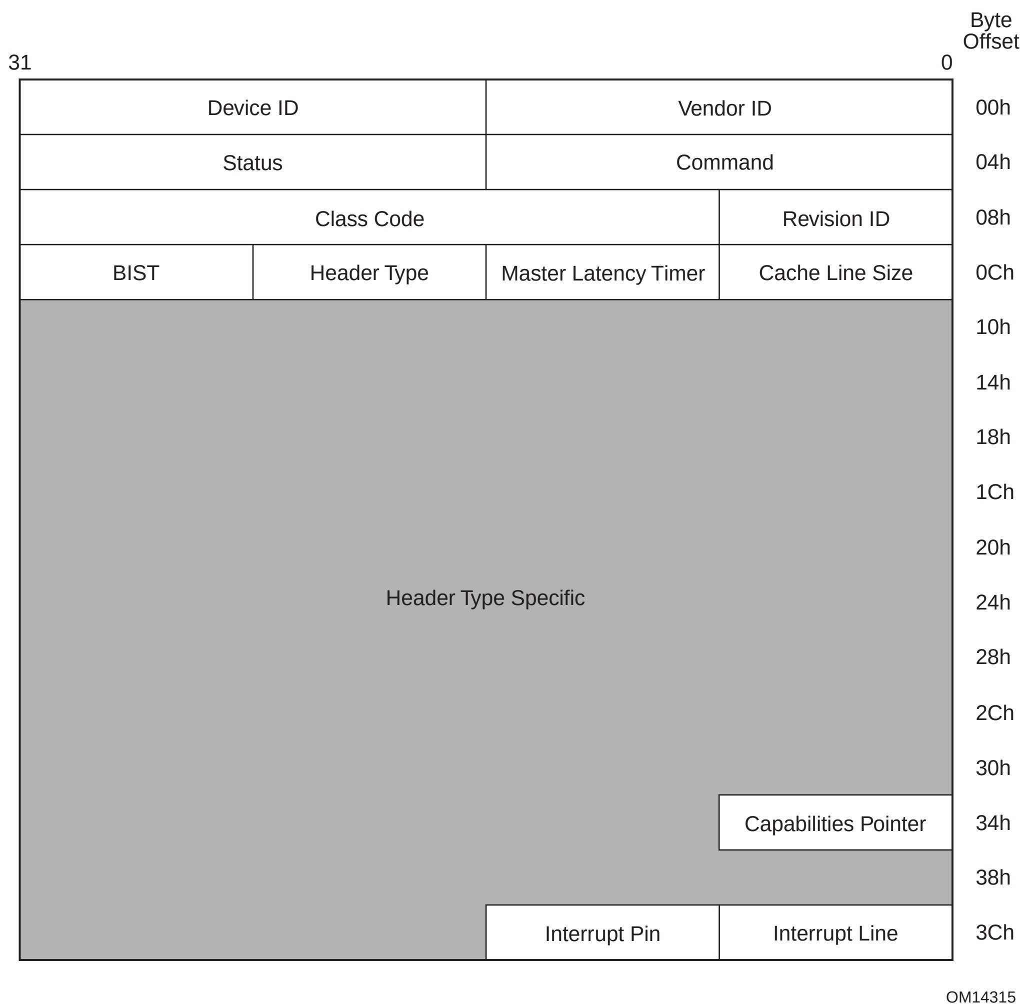

Figure 7-4: Common Configuration Space Header

These registers are defined for both Type 0 and Type 1 Configuration Space headers. The PCI Express-specific interpretation of these registers is defined in this section.

7.5.1.1. Command Register (Offset 04h)

Table 7-3 establishes the mapping between PCI 3.0 and PCI Express for the PCI 3.0 Configuration Space Command register.

Table 7-3: Command Register

Bit Location	Register Description	Attributes
2	**Bus Master Enable** – Controls the ability of a PCI Express Endpoint to issue Memory[95] and I/O Read/Write Requests, and the ability of a Root or Switch Port to forward Memory and I/O Read/Write Requests in the Upstream direction *Endpoints:* When this bit is Set, the PCI Express Function is allowed to issue Memory or I/O Requests. When this bit is Clear, the PCI Express Function is not allowed to issue any Memory or I/O Requests. Note that as MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit to 0b disables MSI/MSI-X interrupt Messages as well. Requests other than Memory or I/O Requests are not controlled by this bit. Default value of this bit is 0b. This bit is hardwired to 0b if a Function does not generate Memory or I/O Requests. *Root and Switch Ports:* This bit controls forwarding of Memory or I/O Requests by a Switch or Root Port in the Upstream direction. When this bit is 0b, Memory and I/O Requests received at a Root Port or the Downstream side of a Switch Port must be handled as Unsupported Requests (UR), and for Non-Posted Requests a Completion with UR completion status must be returned. This bit does not affect forwarding of Completions in either the Upstream or Downstream direction. The forwarding of Requests other than Memory or I/O Requests is not controlled by this bit. Default value of this bit is 0b.	RW
3	**Special Cycle Enable** – Does not apply to PCI Express and must be hardwired to 0b.	RO
4	**Memory Write and Invalidate** – Does not apply to PCI Express and must be hardwired to 0b.	RO

[95] The AtomicOp Requester Enable bit in the Device Control 2 register must also be Set in order for an AtomicOp Requester to initiate AtomicOp Requests, which are Memory Requests.

Bit Location	Register Description	Attributes
5	**VGA Palette Snoop** – Does not apply to PCI Express and must be hardwired to 0b.	RO
6	**Parity Error Response** – See Section 7.5.1.7. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Status register. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
7	**IDSEL Stepping/Wait Cycle Control** – Does not apply to PCI Express and must be hardwired to 0b.	RO
8	**SERR# Enable** – See Section 7.5.1.7. When Set, this bit enables reporting of Non-fatal and Fatal errors detected by the Function to the Root Complex. Note that errors are reported if enabled either through this bit or through the PCI Express specific bits in the Device Control register (see Section 7.8.4). In addition, for Functions with Type 1 Configuration Space headers, this bit controls transmission by the primary interface of ERR_NONFATAL and ERR_FATAL error Messages forwarded from the secondary interface. This bit does not affect the transmission of forwarded ERR_COR messages. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
9	**Fast Back-to-Back Transactions Enable** – Does not apply to PCI Express and must be hardwired to 0b.	RO

Bit Location	Register Description	Attributes
10	**Interrupt Disable** – Controls the ability of a PCI Express Function to generate INTx interrupts. When Set, Functions are prevented from asserting INTx interrupts. Any INTx emulation interrupts already asserted by the Function must be deasserted when this bit is Set. As described in Section 2.2.8.1, INTx interrupts use virtual wires that must, if asserted, be deasserted using the appropriate Deassert_INTx message(s) when this bit is Set. Only the INTx virtual wire interrupt(s) associated with the Function(s) for which this bit is Set are affected. For Endpoints that generate INTx interrupts, this bit is required. For Endpoints that do not generate INTx interrupts this bit is optional. If not implemented, this bit must be hardwired to 0b. For Root Ports, Switch Ports, and Bridges that generate INTx interrupts on their own behalf, this bit is required. This bit has no effect on interrupts that pass through the Port from the secondary side. For Root Ports, Switch Ports, and Bridges that do not generate INTx interrupts on their own behalf this bit is optional. If not implemented, this bit must be hardwired to 0b. Default value of this bit is 0b.	RW

7.5.1.2. Status Register (Offset 06h)

Table 7-4 establishes the mapping between PCI 3.0 and PCI Express for PCI 3.0 Configuration Space Status register.

Table 7-4: Status Register

Bit Location	Register Description	Attributes
3	**Interrupt Status** – When Set, indicates that an INTx emulation interrupt is pending internally in the Function. Note that INTx emulation interrupts forwarded by Root and Switch Ports from devices Downstream of the Root or Switch Port are not reflected in this bit. Default value of this bit is 0b.	RO
4	**Capabilities List** – Indicates the presence of an Extended Capability list item. Since all PCI Express device Functions are required to implement the PCI Express Capability structure, this bit must be hardwired to 1b.	RO
5	**66 MHz Capable** – Does not apply to PCI Express and must be hardwired to 0b.	RO
7	**Fast Back-to-Back Transactions Capable** – Does not apply to PCI Express and must be hardwired to 0b.	RO

Bit Location	Register Description	Attributes
8	**Master Data Parity Error** – See Section 7.5.1.7. This bit is Set by an Endpoint Function if the Parity Error Response bit in the Command register is 1b and either of the following two conditions occurs: • Endpoint receives a Poisoned Completion • Endpoint transmits a Poisioned Request This bit is Set by a Root Port, Switch Upstream Port, or Switch Downstream Port if the Parity Error Response bit in the Command register is 1b and either of the following two conditions occurs: • Port receives a Poisoned Completion going Downstream • Port transmits a Poisoned Request Upstream If the Parity Error Response bit is 0b, this bit is never Set. Default value of this bit is 0b.	RW1C
10:9	**DEVSEL Timing** – Does not apply to PCI Express and must be hardwired to 00b.	RO
11	**Signaled Target Abort** – See Section 7.5.1.7. This bit is Set when a Function completes a Posted or Non-Posted Request as a Completer Abort error. This applies to a Function with a Type 1 Configuration header when the Completer Abort was generated by its Primary Side. Default value of this bit is 0b.	RW1C
12	**Received Target Abort** – See Section 7.5.1.7. This bit is Set when a Requester receives a Completion with Completer Abort Completion Status. On a Function with a Type 1 Configuration header, the bit is Set when the Completer Abort is received by its Primary Side. Default value of this bit is 0b.	RW1C
13	**Received Master Abort** – See Section 7.5.1.7. This bit is Set when a Requester receives a Completion with Unsupported Request Completion Status. On a Function with a Type 1 Configuration header, the bit is Set when the Unsupported Request is received by its Primary Side. Default value of this bit is 0b.	RW1C
14	**Signaled System Error** – See Section 7.5.1.7. This bit is Set when a Function sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR# Enable bit in the Command register is 1. Default value of this bit is 0b.	RW1C

Bit Location	Register Description	Attributes
15	**Detected Parity Error** – See Section 7.5.1.7. This bit is Set by a Function whenever it receives a Poisoned TLP, regardless of the state the Parity Error Response bit in the Command register. On a Function with a Type 1 Configuration header, the bit is Set when the Poisoned TLP is received by its Primary Side. Default value of this bit is 0b.	RW1C

7.5.1.3. Cache Line Size Register (Offset 0Ch)

The Cache Line Size register is set by the system firmware or the operating system to system cache line size. However, note that legacy PCI 3.0 software may not always be able to program this field correctly especially in the case of Hot-Plug devices. This field is implemented by PCI Express devices as a read-write field for legacy compatibility purposes but has no effect on any PCI Express device behavior.

7.5.1.4. Latency Timer Register (Offset 0Dh)

This register is also referred to as Primary Latency Timer for Type 1 Configuration Space header Functions. The Latency Timer does not apply to PCI Express. This register must be hardwired to 00h.

7.5.1.5. Interrupt Line Register (Offset 3Ch)

As in PCI 3.0, the Interrupt Line register communicates interrupt line routing information. The register is read/write and must be implemented by any Function that uses an interrupt pin (see following description). Values in this register are programmed by system software and are system architecture specific. The Function itself does not use this value; rather the value in this register is used by device drivers and operating systems.

7.5.1.6. Interrupt Pin Register (Offset 3Dh)

The Interrupt Pin register is a read-only register that identifies the legacy interrupt Message(s) the Function uses (see Section 6.1 for further details). Valid values are 1, 2, 3, and 4 that map to legacy interrupt Messages for INTA, INTB, INTC, and INTD respectively. A value of 00h indicates that the Function uses no legacy interrupt Message(s).

7.5.1.7. Error Registers

The Error Control/Status register bits in the Command and Status registers (see Section 7.5.1.1 and Section 7.5.1.2 respectively) and the Bridge Control and Secondary Status registers of Type 1 Configuration Space header Functions (see Section 7.5.3.6 and Section 7.5.3.4 respectively) control PCI-compatible error reporting for both PCI and PCI Express device Functions. Mapping of PCI

Express errors onto PCI errors is also discussed in Section 6.2.7.1. In addition to the PCI-compatible error control and status, PCI Express error reporting may be controlled separately from PCI device Functions through the PCI Express Capability structure described in Section 7.8. The PCI-compatible Error Control and Status register fields do not have any effect on PCI Express error reporting enabled through the PCI Express Capability structure. PCI Express device Functions may implement optional advanced error reporting as described in Section 7.10.

For PCI Express Root Ports represented by a PCI 3.0 Type 1 Configuration Space header:

- The primary side Error Control/Status registers apply to errors detected on the internal logic associated with the Root Complex.
- The secondary side Error Control/Status registers apply to errors detected on the Link originating from the Root Port.

For PCI Express Switch Upstream Ports represented by a PCI 3.0 Type 1 Configuration Space header:

- The primary side Error Control/Status registers apply to errors detected on the Upstream Link of the Switch.
- The secondary side Error Control/Status registers apply to errors detected on the internal logic of the Switch.

For PCI Express Switch Downstream Ports represented by a PCI 3.0 Type 1 Configuration Space header:

- The primary side Error Control/Status registers apply to errors detected on the internal logic of the Switch.
- The secondary side Error Control/Status registers apply to errors detected on the Downstream Link originating from the Switch Port.

7.5.2. Type 0 Configuration Space Header

Figure 7-5 details allocation for register fields of PCI 3.0 Type 0 Configuration Space header for PCI Express device Functions.

31			0	Byte Offset
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST	Header Type	Master Latency Timer	Cache Line Size	0Ch
Base Address Registers				10h
				14h
				18h
				1Ch
				20h
				24h
Cardbus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved			Capabilities Pointer	34h
Reserved				38h
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	3Ch

OM14316

Figure 7-5: Type 0 Configuration Space Header

Section 7.5.1 details the PCI Express-specific registers that are valid for all Configuration Space header types. The PCI Express-specific interpretation of registers specific to Type 0 PCI 3.0 Configuration Space header is defined in this section.

7.5.2.1. Base Address Registers (Offset 10h - 24h)

A PCI Express Function requesting Memory Space through a BAR must set the BAR's Prefetchable bit unless the range contains locations with read side effects or locations in which the Function does not tolerate write merging. It is strongly encouraged that resources mapped into Memory Space be designed as prefetchable whenever possible. PCI Express Functions other than Legacy Endpoints must support 64-bit addressing for any Base Address register that requests prefetchable Memory Space. The minimum Memory Space address range requested by a BAR is 128 bytes. The attributes for some of the bits in the BAR are affected by the Resizable BAR Capability, if it is implemented.

7.5.2.2. Min_Gnt/Max_Lat Registers (Offset 3Eh/3Fh)

These registers do not apply to PCI Express. They must be read-only and hardwired to 00h.

7.5.3. Type 1 Configuration Space Header

Figure 7-6 details allocation for register fields of PCI 3.0 Type 1 Configuration Space header for Switch and Root Complex virtual PCI Bridges.

31			0	Byte Offset
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST	Header Type	Primary Latency Timer	Cache Line Size	0Ch
Base Address Register 0				10h
Base Address Register 1				14h
Secondary Latency Timer	Subordinate Bus Number	Secondary Bus Number	Primary Bus Number	18h
Secondary Status		I/O Limit	I/O Base	1Ch
Memory Limit		Memory Base		20h
Prefetchable Memory Limit		Prefetchable Memory Base		24h
Prefetchable Base Upper 32 Bits				28h
Prefetchable Limit Upper 32 Bits				2Ch
I/O Limit Upper 16 Bits		I/O Base Upper 16 Bits		30h
Reserved			Capability Pointer	34h
Expansion ROM Base Address				38h
Bridge Control		Interrupt Pin	Interrupt Line	3Ch

OM14317

Figure 7-6: Type 1 Configuration Space Header

Section 7.5.1 details the PCI Express-specific registers that are valid for all Configuration Space header types. The PCI Express-specific interpretation of registers specific to Type 1 PCI 3.0 Configuration Space header is defined in this section. Register interpretations described in this section apply to PCI-PCI Bridge structures representing Switch and Root Ports; other device

Functions such as PCI Express to PCI/PCI-X Bridges with Type 1 PCI 3.0 Configuration Space headers are not covered by this section.

7.5.3.1. Base Address Registers (Offset 10h/14h)

A PCI Express Function requesting memory resources through a BAR must set the BAR's Prefetchable bit unless the range contains locations with read side effects or locations in which the Function does not tolerate write merging. It is strongly encouraged that memory-mapped resources be designed as prefetchable whenever possible. PCI Express device Functions other than Legacy Endpoints must support 64-bit addressing for any Base Address register that requests prefetchable memory resources. The minimum memory address range requested by a BAR is 128 bytes.

7.5.3.2. Primary Bus Number (Offset 18h)

Except as noted, this register is not used by PCI Express Functions but must be implemented as read-write for compatibility with legacy software. PCI Express Functions capture the Bus (and Device) Number as described in Section 2.2.6. Refer to *PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0* for exceptions to this requirement.

7.5.3.3. Secondary Latency Timer (Offset 1Bh)

This register does not apply to PCI Express. It must be read-only and hardwired to 00h.

7.5.3.4. Secondary Status Register (Offset 1Eh)

Table 7-5 establishes the mapping between PCI 3.0 and PCI Express for PCI 3.0 Configuration Space Secondary Status register.

Table 7-5: Secondary Status Register

Bit Location	Register Description	Attributes
5	**66 MHz Capable** – Does not apply to PCI Express and must be hardwired to 0b.	RO
7	**Fast Back-to-Back Transactions Capable** – Does not apply to PCI Express and must be hardwired to 0b.	RO
8	**Master Data Parity Error** – See Section 7.5.1.7. This bit is Set by a Root Port, Switch Upstream Port, or Switch Downstream Port if the Parity Error Response Enable bit in the Bridge Control register is Set and either of the following two conditions occurs: ● Port receives a Poisoned Completion coming Upstream ● Port transmits a Poisoned Request Downstream If the Parity Error Response Enable bit is Clear, this bit is never Set. Default value of this bit is 0b.	RW1C

Bit Location	Register Description	Attributes
10:9	**DEVSEL Timing** – Does not apply to PCI Express and must be hardwired to 00b.	RO
11	**Signaled Target Abort** – See Section 7.5.1.7. This bit is Set when the Secondary Side for Type 1 Configuration Space header Function (for Requests completed by the Type 1 header Function itself) completes a Posted or Non-Posted Request as a Completer Abort error. Default value of this bit is 0b.	RW1C
12	**Received Target Abort** – See Section 7.5.1.7. This bit is Set when the Secondary Side for Type 1 Configuration Space header Function (for Requests initiated by the Type 1 header Function itself) receives a Completion with Completer Abort Completion Status. Default value of this bit is 0b.	RW1C
13	**Received Master Abort** – See Section 7.5.1.7. This bit is Set when the Secondary Side for Type 1 Configuration Space header Function (for Requests initiated by the Type 1 header Function itself) receives a Completion with Unsupported Request Completion Status. Default value of this bit is 0b.	RW1C
14	**Received System Error** – See Section 7.5.1.7. This bit is Set when the Secondary Side for a Type 1 Configuration Space header Function receives an ERR_FATAL or ERR_NONFATAL Message. Default value of this bit is 0b.	RW1C
15	**Detected Parity Error** – See Section 7.5.1.7. This bit is Set by the Secondary Side for a Type 1 Configuration Space header Function whenever it receives a Poisoned TLP, regardless of the state the Parity Error Response Enable bit in the Bridge Control register. Default value of this bit is 0b.	RW1C

7.5.3.5. Prefetchable Memory Base/Limit (Offset 24h)

The Prefetchable Memory Base and Prefetchable Memory Limit registers must indicate that 64-bit addresses are supported, as defined in *PCI-to-PCI Bridge Architecture Specification, Revision 1.2*.

7.5.3.6. Bridge Control Register (Offset 3Eh)

Table 7-6 establishes the mapping between PCI 3.0 and PCI Express for the PCI 3.0 Configuration Space Bridge Control register.

Table 7-6: Bridge Control Register

Bit Location	Register Description	Attributes
0	**Parity Error Response Enable** – See Section 7.5.1.7. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register. Default value of this bit is 0b.	RW
1	**SERR# Enable** – See Section 7.5.1.7. This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL from secondary to primary. Default value of this bit is 0b.	RW
5	**Master Abort Mode** – Does not apply to PCI Express and must be hardwired to 0b.	RO
6	**Secondary Bus Reset** – Setting this bit triggers a hot reset on the corresponding PCI Express Port. Software must ensure a minimum reset duration (T_{rst}) as defined in the *PCI Local Bus Specification, Revision 3.0*. Software and systems must honor first-access-following-reset timing requirements defined in Section 6.6. Port configuration registers must not be changed, except as required to update Port status. Default value of this bit is 0b.	RW
7	**Fast Back-to-Back Transactions Enable** – Does not apply to PCI Express and must be hardwired to 0b.	RO
8	**Primary Discard Timer** – Does not apply to PCI Express and must be hardwired to 0b.	RO
9	**Secondary Discard Timer** – Does not apply to PCI Express and must be hardwired to 0b.	RO
10	**Discard Timer Status** – Does not apply to PCI Express and must be hardwired to 0b.	RO
11	**Discard Timer SERR# Enable** – Does not apply to PCI Express and must be hardwired to 0b.	RO

7.6. PCI Power Management Capability Structure

This structure is required for all PCI Express device Functions. This Capability is defined in the *PCI Bus Power Management Interface Specification, Revision. 1.2*. The functionality associated with this structure is the same for PCI Express as it is for conventional PCI, and only the added requirements associated with PCI Express are included here.

PCI Express device Functions are required to support D0 and D3 device states (see Section 5.1.1); PCI-PCI Bridge structures representing PCI Express Ports as described in Section 7.1 are required to indicate PME Message passing capability due to the in-band nature of PME messaging for PCI Express.

The PME Status bit for the PCI-PCI Bridge structure representing PCI Express Ports, however, is only Set when the PCI-PCI Bridge Function is itself generating a PME. The PME Status bit is not Set when the Bridge is propagating a PME Message but the PCI-PCI Bridge Function itself is not internally generating a PME.

Figure 7-7 details allocation of register fields for Power Management Capabilities register and Table 7-7 describes the added requirements for this register.

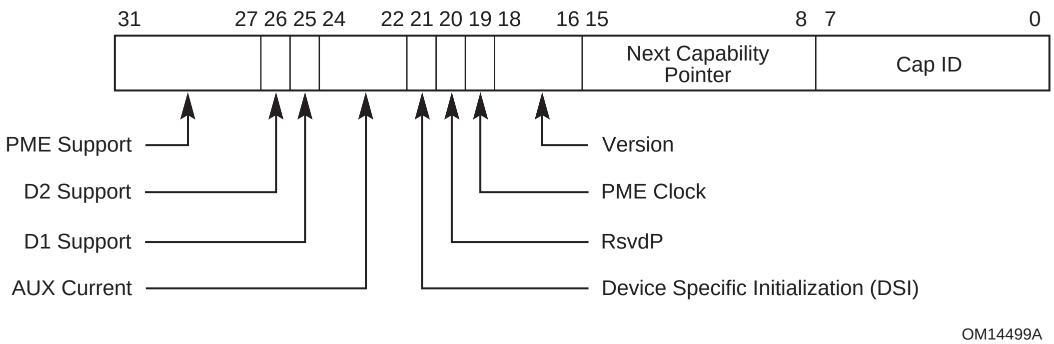

Figure 7-7: Power Management Capabilities Register

Table 7-7: Power Management Capabilities Register Added Requirements

Bit Location	Register Description	Attributes
19	**PME Clock** – Does not apply to PCI Express and must be hardwired to 0b.	Unchanged
31:27	**PME Support** – For a device Function, this 5-bit field indicates the power states in which the Function may generate a PME. Bits 31, 30, and 27 must be Set for PCI-PCI Bridge structures representing Ports on Root Complexes/Switches to indicate that the Bridge will forward PME Messages.	Unchanged

Figure 7-8 details allocation of the register fields for the Power Management Status and Control register and Table 7-8 describes the added requirements for this register.

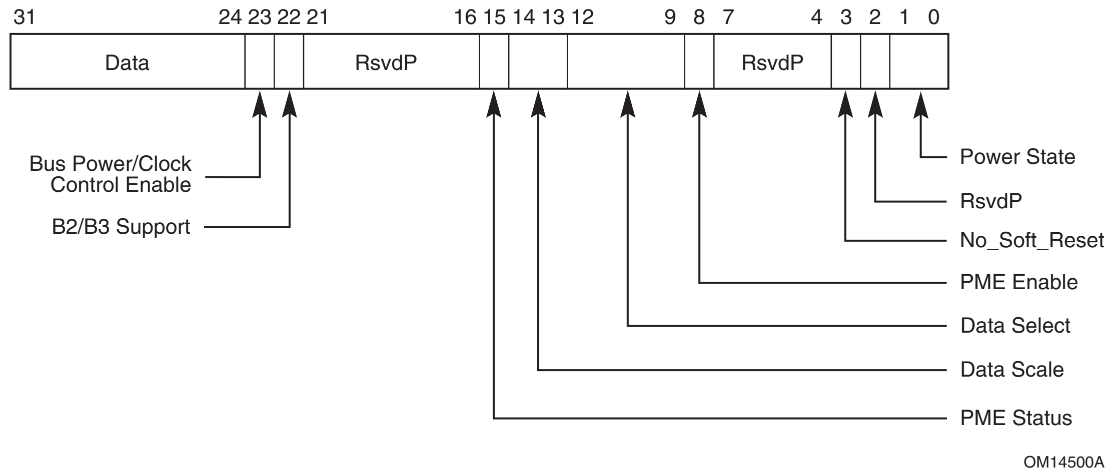

Figure 7-8: Power Management Status/Control Register

Table 7-8: Power Management Status/Control Register Added Requirements

Bit Location	Register Description	Attributes
8	**PME Enable** – No added requirements Note: Device Functions that consume AUX power must preserve the value of this sticky register when AUX power is available. In such Functions, this register value is not modified by Conventional Reset or FLR.	Unchanged
15	**PME Status** – No added requirements Note: Device Functions that consume AUX power must preserve the value of this sticky register when AUX power is available. In such Functions, this register value is not modified by Conventional Reset or FLR.	Unchanged
22	**B2/B3 Support**	Unchanged
23	**Bus Power/Clock Control Enable**	Unchanged

7.7. MSI and MSI-X Capability Structures

All PCI Express device Functions that are capable of generating interrupts must implement MSI or MSI-X or both. MSI, MSI-X, and their Capability structures are defined in the *PCI Local Bus Specification, Revision 3.0*. The functionality associated with these structures defined by conventional PCI is also required for PCI Express. Only added requirements associated with PCI Express are described here.

7.7.1. Vector Control for MSI-X Table Entries

If a Function implements a TPH Requester Capability structure and an MSI-X Capability structure, the Function can optionally use the Vector Control register in each MSI-X Table Entry to store a Steering Tag. See Section 6.17.

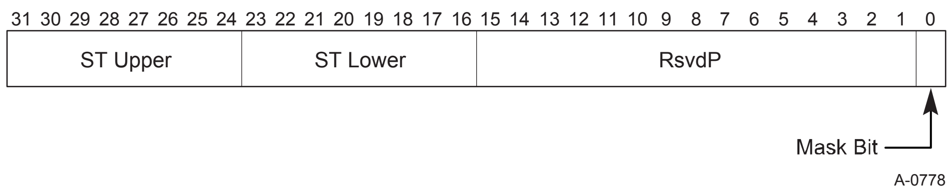

Figure 7-9: Vector Control for MSI-X Table Entries

Table 7-9: Vector Control for MSI-X Table Entries

Bit Location	Register Description	Attributes
0	**Mask Bit** – No added requirements	Unchanged
23:16	**ST Lower** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 10b, then this field contains the lower 8 bits of a Steering Tag. Otherwise, this field is RsvdP. Default value of this field is 0h.	RW
31:24	**ST Upper** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 10b, and the Extended TPH Requester Supported bit is Set, then this field contains the upper 8 bits of a Steering Tag. Otherwise, this field is RsvdP. Default value of this field is 0h.	RW

7.8. PCI Express Capability Structure

PCI Express defines a Capability structure in PCI 3.0 compatible Configuration Space (first 256 bytes) as shown in Figure 7-3. This structure allows identification of a PCI Express device Function and indicates support for new PCI Express features. The PCI Express Capability structure is required for PCI Express device Functions. The Capability structure is a mechanism for enabling PCI software transparent features requiring support on legacy operating systems. In addition to identifying a PCI Express device Function, the PCI Express Capability structure is used to provide access to PCI Express specific Control/Status registers and related Power Management enhancements.

Figure 7-10 details allocation of register fields in the PCI Express Capability structure.

The PCI Express Capabilities, Device Capabilities, Device Status, and Device Control registers are required for all PCI Express device Functions. Device Capabilities 2, Device Status 2, and Device Control 2 registers are required for all PCI Express device Functions that implement capabilities requiring those registers. For device Functions that do not implement the Device Capabilities 2, Device Status 2, and Device Control 2 registers, these spaces must be hardwired to 0b.

The Link Capabilities, Link Status, and Link Control registers are required for all Root Ports, Switch Ports, Bridges, and Endpoints that are not Root Complex Integrated Endpoints. For Functions that do not implement the Link Capabilities, Link Status, and Link Control registers, these spaces must be hardwired to 0. Link Capabilities 2, Link Status 2, and Link Control 2 registers are required for all Root Ports, Switch Ports, Bridges, and Endpoints (except for Root Complex Integrated Endpoints) that implement capabilities requiring those registers. For Functions that do not implement the Link Capabilities 2, Link Status 2, and Link Control 2 registers, these spaces must be hardwired to 0b.

Slot Capabilities, Slot Status, and Slot Control registers are required for Switch Downstream and Root Ports if a slot is implemented on the Port (indicated by the Slot Implemented bit in the PCI Express Capabilities register). For Functions that do not implement the Slot Capabilities, Slot Status, and Slot Control registers, these spaces must be hardwired to 0b, with the exception of the Presence Detect State bit in the Slot Status register of Downstream Ports, which must be hardwired to 1b (see Section 7.8.11). Slot Capabilities 2, Slot Status 2, and Slot Control 2 registers are required for Switch Downstream and Root Ports if a slot is implemented on the Port and the Function implements capabilities requiring those registers. For Functions that do not implement the Slot Capabilities 2, Slot Status 2, and Slot Control 2 registers, these spaces must be hardwired to 0b.

Root Ports and Root Complex Event Collectors must implement the Root Capabilities, Root Status, and Root Control registers. For Functions that do not implement the Root Capabilities, Root Status, and Root Control registers, these spaces must be hardwired to 0b.

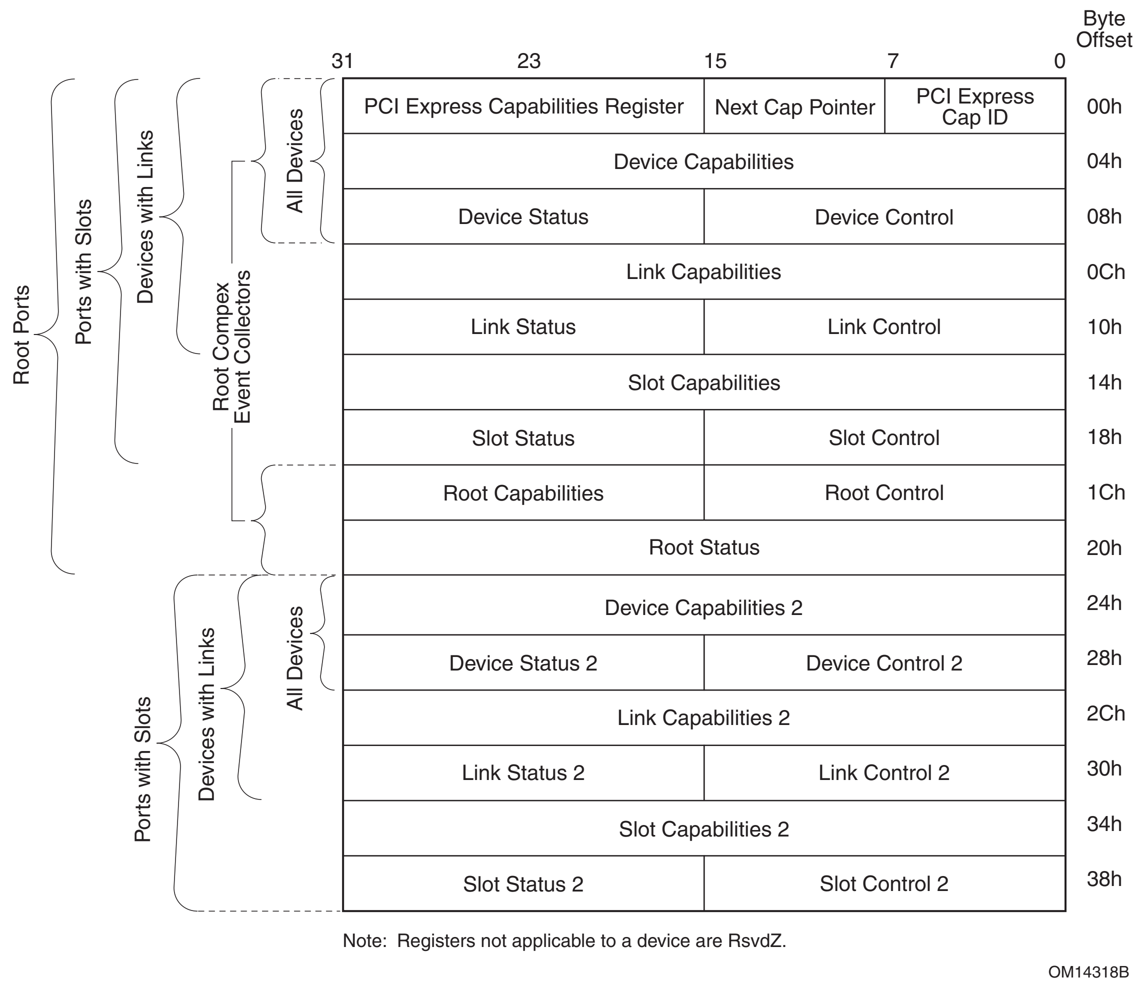

Figure 7-10: PCI Express Capability Structure

7.8.1. PCI Express Capability List Register (Offset 00h)

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 Configuration Space Capability list. Figure 7-11 details allocation of register fields in the PCI Express Capability List register; Table 7-10 provides the respective bit definitions.

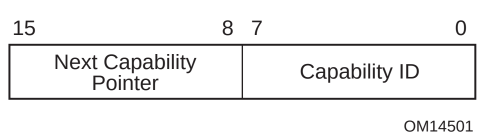

Figure 7-11: PCI Express Capability List Register

Table 7-10: PCI Express Capability List Register

Bit Location	Register Description	Attributes
7:0	**Capability ID** – Indicates the PCI Express Capability structure. This field must return a Capability ID of 10h indicating that this is a PCI Express Capability structure.	RO
15:8	**Next Capability Pointer** – This field contains the offset to the next PCI Capability structure or 00h if no other items exist in the linked list of capabilities.	RO

7.8.2. PCI Express Capabilities Register (Offset 02h)

The PCI Express Capabilities register identifies PCI Express device Function type and associated capabilities. Figure 7-12 details allocation of register fields in the PCI Express Capabilities register; Table 7-11 provides the respective bit definitions.

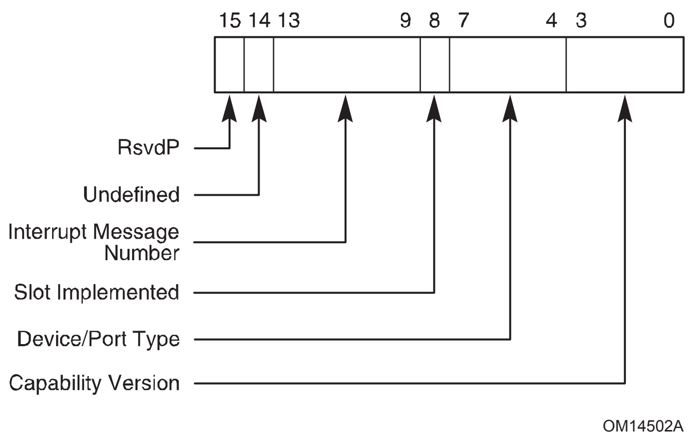

Figure 7-12: PCI Express Capabilities Register

Table 7-11: PCI Express Capabilities Register

Bit Location	Register Description	Attributes
3:0	**Capability Version** – Indicates PCI-SIG defined PCI Express Capability structure version number. A version of the specification that changes the PCI Express Capability structure in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the PCI Express Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a PCI Express Capability structure that is compatible with that piece of software. Must be hardwired to 2h for Functions compliant to this specification.	RO

Bit Location	Register Description	Attributes
7:4	**Device/Port Type** – Indicates the specific type of this PCI Express Function. Note that different Functions in a multi-Function device can generally be of different types. Defined encodings are: 0000b PCI Express Endpoint 0001b Legacy PCI Express Endpoint 0100b Root Port of PCI Express Root Complex* 0101b Upstream Port of PCI Express Switch* 0110b Downstream Port of PCI Express Switch* 0111b PCI Express to PCI/PCI-X Bridge* 1000b PCI/PCI-X to PCI Express Bridge* 1001b Root Complex Integrated Endpoint 1010b Root Complex Event Collector *This value is only valid for Functions that implement a Type 01h PCI Configuration Space header. All other encodings are reserved. Note that the different Endpoint types have notably different requirements in Section 1.3.2 regarding I/O resources, Extended Configuration Space, and other capabilities.	RO
8	**Slot Implemented** – When Set, this bit indicates that the PCI Express Link associated with this Port is connected to a slot (as compared to being connected to an integrated component or being disabled). This field is valid for the following PCI Express Device/Port Types: • Root Port of PCI Express Root Complex • Downstream Port of PCI Express Switch	HwInit

Bit Location	Register Description	Attributes
13:9	**Interrupt Message Number** – This field indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability structure. For MSI, the value in this field indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register. For MSI-X, the value in this field indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If MSI-X is enabled, the value in this field must indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this field must indicate the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this field is undefined.	RO
14	The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate support for TCS Routing. System software should ignore the value read from this bit. System software is permitted to write any value to this bit.	RO

7.8.3. Device Capabilities Register (Offset 04h)

The Device Capabilities register identifies PCI Express device specific capabilities. Figure 7-13 details allocation of register fields in the Device Capabilities register; Table 7-12 provides the respective bit definitions.

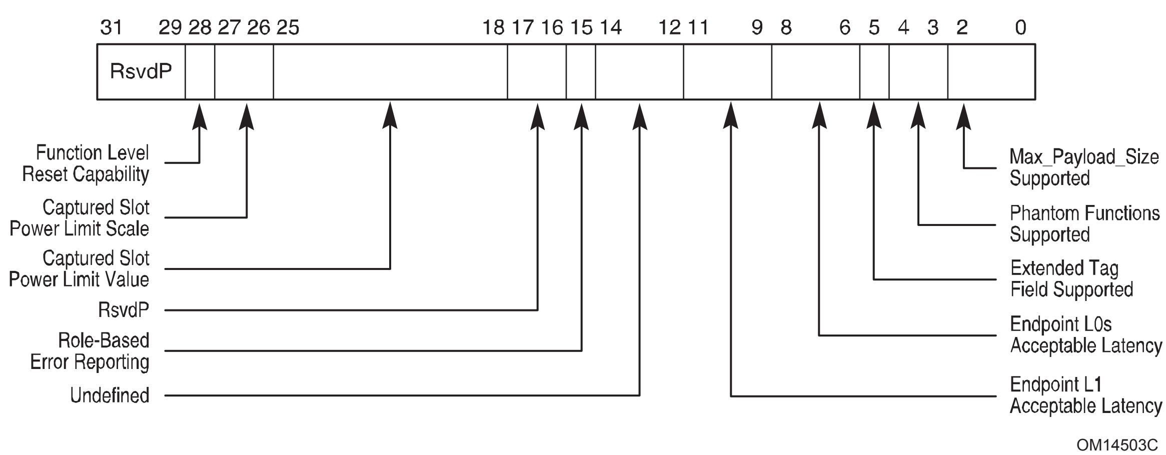

Figure 7-13: Device Capabilities Register

Table 7-12: Device Capabilities Register

Bit Location	Register Description	Attributes
2:0	**Max_Payload_Size Supported** – This field indicates the maximum payload size that the Function can support for TLPs. Defined encodings are: 000b 128 bytes max payload size 001b 256 bytes max payload size 010b 512 bytes max payload size 011b 1024 bytes max payload size 100b 2048 bytes max payload size 101b 4096 bytes max payload size 110b Reserved 111b Reserved The Functions of a multi-Function device are permitted to report different values for this field.	RO

Bit Location	Register Description	Attributes
4:3	**Phantom Functions Supported** – This field indicates the support for use of unclaimed Function Numbers to extend the number of outstanding transactions allowed by logically combining unclaimed Function Numbers (called Phantom Functions) with the Tag identifier (see Section 2.2.6.2 for a description of Tag Extensions). With every Function in an ARI Device, the Phantom Functions Supported field must be set to 00b. The remainder of this field description applies only to non-ARI multi-Function devices. This field indicates the number of most significant bits of the Function Number portion of Requester ID that are logically combined with the Tag identifier. Defined encodings are: 00b No Function Number bits are used for Phantom Functions. Multi-Function devices are permitted to implement up to 8 independent Functions. 01b The most significant bit of the Function number in Requester ID is used for Phantom Functions; a multi-Function device is permitted to implement Functions 0-3. Functions 0, 1, 2, and 3 are permitted to use Function Numbers 4, 5, 6, and 7 respectively as Phantom Functions. 10b The two most significant bits of Function Number in Requester ID are used for Phantom Functions; a multi-Function device is permitted to implement Functions 0-1. Function 0 is permitted to use Function Numbers 2, 4, and 6 for Phantom Functions. 4, and 6 for Phantom Functions. Function 1 is permitted to use Function Numbers 3, 5, and 7 as Phantom Functions. 11b All 3 bits of Function Number in Requester ID used for Phantom Functions. The device must have a single Function 0 that is permitted to use all other Function Numbers as Phantom Functions. Note that Phantom Function support for the Function must be enabled by the Phantom Functions Enable field in the Device Control register before the Function is permitted to use the Function Number field in the Requester ID for Phantom Functions.	RO
5	**Extended Tag Field Supported** – This bit indicates the maximum supported size of the Tag field as a Requester. Defined encodings are: 0b 5-bit Tag field supported 1b 8-bit Tag field supported Note that 8-bit Tag field generation must be enabled by the Extended Tag Field Enable bit in the Device Control register before 8-bit Tags can be generated by the Requester.	RO

Bit Location	Register Description	Attributes
8:6	**Endpoint L0s Acceptable Latency** – This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance. Defined encodings are: 000b Maximum of 64 ns 001b Maximum of 128 ns 010b Maximum of 256 ns 011b Maximum of 512 ns 100b Maximum of 1 μs 101b Maximum of 2 μs 110b Maximum of 4 μs 111b No limit For Functions other than Endpoints, this field is Reserved and must be hardwired to 000b.	RO
11:9	**Endpoint L1 Acceptable Latency** – This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance. Defined encodings are: 000b Maximum of 1 μs 001b Maximum of 2 μs 010b Maximum of 4 μs 011b Maximum of 8 μs 100b Maximum of 16 μs 101b Maximum of 32 μs 110b Maximum of 64 μs 111b No limit For Functions other than Endpoints, this field is Reserved and must be hardwired to 000b.	RO

Bit Location	Register Description	Attributes
12	The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate that an Attention Button is implemented on the adapter and electrically controlled by the component on the adapter. System software must ignore the value read from this bit. System software is permitted to write any value to this bit.	RO
13	The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate that an Attention Indicator is implemented on the adapter and electrically controlled by the component on the adapter. System software must ignore the value read from this bit. System software is permitted to write any value to this bit.	RO
14	The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate that a Power Indicator is implemented on the adapter and electrically controlled by the component on the adapter. System software must ignore the value read from this bit. System software is permitted to write any value to this bit.	RO
15	**Role-Based Error Reporting** – When Set, this bit indicates that the Function implements the functionality originally defined in the Error Reporting ECN for *PCI Express Base Specification, Revision 1.0a*, and later incorporated into *PCI Express Base Specification, Revision 1.1*. This bit must be Set by all Functions conforming to the ECN, *PCI Express Base Specification, Revision 1.1*., or subsequent *PCI Express Base Specification* revisions.	RO
25:18	**Captured Slot Power Limit Value** (Upstream Ports only) – In combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by slot. Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. This value is set by the Set_Slot_Power_Limit Message or hardwired to 00h (see Section 6.9). The default value is 00h.	RO
27:26	**Captured Slot Power Limit Scale** (Upstream Ports only) – Specifies the scale used for the Slot Power Limit Value. Range of Values: 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x This value is set by the Set_Slot_Power_Limit Message or hardwired to 00b (see Section 6.9). The default value is 00b.	RO

Bit Location	Register Description	Attributes
28	**Function Level Reset Capability** – A value of 1b indicates the Function supports the optional Function Level Reset mechanism described in Section 6.6.2. This field applies to Endpoints only. For all other Function types this bit must be hardwired to 0b.	RO

7.8.4. Device Control Register (Offset 08h)

The Device Control register controls PCI Express device specific parameters. Figure 7-14 details allocation of register fields in the Device Control register; Table 7-13 provides the respective bit definitions.

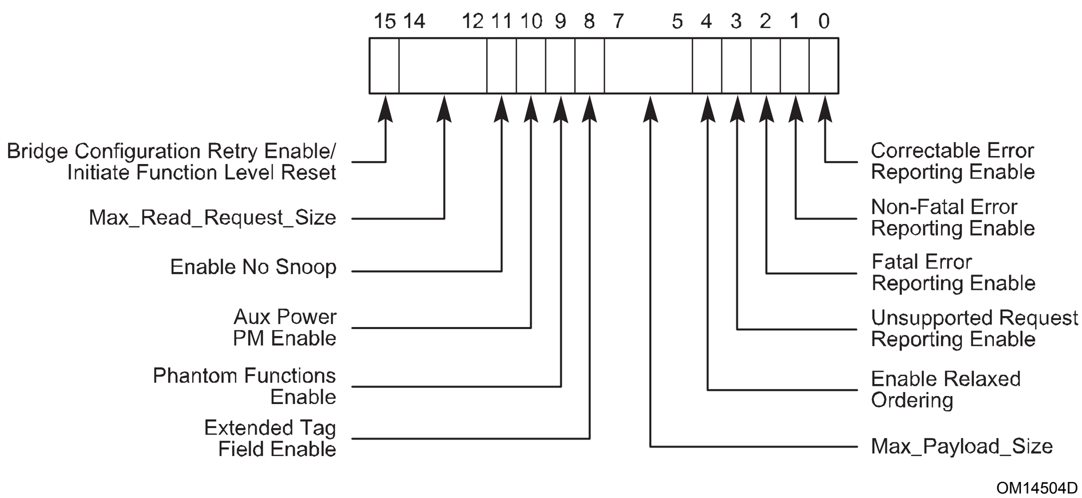

Figure 7-14: Device Control Register

Table 7-13: Device Control Register

Bit Location	Register Description	Attributes
0	**Correctable Error Reporting Enable** – This bit, in conjunction with other bits, controls sending ERR_COR Messages (see Section 6.2.5 and Section 6.2.6 for details). For a multi-Function device, this bit controls error reporting for each Function from point-of-view of the respective Function. For a Root Port, the reporting of correctable errors is internal to the root. No external ERR_COR Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
1	**Non-Fatal Error Reporting Enable** – This bit, in conjunction with other bits, controls sending ERR_NONFATAL Messages (see Section 6.2.5 and Section 6.2.6 for details). For a multi-Function device, this bit controls error reporting for each Function from point-of-view of the respective Function. For a Root Port, the reporting of Non-fatal errors is internal to the root. No external ERR_NONFATAL Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
2	**Fatal Error Reporting Enable** – This bit, in conjunction with other bits, controls sending ERR_FATAL Messages (see Section 6.2.5 and Section 6.2.6 for details). For a multi-Function device, this bit controls error reporting for each Function from point-of-view of the respective Function. For a Root Port, the reporting of Fatal errors is internal to the root. No external ERR_FATAL Message is generated. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW
3	**Unsupported Request Reporting Enable** – This bit, in conjunction with other bits, controls the signaling of Unsupported Requests by sending error Messages (see Section 6.2.5 and Section 6.2.6 for details). For a multi-Function device, this bit controls error reporting for each Function from point-of-view of the respective Function. A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
4	**Enable Relaxed Ordering** – If this bit is Set, the Function is permitted to set the Relaxed Ordering bit in the Attributes field of transactions it initiates that do not require strong write ordering (see Section 2.2.6.4 and Section 2.4). A Function is permitted to hardwire this bit to 0b if it never sets the Relaxed Ordering attribute in transactions it initiates as a Requester. Default value of this bit is 1b.	RW
7:5	**Max_Payload_Size** – This field sets maximum TLP payload size for the Function. As a Receiver, the Function must handle TLPs as large as the set value. As a Transmitter, the Function must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the Max_Payload_Size Supported in the Device Capabilities register (see Section 7.8.3). Defined encodings for this field are: 000b 128 bytes max payload size 001b 256 bytes max payload size 010b 512 bytes max payload size 011b 1024 bytes max payload size 100b 2048 bytes max payload size 101b 4096 bytes max payload size 110b Reserved 111b Reserved Functions that support only the 128-byte max payload size are permitted to hardwire this field to 000b. System software is not required to program the same value for this field for all the Functions of a multi-Function device. Refer to Section 2.2.2 for important guidance. For ARI Devices, Max_Payload_Size is determined solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Default value of this field is 000b.	RW
8	**Extended Tag Field Enable** – When Set, this bit enables a Function to use an 8-bit Tag field as a Requester. If the bit is Clear, the Function is restricted to a 5-bit Tag field (see Section 2.2.6.2 for a description of Tag extensions). Functions that do not implement this capability hardwire this bit to 0b. Default value of this bit is implementation specific.	RW

Bit Location	Register Description	Attributes
9	**Phantom Functions Enable** – When Set, this bit enables a Function to use unclaimed Functions as Phantom Functions to extend the number of outstanding transaction identifiers. If the bit is Clear, the Function is not allowed to use Phantom Functions (see Section 2.2.6.2 for a description of Tag extensions). Functions that do not implement this capability hardwire this bit to 0b. Default value of this bit is 0b.	RW
10	**Auxiliary (AUX) Power PM Enable** – When Set this bit, enables a Function to draw AUX power independent of PME AUX power. Functions that require AUX power on legacy operating systems should continue to indicate PME AUX power requirements. AUX power is allocated as requested in the AUX_Current field of the Power Management Capabilities register (PMC), independent of the PME_En bit in the Power Management Control/Status register (PMCSR) (see Chapter 5). For multi-Function devices, a component is allowed to draw AUX power if at least one of the Functions has this bit set. Note: Functions that consume AUX power must preserve the value of this sticky register when AUX power is available. In such Functions, this register value is not modified by Conventional Reset. Functions that do not implement this capability hardwire this bit to 0b.	RWS
11	**Enable No Snoop** – If this bit is Set, the Function is permitted to Set the No Snoop bit in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency (see Section 2.2.6.5). Note that setting this bit to 1b should not cause a Function to Set the No Snoop attribute on all transactions that it initiates. Even when this bit is Set, a Function is only permitted to Set the No Snoop attribute on a transaction when it can guarantee that the address of the transaction is not stored in any cache in the system. This bit is permitted to be hardwired to 0b if a Function would never Set the No Snoop attribute in transactions it initiates. Default value of this bit is 1b.	RW

Bit Location	Register Description	Attributes
14:12	**Max_Read_Request_Size** – This field sets the maximum Read Request size for the Function as a Requester. The Function must not generate Read Requests with size exceeding the set value. Defined encodings for this field are: 000b 128 bytes maximum Read Request size 001b 256 bytes maximum Read Request size 010b 512 bytes maximum Read Request size 011b 1024 bytes maximum Read Request size 100b 2048 bytes maximum Read Request size 101b 4096 bytes maximum Read Request size 110b Reserved 111b Reserved Functions that do not generate Read Requests larger than 128 bytes and Functions that do not generate Read Requests on their own behalf are permitted to implement this field as Read Only (RO) with a value of 000b. Default value of this field is 010b.	RW
15	*PCI Express to PCI/PCI-X Bridges:* **Bridge Configuration Retry Enable** – When Set, this bit enables PCI Express to PCI/PCI-X bridges to return Configuration Request Retry Status (CRS) in response to Configuration Requests that target devices below the bridge. Refer to the *PCI Express to PCI/PCI-X Bridge Specification, Revision 1.0* for further details. Default value of this bit is 0b. *Endpoints with Function Level Reset Capability set to 1b:* **Initiate Function Level Reset** – A write of 1b initiates Function Level Reset to the Function. The value read by software from this bit is always 0b. *All others:* **Reserved** – Must hardwire the bit to 0b.	*PCI Express to PCI/PCI-X Bridges:* RW *FLR Capable Endpoints:* RW *All others:* RsvdP

IMPLEMENTATION NOTE

Software UR Reporting Compatibility with 1.0a Devices

With 1.0a device Functions,[96] if the Unsupported Request Reporting Enable bit is set, the Function when operating as a Completer will send an uncorrectable error Message (if enabled) when a UR error is detected. On platforms where an uncorrectable error Message is handled as a System Error, this will break PC-compatible Configuration Space probing, so software/firmware on such platforms may need to avoid setting the Unsupported Request Reporting Enable bit.

With device Functions implementing Role-Based Error Reporting, setting the Unsupported Request Reporting Enable bit will not interfere with PC-compatible Configuration Space probing, assuming that the severity for UR is left at its default of non-fatal. However, setting the Unsupported Request Reporting Enable bit will enable the Function to report UR errors[97] detected with posted Requests, helping avoid this case for potential silent data corruption.

On platforms where robust error handling and PC-compatible Configuration Space probing is required, it is suggested that software or firmware have the Unsupported Request Reporting Enable bit Set for Role-Based Error Reporting Functions, but clear for 1.0a Functions. Software or firmware can distinguish the two classes of Functions by examining the Role-Based Error Reporting bit in the Device Capabilities register.

IMPLEMENTATION NOTE

Use of Max_Payload_Size

The Max_Payload_Size mechanism allows software to control the maximum payload in packets sent by Endpoints to balance latency versus bandwidth trade-offs, particularly for isochronous traffic.

If software chooses to program the Max_Payload_Size of various System Elements to non-default values, it must take care to ensure that each packet does not exceed the Max_Payload_Size parameter of any System Element along the packet's path. Otherwise, the packet will be rejected by the System Element whose Max_Payload_Size parameter is too small.

Discussion of specific algorithms used to configure Max_Payload_Size to meet this requirement is beyond the scope of this specification, but software should base its algorithm upon factors such as the following:

- the Max_Payload_Size capability of each System Element within a hierarchy
- awareness of when System Elements are added or removed through Hot-Plug operations

[96] In this context, "1.0a devices" are devices that do not implement Role-Based Error Reporting.

[97] With Role-Based Error Reporting devices, setting the SERR# Enable bit in the Command register also implicitly enables UR reporting.

- knowing which System Elements send packets to each other, what type of traffic is carried, what type of transactions are used, or if packet sizes are constrained by other mechanisms

For the case of firmware that configures System Elements in preparation for running legacy operating system environments, the firmware may need to avoid programming a Max_Payload_Size above the default of 128 bytes, which is the minimum supported by Endpoints.

For example, if the operating system environment does not comprehend PCI Express, firmware probably should not program a non-default Max_Payload_Size for a hierarchy that supports Hot-Plug operations. Otherwise, if no software is present to manage Max_Payload_Size settings when a new element is added, improper operation may result. Note that a newly added element may not even support a Max_Payload_Size setting as large as the rest of the hierarchy, in which case software may need to deny enabling the new element or reduce the Max_Payload_Size settings of other elements.

IMPLEMENTATION NOTE

Use of Max_Read_Request_Size

The Max_Read_Request_Size mechanism allows improved control of bandwidth allocation in systems where quality of service (QoS) is important for the target applications. For example, an arbitration scheme based on counting Requests (and not the sizes of those Requests) provides imprecise bandwidth allocation when some Requesters use much larger sizes than others. The Max_Read_Request_Size mechanism can be used to force more uniform allocation of bandwidth, by restricting the upper size of Read Requests.

7.8.5. Device Status Register (Offset 0Ah)

The Device Status register provides information about PCI Express device (Function) specific parameters. Figure 7-15 details allocation of register fields in the Device Status register; Table 7-14 provides the respective bit definitions.

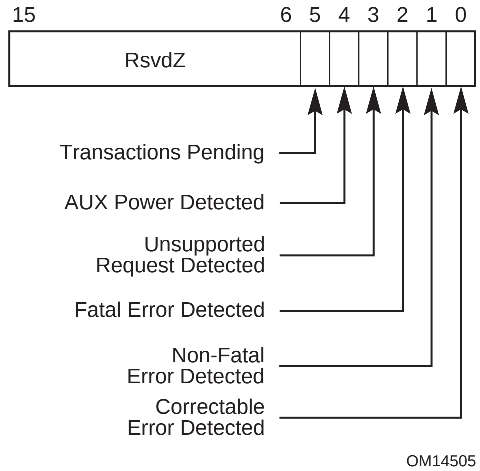

Figure 7-15: Device Status Register

Table 7-14: Device Status Register

Bit Location	Register Description	Attributes
0	**Correctable Error Detected** – This bit indicates status of correctable errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a multi-Function device, each Function indicates status of errors as perceived by the respective Function. For Functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Correctable Error Mask register. Default value of this bit is 0b.	RW1C
1	**Non-Fatal Error Detected** – This bit indicates status of Non-fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a multi-Function device, each Function indicates status of errors as perceived by the respective Function. For Functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Uncorrectable Error Mask register. Default value of this bit is 0b.	RW1C
2	**Fatal Error Detected** – This bit indicates status of Fatal errors detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a multi-Function device, each Function indicates status of errors as perceived by the respective Function. For Functions supporting Advanced Error Handling, errors are logged in this register regardless of the settings of the Uncorrectable Error Mask register. Default value of this bit is 0b.	RW1C

Bit Location	Register Description	Attributes
3	**Unsupported Request Detected** – This bit indicates that the Function received an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. For a multi-Function device, each Function indicates status of errors as perceived by the respective Function. Default value of this bit is 0b.	RW1C
4	**AUX Power Detected** – Functions that require AUX power report this bit as Set if AUX power is detected by the Function.	RO
5	**Transactions Pending** – *Endpoints:* When Set, this bit indicates that the Function has issued Non-Posted Requests that have not been completed. A Function reports this bit cleared only when all outstanding Non-Posted Requests have completed or have been terminated by the Completion Timeout mechanism. This bit must also be cleared upon the completion of an FLR. *Root and Switch Ports:* When Set, this bit indicates that a Port has issued Non-Posted Requests on its own behalf (using the Port's own Requester ID) which have not been completed. The Port reports this bit cleared only when all such outstanding Non-Posted Requests have completed or have been terminated by the Completion Timeout mechanism. Note that Root and Switch Ports implementing only the functionality required by this document do not issue Non-Posted Requests on their own behalf, and therefore are not subject to this case. Root and Switch Ports that do not issue Non-Posted Requests on their own behalf hardwire this bit to 0b.	RO

7.8.6. Link Capabilities Register (Offset 0Ch)

The Link Capabilities register identifies PCI Express Link specific capabilities. Figure 7-16 details allocation of register fields in the Link Capabilities register; Table 7-15 provides the respective bit definitions.

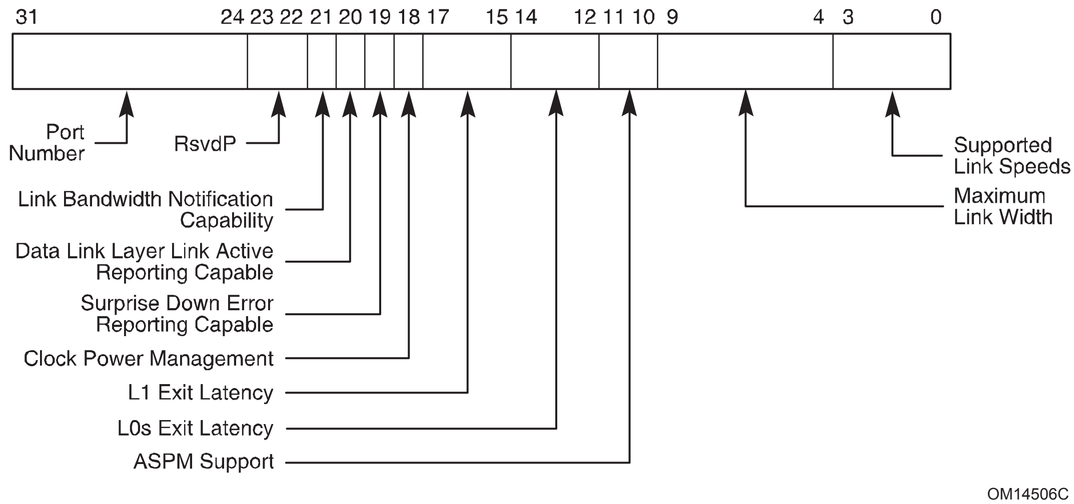

Figure 7-16: Link Capabilities Register

Table 7-15: Link Capabilities Register

Bit Location	Register Description	Attributes
3:0	**Supported Link Speeds** – This field indicates the supported Link speed(s) of the associated Port. Defined encodings are: 0001b 2.5 GT/s Link speed supported 0010b 5.0 GT/s and 2.5 GT/s Link speeds supported All other encodings are reserved. Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	RO

Bit Location	Register Description	Attributes
9:4	**Maximum Link Width** – This field indicates the maximum Link width (xN – corresponding to N Lanes) implemented by the component. This value is permitted to exceed the number of Lanes routed to the slot (Downstream Port), adapter connector (Upstream Port), or in the case of component-to-component connections, the actual wired connection width. Defined encodings are: 000000b Reserved 000001b x1 000010b x2 000100b x4 001000b x8 001100b x12 010000b x16 100000b x32 Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	RO
11:10	**Active State Power Management (ASPM) Support** – This field indicates the level of ASPM supported on the given PCI Express Link. Defined encodings are: 00b Reserved 01b L0s Entry Supported 10b Reserved 11b L0s and L1 Supported Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	RO

Bit Location	Register Description	Attributes
14:12	**L0s Exit Latency** – This field indicates the L0s exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. Defined encodings are: 000b Less than 64 ns 001b 64 ns to less than 128 ns 010b 128 ns to less than 256 ns 011b 256 ns to less than 512 ns 100b 512 ns to less than 1 µs 101b 1 µs to less than 2 µs 110b 2 µs-4 µs 111b More than 4 µs Note that exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	RO
17:15	**L1 Exit Latency** – This field indicates the L1 exit latency for the given PCI Express Link. The value reported indicates the length of time this Port requires to complete transition from L1 to L0. Defined encodings are: 000b Less than 1µs 001b 1 µs to less than 2 µs 010b 2 µs to less than 4 µs 011b 4 µs to less than 8 µs 100b 8 µs to less than 16 µs 101b 16 µs to less than 32 µs 110b 32 µs-64 µs 111b More than 64 µs Note that exit latencies may be influenced by PCI Express reference clock configuration depending upon whether a component uses a common or separate reference clock. Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	RO

Bit Location	Register Description	Attributes
18	**Clock Power Management** – For Upstream Ports, a value of 1b in this bit indicates that the component tolerates the removal of any reference clock(s) via the "clock request" (CLKREQ#) mechanism when the Link is in the L1 and L2/L3 Ready Link states. A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. This Capability is applicable only in form factors that support "clock request" (CLKREQ#) capability. For a multi-Function device associated with an Upstream Port, each Function indicates its capability independently. Power Management configuration software must only permit reference clock removal if all Functions of the multi-Function device indicate a 1b in this bit. For ARI Devices, all Functions must indicate the same value in this bit. For Downstream Ports, this bit must be hardwired to 0b.	RO
19	**Surprise Down Error Reporting Capable** – For a Downstream Port, this bit must be Set if the component supports the optional capability of detecting and reporting a Surprise Down error condition. For Upstream Ports and components that do not support this optional capability, this bit must be hardwired to 0b.	RO
20	**Data Link Layer Link Active Reporting Capable** – For a Downstream Port, this bit must be hardwired to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine. For a hot-plug capable Downstream Port (as indicated by the Hot-Plug Capable bit of the Slot Capabilities register), this bit must be hardwired to 1b. For Upstream Ports and components that do not support this optional capability, this bit must be hardwired to 0b.	RO
21	**Link Bandwidth Notification Capability** – A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. This capability is required for all Root Ports and Switch Downstream Ports supporting Links wider than x1 and/or multiple Link speeds. This field is not applicable and is reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b.	RO
31:24	**Port Number** – This field indicates the PCI Express Port number for the given PCI Express Link. Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions.	HwInit

If the L1 state is not supported for ASPM (as reported in the ASPM Support field), then the L1 Exit latency field is ignored.

7.8.7. Link Control Register (Offset 10h)

The Link Control register controls PCI Express Link specific parameters. Figure 7-17 details allocation of register fields in the Link Control register; Table 7-16 provides the respective bit definitions.

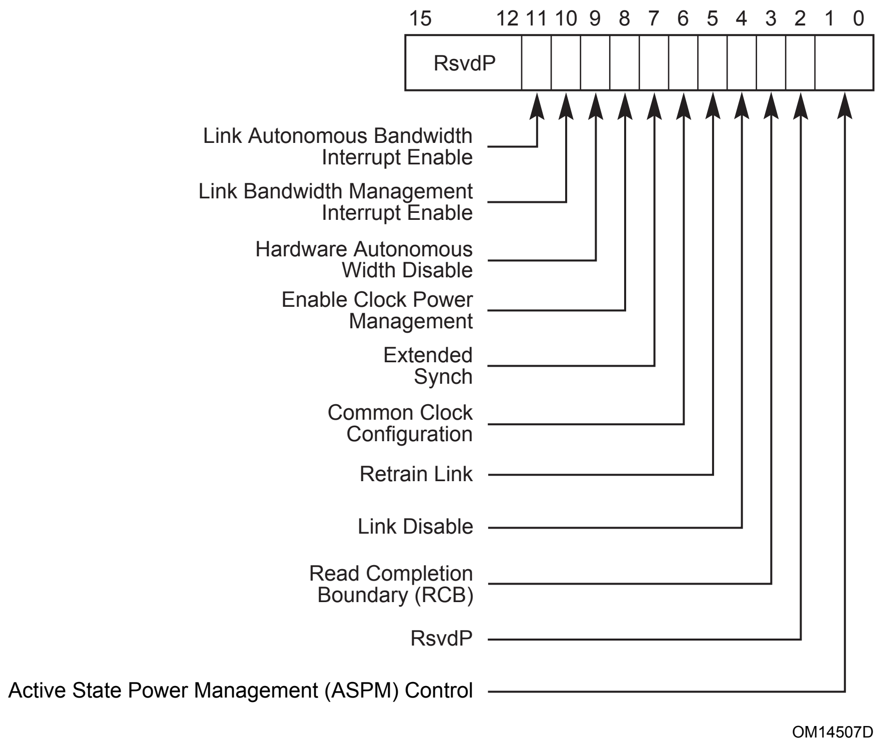

Figure 7-17: Link Control Register

Table 7-16: Link Control Register

Bit Location	Register Description	Attributes
1:0	**Active State Power Management (ASPM) Control** – This field controls the level of ASPM supported on the given PCI Express Link. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled Note: "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. When disabling ASPM L1, software must disable ASPM L1 in the Downstream component on a Link prior to disabling ASPM L1 in the Upstream component on that Link. ASPM L1 must only be enabled on the Downstream component if both components on a Link support ASPM L1. For multi-Function devices (including ARI Devices), it is recommended that software program the same value for this field in all Functions. For non-ARI multi-Function devices,only capabilities enabled in all Functions are enabled for the component as a whole. For ARI Devices, ASPM Control is determined solely by the setting in Function 0, regardless of Function 0's D-state. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Default value of this field is 00b unless otherwise required by a particular form factor.	RW

Bit Location	Register Description	Attributes
3	*Root Ports:* **Read Completion Boundary (RCB)** – Indicates the RCB value for the Root Port. Refer to Section 2.3.1.1 for the definition of the parameter RCB. Defined encodings are: 0b 64 byte 1b 128 byte This bit is hardwired for a Root Port and returns its RCB support capabilities. *Endpoints and Bridges:* **Read Completion Boundary (RCB)** – Optionally Set by configuration software to indicate the RCB value of the Root Port Upstream from the Endpoint or Bridge. Refer to Section 2.3.1.1 for the definition of the parameter RCB. Defined encodings are: 0b 64 byte 1b 128 byte Configuration software must only Set this bit if the Root Port Upstream from the Endpoint or Bridge reports an RCB value of 128 bytes (a value of 1b in the Read Completion Boundary bit). Default value of this bit is 0b. Functions that do not implement this feature must hardwire the bit to 0b. *Switch Ports:* Not applicable – must hardwire the bit to 0b	*Root Ports:* RO *Endpoints and Bridges:* RW *Switch Ports:* RO
4	**Link Disable** – This bit disables the Link by directing the LTSSM to the Disabled state when Set; this bit is reserved on Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. Writes to this bit are immediately reflected in the value read from the bit, regardless of actual Link state. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
5	**Retrain Link** – A write of 1b to this bit initiates Link retraining by directing the Physical Layer LTSSM to the Recovery state. If the LTSSM is already in Recovery or Configuration, re-entering Recovery is permitted but not required. Reads of this bit always return 0b. It is permitted to write 1b to this bit while simultaneously writing modified values to other fields in this register. If the LTSSM is not already in Recovery or Configuration, the resulting Link training must use the modified values. If the LTSSM is already in Recovery or Configuration, the modified values are not required to affect the Link training that's already in progress. This bit is not applicable and is reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. This bit always returns 0b when read.	RW
6	**Common Clock Configuration** – When Set, this bit indicates that this component and the component at the opposite end of this Link are operating with a distributed common reference clock. A value of 0b indicates that this component and the component at the opposite end of this Link are operating with asynchronous reference clock. For non-ARI multi-Function devices, software must program the same value for this bit in all Functions. If not all Functions are Set, then the component must as a whole assume that its reference clock is not common with the Upstream component. For ARI Devices, Common Clock Configuration is determined solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Components utilize this common clock configuration information to report the correct L0s and L1 Exit Latencies. After changing the value in this bit in both components on a Link, software must trigger the Link to retrain by writing a 1b to the Retrain Link bit of the Downstream Port. Default value of this bit is 0b.	RW
7	**Extended Synch** – When Set, this bit forces the transmission of additional Ordered Sets when exiting the L0s state (see Section 4.2.4.5) and when in the Recovery state (see Section 4.2.6.4.1). This mode provides external devices (e.g., logic analyzers) monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 state and resumes communication. For multi-Function devices if any Function has this bit Set, then the component must transmit the additional Ordered Sets when exiting L0s. Default value for this bit is 0b.	RW

Bit Location	Register Description	Attributes
8	**Enable Clock Power Management** – Applicable only for Upstream Ports and with form factors that support a “Clock Request” (CLKREQ#) mechanism, this bit operates as follows: 0b Clock power management is disabled and device must hold CLKREQ# signal low. 1b When this bit is Set, the device is permitted to use CLKREQ# signal to power manage Link clock according to protocol defined in appropriate form factor specification. For a non-ARI multi-Function device, power-management-configuration software must only Set this bit if all Functions of the multi-Function device indicate a 1b in the Clock Power Management bit of the Link Capabilities register. The component is permitted to use the CLKREQ# signal to power manage Link clock only if this bit is Set for all Functions. For ARI Devices, Clock Power Management is enabled solely by the setting in Function 0. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component. Downstream Ports and components that do not support Clock Power Management (as indicated by a 0b value in the Clock Power Management bit of the Link Capabilities register) must hardwire this bit to 0b. Default value of this bit is 0b.	RW
9	**Hardware Autonomous Width Disable** – When Set, this bit disables hardware from changing the Link width for reasons other than attempting to correct unreliable Link operation by reducing Link width. For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RW, and only Function 0 controls the component’s Link behavior. In all other Functions of that device, this bit is of type RsvdP. Components that do not implement the ability autonomously to change Link width are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RW/RsvdP (see description)
10	**Link Bandwidth Management Interrupt Enable** – When Set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been Set. This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
11	**Link Autonomous Bandwidth Interrupt Enable** – When Set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been Set. This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b. Default value of this bit is 0b.	RW

IMPLEMENTATION NOTE

Software Compatibility with ARI Devices

With the ASPM Control field, Common Clock Configuration bit, and Enable Clock Power Management bit in the Link Control register, there are potential software compatibility issues with ARI Devices since these controls operate strictly off the settings in Function 0 instead of the settings in all Functions.

With compliant software, there should be no issues with the Common Clock Configuration bit, since software is required to set this bit the same in all Functions.

With the Enable Clock Power Management bit, there should be no compatibility issues with software that sets this bit the same in all Functions. However, if software does not set this bit the same in all Functions, and relies on each Function having the ability to prevent Clock Power Management from being enabled, such software may have compatibility issues with ARI Devices.

With the ASPM Control field, there should be no compatibility issues with software that sets this bit the same in all Functions. However, if software does not set this bit the same in all Functions, and relies on each Function in D0 state having the ability to prevent ASPM from being enabled, such software may have compatibility issues with ARI Devices.

IMPLEMENTATION NOTE

Avoiding Race Conditions When Using the Retrain Link Bit

When software changes Link control parameters and writes a 1b to the Retrain Link bit in order to initiate Link training using the new parameter settings, special care is required in order to avoid certain race conditions. At any instant the LTSSM may transition to the Recovery or Configuration state due to normal Link activity, without software awareness. If the LTSSM is already in Recovery or Configuration when software writes updated parameters to the Link Control register, as well as a 1b, to the Retrain Link bit, the LTSSM might not use the updated parameter settings with the current Link training, and the current Link training might not achieve the results that software intended.

To avoid this potential race condition, it is highly recommended that software use the following algorithm or something similar:

1. Software sets the relevant Link control parameters to the desired settings without writing a 1b to the Retrain Link bit.

2. Software polls the Link Training bit in the Link Status register until the value returned is 0b.

3. Software writes a 1b to the Retrain Link bit without changing any other fields in the Link Control register.

The above algorithm guarantees that Link training will be based on the Link control parameter settings that software intends.

IMPLEMENTATION NOTE

Use of the Slot Clock Configuration and Common Clock Configuration Bits

In order to determine the common clocking configuration of components on opposite ends of a Link that crosses a connector, two pieces of information are required. The following description defines these requirements.

The first necessary piece of information is whether the Port that connects to the slot uses a clock that has a common source and therefore constant phase relationship to the clock signal provided on the slot. This information is provided by the system side component through a hardware initialized bit (Slot Clock Configuration) in its Link Status register. Note that some electromechanical form factor specifications may require the Port that connects to the slot use a clock that has a common source to the clock signal provided on the slot.

The second necessary piece of information is whether the component on the adapter uses the clock supplied on the slot or one generated locally on the adapter. The adapter design and layout will determine whether the component is connected to the clock source provided by the slot. A component going onto this adapter should have some hardware initialized method for the adapter design/designer to indicate the configuration used for this particular adapter design. This information is reported by bit 12 (Slot Clock Configuration) in the Link Status register of each Function in the Upstream Port. Note that some electromechanical form factor specifications may require the Port on the adapter to use the clock signal provided on the connector.

System firmware or software will read this value from the components on both ends of a physical Link. If both components report the use of a common clock connection this firmware/software will program bit 6 (Common Clock Configuration) of the Link Control register to a one on both components connected to the Link. Each component uses this bit to determine the length of time required to re-synch its Receiver to the opposing component's Transmitter when exiting L0s.

This value is reported as a time value in bits 12-14 of the Link Capabilities register (offset 0Ch) and is sent to the opposing Transmitter as part if the initialization process as N_FTS. Components would be expected to require much longer synch times without common clocking and would therefore report a longer L0s exit latency in bits 12-14 of the Link Capabilities register and would send a larger number for N_FTS during training. This forces a requirement that whatever software changes this bit should force a Link retrain in order to get the correct N-FTS set for the Receivers at both ends of the Link.

7.8.8. Link Status Register (Offset 12h)

The Link Status register provides information about PCI Express Link specific parameters. Figure 7-18 details allocation of register fields in the Link Status register; Table 7-17 provides the respective bit definitions.

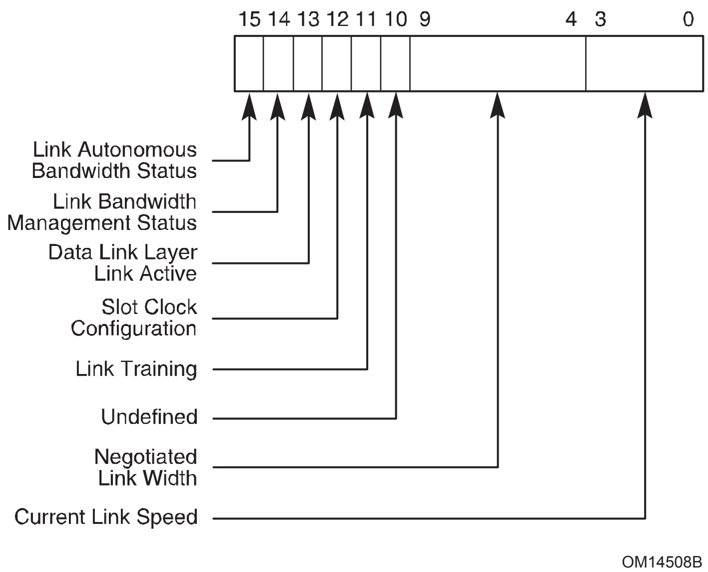

Figure 7-18: Link Status Register

Table 7-17: Link Status Register

Bit Location	Register Description	Attributes
3:0	**Current Link Speed** – This field indicates the negotiated Link speed of the given PCI Express Link. Defined encodings are: 0001b 2.5 GT/s PCI Express Link 0010b 5.0 GT/s PCI Express Link All other encodings are reserved. The value in this field is undefined when the Link is not up.	RO

Bit Location	Register Description	Attributes
9:4	**Negotiated Link Width** – This field indicates the negotiated width of the given PCI Express Link. Defined encodings are: 00 0001b x1 00 0010b x2 00 0100b x4 00 1000b x8 00 1100b x12 01 0000b x16 10 0000b x32 All other encodings are reserved. The value in this field is undefined when the Link is not up.	RO
10	**Undefined** – The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate a Link Training Error. System software must ignore the value read from this bit. System software is permitted to write any value to this bit.	RO
11	**Link Training** – This read-only bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state, or that 1b was written to the Retrain Link bit but Link training has not yet begun. Hardware clears this bit when the LTSSM exits the Configuration/Recovery state. This bit is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches, and must be hardwired to 0b.	RO
12	**Slot Clock Configuration** – This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. If the device uses an independent clock irrespective of the presence of a reference on the connector, this bit must be clear. For a multi-Function device, each Function must report the same value for this bit.	HwInit
13	**Data Link Layer Link Active** – This bit indicates the status of the Data Link Control and Management State Machine. It returns a 1b to indicate the DL_Active state, 0b otherwise. This bit must be implemented if the corresponding Data Link Layer Link Active Reporting capability bit is implemented. Otherwise, this bit must be hardwired to 0b.	RO

Bit Location	Register Description	Attributes
14	**Link Bandwidth Management Status** – This bit is Set by hardware to indicate that either of the following has occurred without the Port transitioning through DL_Down status: • A Link retraining has completed following a write of 1b to the Retrain Link bit. Note: This bit is Set following any write of 1b to the Retrain Link bit, including when the Link is in the process of retraining for some other reason. • Hardware has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change. This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b. The default value of this bit is 0b.	RW1C
15	**Link Autonomous Bandwidth Status** – This bit is Set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was indicated as an autonomous change. This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b. The default value of this bit is 0b.	RW1C

7.8.9. Slot Capabilities Register (Offset 14h)

The Slot Capabilities register identifies PCI Express slot specific capabilities. Figure 7-19 details allocation of register fields in the Slot Capabilities register; Table 7-18 provides the respective bit definitions.

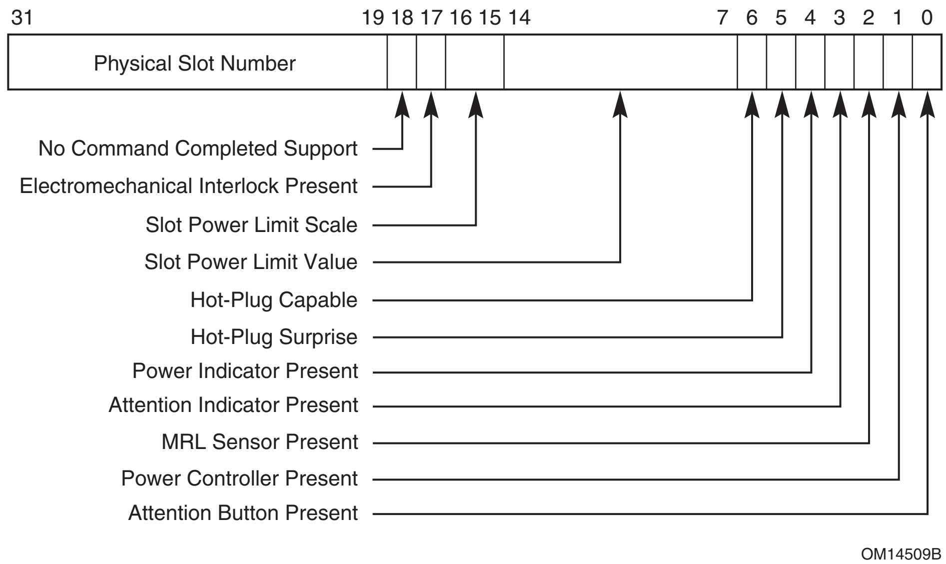

Figure 7-19: Slot Capabilities Register

Table 7-18: Slot Capabilities Register

Bit Location	Register Description	Attributes
0	**Attention Button Present** – When Set, this bit indicates that an Attention Button for this slot is electrically controlled by the chassis.	HwInit
1	**Power Controller Present** – When Set, this bit indicates that a software programmable Power Controller is implemented for this slot/adapter (depending on form factor).	HwInit
2	**MRL Sensor Present** – When Set, this bit indicates that an MRL Sensor is implemented on the chassis for this slot.	HwInit
3	**Attention Indicator Present** – When Set, this bit indicates that an Attention Indicator is electrically controlled by the chassis.	HwInit
4	**Power Indicator Present** – When Set, this bit indicates that a Power Indicator is electrically controlled by the chassis for this slot.	HwInit
5	**Hot-Plug Surprise** – When Set, this bit indicates that an adapter present in this slot might be removed from the system without any prior notification. This is a form factor specific capability. This bit is an indication to the operating system to allow for such removal without impacting continued software operation.	HwInit

Bit Location	Register Description	Attributes
6	**Hot-Plug Capable** – When Set, this bit indicates that this slot is capable of supporting hot-plug operations.	HwInit
14:7	**Slot Power Limit Value** – In combination with the Slot Power Limit Scale value, specifies the upper limit on power supplied by slot (see Section 6.9). Power limit (in Watts) calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field except when the Slot Power Limit Scale field equals 00b (1.0x) and Slot Power Limit Value exceeds EFh, the following alternative encodings are used: F0h = 250 W Slot Power Limit F1h = 275 W Slot Power Limit F2h = 300 W Slot Power Limit F3h to FFh = reserved This register must be implemented if the Slot Implemented bit is Set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. The default value prior to hardware/firmware initialization is 0000 0000b.	HwInit
16:15	**Slot Power Limit Scale** – Specifies the scale used for the Slot Power Limit Value (see Section 6.9). Range of Values: 00b = 1.0x 01b = 0.1x 10b = 0.01x 11b = 0.001x This register must be implemented if the Slot Implemented bit is Set. Writes to this register also cause the Port to send the Set_Slot_Power_Limit Message. The default value prior to hardware/firmware initialization is 00b.	HwInit
17	**Electromechanical Interlock Present** – When Set, this bit indicates that an Electromechanical Interlock is implemented on the chassis for this slot.	HwInit
18	**No Command Completed Support** – When Set, this bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller. This bit is only permitted to be Set if the hot-plug capable Port is able to accept writes to all fields of the Slot Control register without delay between successive writes.	HwInit

Bit Location	Register Description	Attributes
31:19	**Physical Slot Number** – This field indicates the physical slot number attached to this Port. This field must be hardware initialized to a value that assigns a slot number that is unique within the chassis, regardless of the form factor associated with the slot. This field must be initialized to zero for Ports connected to devices that are either integrated on the system board or integrated within the same silicon as the Switch device or Root Port.	HwInit

7.8.10. Slot Control Register (Offset 18h)

The Slot Control register controls PCI Express Slot specific parameters. Figure 7-20 details allocation of register fields in the Slot Control register; Table 7-19 provides the respective bit definitions.

Attention Indicator Control, Power Indicator Control, and Power Controller Control fields of the Slot Control register do not have a defined default value. If these fields are implemented, it is the responsibility of either system firmware or operating system software to (re)initialize these fields after a reset of the Link.

In hot-plug capable Downstream Ports, a write to the Slot Control register must cause a hot-plug command to be generated (see Section 6.7.3.2 for details on hot-plug commands). A write to the Slot Control register in a Downstream Port that is not hot-plug capable must not cause a hot-plug command to be executed.

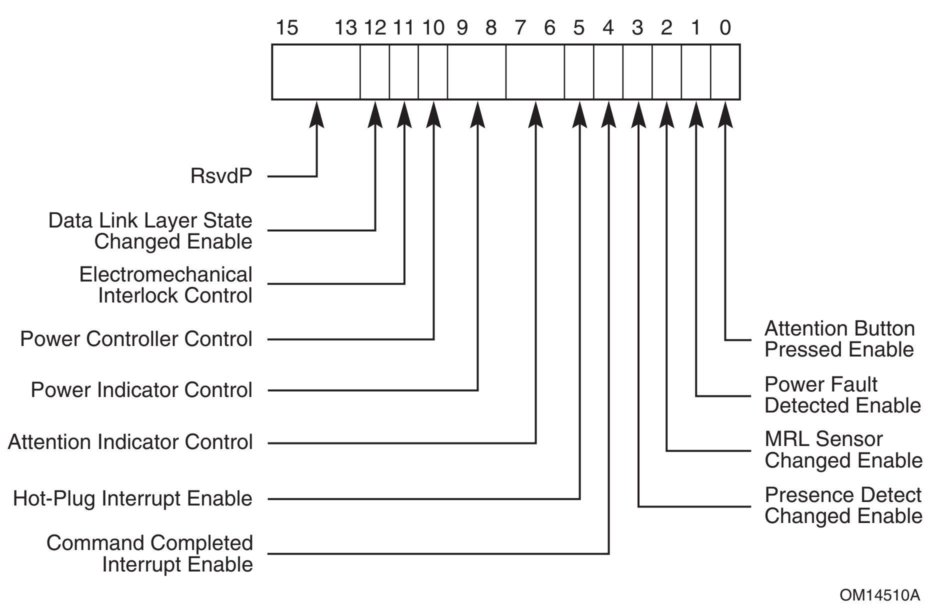

Figure 7-20: Slot Control Register

Table 7-19: Slot Control Register

Bit Location	Register Description	Attributes
0	**Attention Button Pressed Enable** – When Set to 1b, this bit enables software notification on an attention button pressed event (see Section 6.7.3). If the Attention Button Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	RW
1	**Power Fault Detected Enable** – When Set, this bit enables software notification on a power fault event (see Section 6.7.3). If a Power Controller that supports power fault detection is not implemented, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	RW
2	**MRL Sensor Changed Enable** – When Set, this bit enables software notification on a MRL sensor changed event (see Section 6.7.3). If the MRL Sensor Present bit in the Slot Capabilities register is Clear, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	RW
3	**Presence Detect Changed Enable** – When Set, this bit enables software notification on a presence detect changed event(see Section 6.7.3). If the Hot-Plug Capable bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	RW
4	**Command Completed Interrupt Enable** – If Command Completed notification is supported (if the No Command Completed Support bit in the Slot Capabilities register is 0b), when Set, this bit enables software notification when a hot-plug command is completed by the Hot-Plug Controller. If Command Completed notification is not supported, this bit must be hardwired to 0b. Default value of this bit is 0b.	RW
5	**Hot-Plug Interrupt Enable** – When Set, this bit enables generation of an interrupt on enabled hot-plug events. If the Hot Plug Capable bit in the Slot Capabilities register is Clear, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
7:6	**Attention Indicator Control** – If an Attention Indicator is implemented, writes to this field set the Attention Indicator to the Written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. Defined encodings are: 00b Reserved 01b On 10b Blink 11b Off Note: The default value of this field must be one of the non-Reserved values. If the Attention Indicator Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 00b.	RW
9:8	**Power Indicator Control** – If a Power Indicator is implemented, writes to this field set the Power Indicator to the Written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. Defined encodings are: 00b Reserved 01b On 10b Blink 11b Off Note: The default value of this field must be one of the non-Reserved values. If the Power Indicator Present bit in the Slot Capabilities register is 0b, this bit is permitted to be read-only with a value of 00b.	RW

Bit Location	Register Description	Attributes
10	**Power Controller Control** – If a Power Controller is implemented, this bit when written sets the power state of the slot per the defined encodings. Reads of this bit must reflect the value from the latest write, even if the corresponding hot-plug command is not complete, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. Depending on the form factor, the power is turned on/off either to the slot or within the adapter. Note that in some cases the power controller may autonomously remove slot power or not respond to a power-up request based on a detected fault condition, independent of the Power Controller Control setting. The defined encodings are: 0b Power On 1b Power Off If the Power Controller Implemented bit in the Slot Capabilities register is Clear, then writes to this bit have no effect and the read value of this bit is undefined.	RW
11	**Electromechanical Interlock Control** – If an Electromechanical Interlock is implemented, a write of 1b to this bit causes the state of the interlock to toggle. A write of 0b to this bit has no effect. A read of this bit always returns a 0b.	RW
12	**Data Link Layer State Changed Enable** – If the Data Link Layer Link Active Capability is implemented, when Set, this bit enables software notification when Data Link Layer Link Active Reporting bit is changed. If the Data Link Layer Link Active Reporting Capability is not implemented, this bit is permitted to be read-only with a value of 0b. Default value of this bit is 0b.	RW

7.8.11. Slot Status Register (Offset 1Ah)

The Slot Status register provides information about PCI Express Slot specific parameters. Figure 7-21 details allocation of register fields in the Slot Status register; Table 7-20 provides the respective bit definitions. Register fields for status bits not implemented by the device have the RsvdZ attribute.

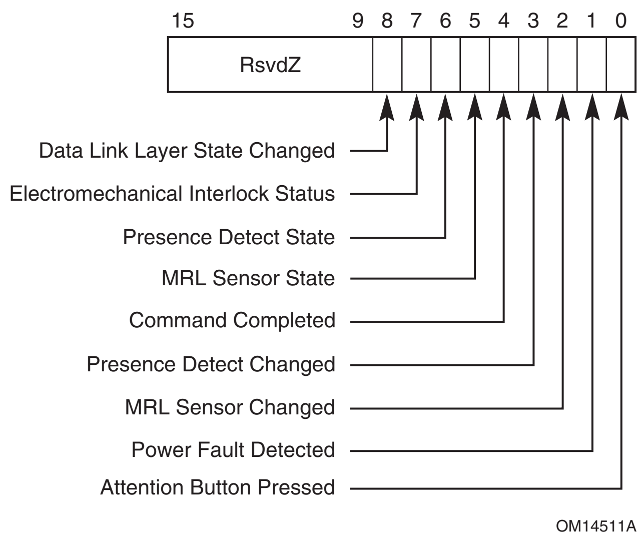

Figure 7-21: Slot Status Register

Table 7-20: Slot Status Register

Bit Location	Register Description	Attributes
0	**Attention Button Pressed** – If an Attention Button is implemented, this bit is Set when the attention button is pressed. If an Attention Button is not supported, this bit must not be Set.	RW1C
1	**Power Fault Detected** – If a Power Controller that supports power fault detection is implemented, this bit is Set when the Power Controller detects a power fault at this slot. Note that, depending on hardware capability, it is possible that a power fault can be detected at any time, independent of the Power Controller Control setting or the occupancy of the slot. If power fault detection is not supported, this bit must not be Set.	RW1C
2	**MRL Sensor Changed** – If an MRL sensor is implemented, this bit is Set when a MRL Sensor state change is detected. If an MRL sensor is not implemented, this bit must not be Set.	RW1C
3	**Presence Detect Changed** – This bit is set when the value reported in the Presence Detect State bit is changed.	RW1C

Bit Location	Register Description	Attributes
4	**Command Completed** – If Command Completed notification is supported (if the No Command Completed Support bit in the Slot Capabilities register is 0b), this bit is Set when a hot-plug command has completed and the Hot-Plug Controller is ready to accept a subsequent command. The Command Completed status bit is Set as an indication to host software that the Hot-Plug Controller has processed the previous command and is ready to receive the next command; it provides no guarantee that the action corresponding to the command is complete. If Command Completed notification is not supported, this bit must be hardwired to 0b.	RW1C
5	**MRL Sensor State** – This bit reports the status of the MRL sensor if implemented. Defined encodings are: 0b MRL Closed 1b MRL Open	RO
6	**Presence Detect State** – This bit indicates the presence of an adapter in the slot, reflected by the logical "OR" of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot's corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement a physical pin presence detect mechanism. Defined encodings are: 0b Slot Empty 1b Card Present in slot This bit must be implemented on all Downstream Ports that implement slots. For Downstream Ports not connected to slots (where the Slot Implemented bit of the PCI Express Capabilities register is 0b), this bit must be hardwired to 1b.	RO
7	**Electromechanical Interlock Status** – If an Electromechanical Interlock is implemented, this bit indicates the status of the Electromechanical Interlock. Defined encodings are: 0b Electromechanical Interlock Disengaged 1b Electromechanical Interlock Engaged	RO
8	**Data Link Layer State Changed** – This bit is Set when the value reported in the Data Link Layer Link Active bit of the Link Status register is changed. In response to a Data Link Layer State Changed event, software must read the Data Link Layer Link Active bit of the Link Status register to determine if the Link is active before initiating configuration cycles to the hot plugged device.	RW1C

IMPLEMENTATION NOTE

No Slot Power Controller

For slots that do not implement a power controller, software must ensure that system power planes are enabled to provide power to slots prior to reading presence detect state.

7.8.12. Root Control Register (Offset 1Ch)

The Root Control register controls PCI Express Root Complex specific parameters. Figure 7-22 details allocation of register fields in the Root Control register; Table 7-21 provides the respective bit definitions.

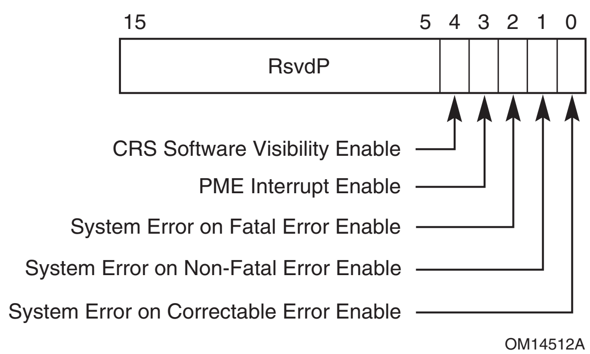

Figure 7-22: Root Control Register

Table 7-21: Root Control Register

Bit Location	Register Description	Attributes
0	**System Error on Correctable Error Enable** – If Set, this bit indicates that a System Error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with this Root Port, or by the Root Port itself. The mechanism for signaling a System Error to the system is system specific. Root Complex Event Collectors provide support for the above-described functionality for Root Complex Integrated Endpoints. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
1	**System Error on Non-Fatal Error Enable** – If Set, this bit indicates that a System Error should be generated if a Non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with this Root Port, or by the Root Port itself. The mechanism for signaling a System Error to the system is system specific. Root Complex Event Collectors provide support for the above-described functionality for Root Complex Integrated Endpoints. Default value of this bit is 0b.	RW
2	**System Error on Fatal Error Enable** – If Set, this bit indicates that a System Error should be generated if a Fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with this Root Port, or by the Root Port itself. The mechanism for signaling a System Error to the system is system specific. Root Complex Event Collectors provide support for the above-described functionality for Root Complex Integrated Endpoints. Default value of this bit is 0b.	RW
3	**PME Interrupt Enable** – When Set, this bit enables PME interrupt generation upon receipt of a PME Message as reflected in the PME Status bit (see Table 7-23). A PME interrupt is also generated if the PME Status bit is Set when this bit is changed from Clear to Set (see Section 6.1.5). Default value of this bit is 0b.	RW
4	**CRS Software Visibility Enable** – When Set, this bit enables the Root Port to return Configuration Request Retry Status (CRS) Completion Status to software (see Section 2.3.1). Root Ports that do not implement this capability must hardwire this bit to 0b. Default value of this bit is 0b.	RW

7.8.13. Root Capabilities Register (Offset 1Eh)

The Root Capabilities register identifies PCI Express Root Port specific capabilities. Figure 7-23 details allocation of register fields in the Root Capabilities register; Table 7-22 provides the respective bit definitions.

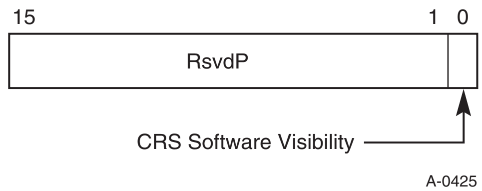

Figure 7-23: Root Capabilities Register

Table 7-22: Root Capabilities Register

Bit Location	Register Description	Attributes
0	**CRS Software Visibility** – When Set, this bit indicates that the Root Port is capable of returning Configuration Request Retry Status (CRS) Completion Status to software (see Section 2.3.1).	RO

7.8.14. Root Status Register (Offset 20h)

The Root Status register provides information about PCI Express device specific parameters. Figure 7-24 details allocation of register fields in the Root Status register; Table 7-23 provides the respective bit definitions.

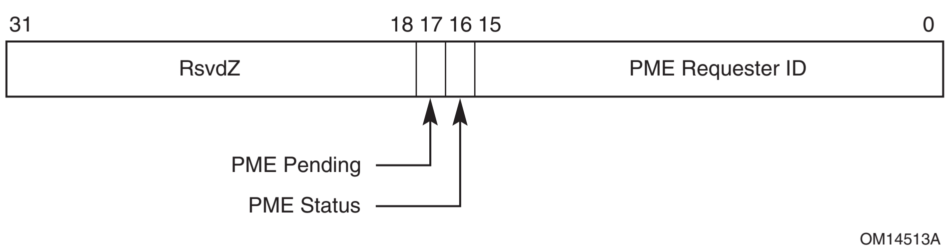

Figure 7-24: Root Status Register

Table 7-23: Root Status Register

Bit Location	Register Description	Attributes
15:0	**PME Requester ID** – This field indicates the PCI Requester ID of the last PME Requester. This field is only valid when the PME Status bit is Set.	RO
16	**PME Status** – This bit indicates that PME was asserted by the PME Requester indicated in the PME Requester ID field. Subsequent PMEs are kept pending until the status register is cleared by software by writing a 1b. Default value of this bit is 0b.	RW1C
17	**PME Pending** – This bit indicates that another PME is pending when the PME Status bit is Set. When the PME Status bit is cleared by software; the PME is delivered by hardware by setting the PME Status bit again and updating the PME Requester ID field appropriately. The PME Pending bit is cleared by hardware if no more PMEs are pending.	RO

7.8.15. Device Capabilities 2 Register (Offset 24h)

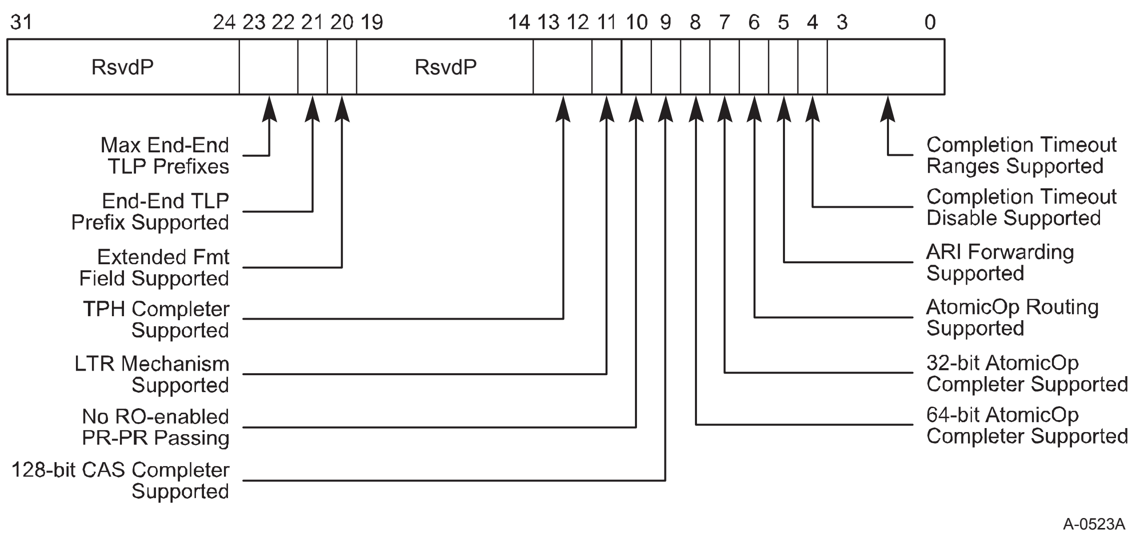

Figure 7-25: Device Capabilities 2 Register

Table 7-24: Device Capabilities 2 Register

Bit Location	Register Description	Attributes
3:0	**Completion Timeout Ranges Supported** – This field indicates device Function support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. This field is applicable only to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is reserved and must be hardwired to 0000b. Four time value ranges are defined: Range A: 50 µs to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4 s Range D: 4 s to 64 s Bits are set according to the table below to show timeout value ranges supported. 0000b Completion Timeout programming not supported – the Function must implement a timeout value in the range 50 µs to 50 ms. 0001b Range A 0010b Range B 0011b Ranges A and B 0110b Ranges B and C 0111b Ranges A, B, and C 1110b Ranges B, C and D 1111b Ranges A, B, C, and D All other values are reserved. It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms.	HwInit

Bit Location	Register Description	Attributes
4	**Completion Timeout Disable Supported** – A value of 1b indicates support for the Completion Timeout Disable mechanism. The Completion Timeout Disable mechanism is required for Endpoints that issue Requests on their own behalf and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. This mechanism is optional for Root Ports. For all other Functions this field is reserved and must be hardwired to 0b.	RO
5	**ARI Forwarding Supported** – Applicable only to Switch Downstream Ports and Root Ports; must be 0b for other Function types. This bit must be set to 1b if a Switch Downstream Port or Root Port supports this optional capability. See Section 6.13 for additional details.	RO
6	**AtomicOp Routing Supported** – Applicable only to Switch Upstream Ports, Switch Downstream Ports, and Root Ports; must be 0b for other Function types. This bit must be set to 1b if the Port supports this optional capability. See Section 6.15 for additional details.	RO
7	**32-bit AtomicOp Completer Supported** – Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. See Section 6.15.3.1 for additional RC requirements.	RO
8	**64-bit AtomicOp Completer Supported** – Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. See Section 6.15.3.1 for additional RC requirements.	RO
9	**128-bit CAS Completer Supported** – Applicable to Functions with Memory Space BARs as well as all Root Ports; must be 0b otherwise. This bit must be set to 1b if the Function supports this optional capability. See Section 6.15 for additional details.	RO
10	**No RO-enabled PR-PR Passing** – If this bit is Set, the routing element never carries out the passing permitted by Table 2-33 entry A2b that is associated with the Relaxed Ordering Attribute field being Set. This bit applies only for Switches and RCs that support peer-to-peer traffic between Root Ports. This bit applies only to Posted Requests being forwarded through the Switch or RC and does not apply to traffic originating or terminating within the Switch or RC itself. All Ports on a Switch or RC must report the same value for this bit. For all other functions, this bit must be 0b.	HwInit

Bit Location	Register Description	Attributes
11	**LTR Mechanism Supported** – A value of 1b indicates support for the optional Latency Tolerance Reporting (LTR) mechanism. Root Ports, Switches and Endpoints are permitted to implement this capability. For a multi-Function device associated with an Upstream Port, each Function must report the same value for this bit. For Bridges and other Functions that do not implement this capability, this bit must be hardwired to 0b.	RO
13:12	**TPH Completer Supported** – Applicable only to Root Ports and Endpoints; must be 00b for other Function types. Defined Encodings are: 00b TPH and Extended TPH Completer not supported. 01b TPH Completer supported; Extended TPH Completer not supported. 10b Reserved. 11 Both TPH and Extended TPH Completer supported. See Section 6.17 for details.	RO
20	**Extended Fmt Field Supported** – If Set, the Function supports the 3-bit definition of the Fmt field. If Clear, the Function supports a 2-bit definition of the Fmt field. See Section 2.2. Must be Set for Functions that support End-End TLP Prefixes. All Functions in an Upstream Port must have the same value for this bit. Each Downstream Port of a component may have a different value for this bit. It is strongly recommended that Functions support the 3-bit definition of the Fmt field.	RO
21	**End-End TLP Prefix Supported** – Indicates whether End-End TLP Prefix support is offered by a Function. Values are: 0b No Support 1b Support is provided to receive TLPs containing End-End TLP Prefixes. This bit is HwInit for Root Ports and is RO for all other Functions. All Ports of a Switch must have the same value for this bit.	RO/HwInit (see description)

Bit Location	Register Description	Attributes
23:22	**Max End-End TLP Prefixes –** Indicates the maximum number of End-End TLP Prefixes supported by this Function. TLPs received by this Function that contain more End-End TLP Prefixes than are supported must be handled as Malformed TLPs (see Section 2.2.10.2). Values are: 01b 1 End-End TLP Prefix 10b 2 End-End TLP Prefixes 11b 3 End-End TLP Prefixes 00b 4 End-End TLP Prefixes If End-End TLP Prefix Supported is Clear, this field is RsvdP. This field is HwInit for Root Ports and is RO for all other Functions. All Root Ports that have the End-End TLP Prefix Supported bit Set must contain the same value for this field. For Switches where End-End TLP Prefix Supported is Set, this field must be 00b indicating support for up to four End-End TLP Prefixes (see Section 2.2.10.2).	

IMPLEMENTATION NOTE

Use of the No RO-enabled PR-PR Passing Bit

The No RO-enabled PR-PR Passing bit allows platforms to utilize PCI-Express switching elements on the path between a requester and completer for requesters that could benefit from a slightly less relaxed ordering model. An example is a device that cannot ensure that multiple overlapping posted writes to the same address are outstanding at the same time. The method by which such a device is enabled to utilize this mode is out of scope of this specification.

7.8.16. Device Control 2 Register (Offset 28h)

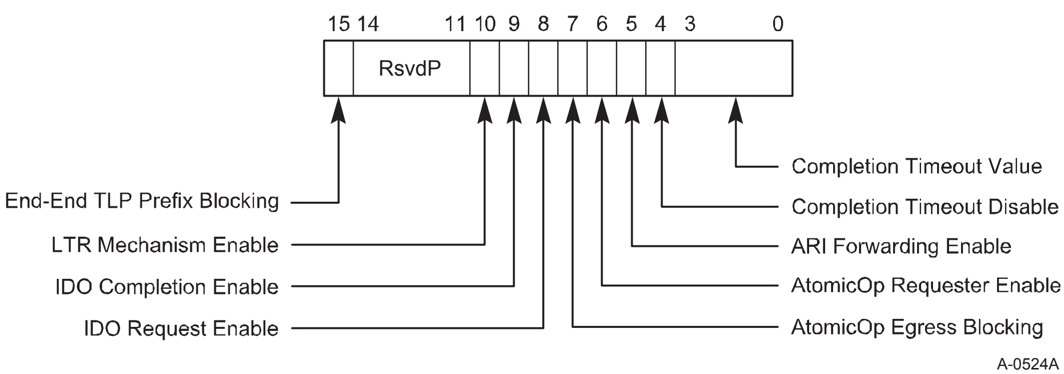

Figure 7-26: Device Control 2 Register

Table 7-25: Device Control 2 Register

Bit Location	Register Description	Attributes
3:0	**Completion Timeout Value** – In device Functions that support Completion Timeout programmability, this field allows system software to modify the Completion Timeout value. This field is applicable to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is reserved and must be hardwired to 0000b. A Function that does not support this optional capability must hardwire this field to 0000b and is required to implement a timeout value in the range 50 μs to 50 ms. Functions that support Completion Timeout programmability must support the values given below corresponding to the programmability ranges indicated in the Completion Timeout Ranges Supported field. Defined encodings: 0000b Default range: 50 μs to 50 ms It is strongly recommended that the Completion Timeout mechanism not expire in less than 10 ms. Values available if Range A (50 μs to 10 ms) programmability range is supported: 0001b 50 μs to 100 μs 0010b 1 ms to 10 ms Values available if Range B (10 ms to 250 ms) programmability range is supported: 0101b 16 ms to 55 ms 0110b 65 ms to 210 ms Values available if Range C (250 ms to 4 s) programmability range is supported: 1001b 260 ms to 900 ms 1010b 1 s to 3.5 s Values available if the Range D (4 s to 64 s) programmability range is supported: 1101b 4 s to 13 s 1110b 17 s to 64 s Values not defined above are reserved. Software is permitted to change the value in this field at any time. For Requests already pending when the Completion Timeout Value is changed, hardware is permitted to use either the new or the old value for the outstanding Requests, and is permitted to base the start time for each Request either on when this value was changed or on when each request was issued. The default value for this field is 0000b.	RW

Bit Location	Register Description	Attributes
4	**Completion Timeout Disable** – When Set, this bit disables the Completion Timeout mechanism. This bit is required for all Functions that support the Completion Timeout Disable Capability. Functions that do not support this optional capability are permitted to hardwire this bit to 0b Software is permitted to Set or Clear this bit at any time. When Set, the Completion Timeout detection mechanism is disabled. If there are outstanding Requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding Requests. If this is done, it is permitted to base the start time for each Request on either the time this bit was cleared or the time each Request was issued. The default value for this bit is 0b.	RW
5	**ARI Forwarding Enable** – When set, the Downstream Port disables its traditional Device Number field being 0 enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port. See Section 6.13. Default value of this bit is 0b. Must be hardwired to 0b if the ARI Forwarding Supported bit is 0b.	RW
6	**AtomicOp Requester Enable** – Applicable only to Endpoints and Root Ports; must be hardwired to 0b for other Function types. The Function is allowed to initiate AtomicOp Requests only if this bit and the Bus Master Enable bit in the Command register are both Set. This bit is required to be RW if the Endpoint or Root Port is capable of initiating AtomicOp Requests, but otherwise is permitted to be hardwired to 0b. This bit does not serve as a capability bit. This bit is permitted to be RW even if no AtomicOp Requester capabilities are supported by the Endpoint or Root Port. Default value of this bit is 0b.	RW
7	**AtomicOp Egress Blocking** – Applicable and mandatory for Switch Upstream Ports, Switch Downstream Ports, and Root Ports that implement AtomicOp routing capability; otherwise must be hardwired to 0b. When this bit is Set, AtomicOp Requests that target going out this Egress Port must be blocked. See Section 6.15.2. Default value of this bit is 0b.	RW
8	**IDO Request Enable** – If this bit is Set, the Function is permitted to set the ID-Based Ordering (IDO) bit (Attribute[2]) of Requests it initiates (see Section 2.2.6.3 and Section 2.4). Endpoints, including RC Integrated Endpoints, and Root Ports are permitted to implement this capability. A Function is permitted to hardwire this bit to 0b if it never sets the IDO attribute in Requests. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
9	**IDO Completion Enable** – If this bit is Set, the Function is permitted to set the ID-Based Ordering (IDO) bit (Attribute[2]) of Completions it returns (see Section 2.2.6.3 and Section 2.4). Endpoints, including RC Integrated Endpoints, and Root Ports are permitted to implement this capability. A Function is permitted to hardwire this bit to 0b if it never sets the IDO attribute in Requests. Default value of this bit is 0b.	RW
10	**LTR Mechanism Enable** – When Set to 1b, this bit enables the Latency Tolerance Reporting (LTR) mechanism. For a Multi-Function device associated with an Upstream Port of a device that implements LTR, the bit in Function 0 is RW, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is RsvdP. Functions that do not implement the LTR mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status.	RW/RsvdP
15	**End-End TLP Prefix Blocking** – Controls whether the routing function is permitted to forward TLPs containing an End-End TLP Prefix. Values are: 0b Forwarding Enabled – Function is permitted to send TLPs with End-End TLP Prefixes. 1b Forwarding Blocked – Function is not permitted to send TLPs with End-End TLP Prefixes. This bit affects TLPs that exit the Switch/Root Complex using the associated Port. It does not affect TLPs forwarded internally within the Switch/Root Complex. It does not affect TLPs that enter through the associated Port, that originate in the associated Port or originate in a Root Complex Integrated Device integrated with the associated Port. As described in Section 2.2.10.2, blocked TLPs are reported by the TLP Prefix Blocked Error. The default value of this bit is 0b. This bit is hardwired to 1b in Root Ports that support End-End TLP Prefixes but do not support forwarding of End-End TLP Prefixes. This bit is applicable to Root Ports and Switch Ports where the End-End TLP Prefix Supported bit is Set. This bit is not applicable and is RsvdP in all other cases.	RW (see description)

7.8.17. Device Status 2 Register (Offset 2Ah)

This section is a placeholder. There are no capabilities that require this register.

This register must be treated by software as RsvdZ.

7.8.18. Link Capabilities 2 Register (Offset 2Ch)

This section is a placeholder. There are no capabilities that require this register.

This register must be treated by software as RsvdP.

7.8.19. Link Control 2 Register (Offset 30h)

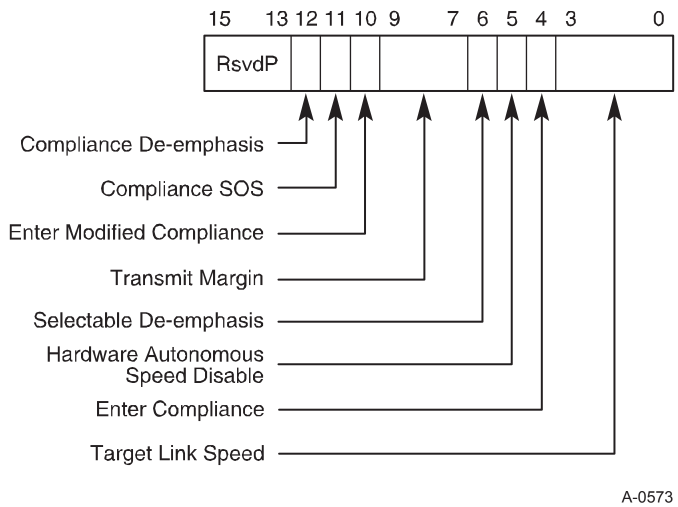

Figure 7-27: Link Control 2 Register

Table 7-26: Link Control 2 Register

Bit Location	Register Description	Attributes
3:0	**Target Link Speed** – For Downstream Ports, this field sets an upper limit on Link operational speed by restricting the values advertised by the Upstream component in its training sequences. Defined encodings are: 0001b 2.5 GT/s Target Link Speed 0010b 5.0 GT/s Target Link Speed All other encodings are reserved. If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, the result is undefined. The default value of this field is the highest Link speed supported by the component (as reported in the Supported Link Speeds field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value. For both Upstream and Downstream Ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode. For a Multi-Function device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. Components that support only the 2.5 GT/s speed are permitted to hardwire this field to 0000b.	RWS/RsvdP (see description)
4	**Enter Compliance** – Software is permitted to force a Link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a Link and then initiating a hot reset on the Link. Default value of this bit following Fundamental Reset is 0b. For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b.	RWS/RsvdP (see description)

Bit Location	Register Description	Attributes
5	**Hardware Autonomous Speed Disable** – When Set, this bit disables hardware from changing the Link speed for device-specific reasons other than attempting to correct unreliable Link operation by reducing Link speed. Initial transition to the highest supported common link speed is not blocked by this bit. For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RWS/RsvdP (see description)
6	**Selectable De-emphasis** – When the Link is operating at 5.0 GT/s speed, this bit is used to control the transmit de-emphasis of the link in specific situations. See Section 4.2.6 for detailed usage information. Encodings: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. This bit is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.	HwInit
9:7	**Transmit Margin** – This field controls the value of the non-deemphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see Chapter 4 for details of how the Transmitter voltage level is determined in various states). Encodings: 000b Normal operating range 001b-111b As defined in Section 4.3.3.3, not all encodings are required to be implemented. For a Multi-Function device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. Default value of this field is 000b. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 000b. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.	RWS/RsvdP (see description)

Bit Location	Register Description	Attributes
10	**Enter Modified Compliance** – When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. Default value of this bit is 0b. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.	RWS/RsvdP (see description)
11	**Compliance SOS** – When set to 1b, the LTSSM is required to send SKP Ordered Sets between sequences when sending the Compliance Pattern or Modified Compliance Pattern. For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. The default value of this bit is 0b. Components that support only the 2.5 GT/s speed are permitted to hardwire this field to 0b.	RWS/RsvdP (see description)
12	**Compliance De-emphasis** – This bit sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Encodings: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s, the setting of this bit has no effect. Components that support only 2.5 GT/s speed are permitted to hardwire this bit to 0b. For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this bit is of type RsvdP. The default value of this bit is 0b. This bit is intended for debug, compliance testing purposes. System firmware and software is allowed to modify this bit only during debug or compliance testing.	RWS/RsvdP (see description)

IMPLEMENTATION NOTE

Selectable De-emphasis Usage

Selectable De-emphasis setting is applicable only to Root Ports and Switch Downstream Ports. The De-emphasis setting is implementation specific and depends on the platform or enclosure in which the Root Port or the Switch Downstream Port is located. System firmware or hardware strapping is used to configure the selectable de-emphasis value. In cases where system firmware cannot be used to set the de-emphasis value (for example, a hot plugged Switch), hardware strapping must be used to set the de-emphasis value.

7.8.20. Link Status 2 Register (Offset 32h)

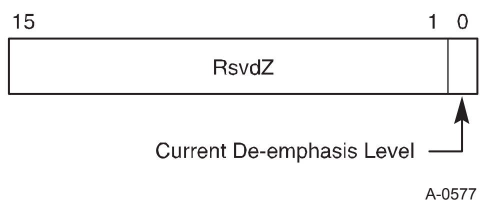

Figure 7-28: Link Status 2 Register

Table 7-27: Link Status 2 Register

Bit Location	Register Description	Attributes
0	**Current De-emphasis Level** – When the Link is operating at 5 GT/s speed, this bit reflects the level of de-emphasis. Encodings: 1b -3.5 dB 0b -6 dB The value in this bit is undefined when the Link is operating at 2.5 GT/s speed. Components that support only the 2.5 GT/s speed are permitted to hardwire this field to 0b.	RO

7.8.21. Slot Capabilities 2 Register (Offset 34h)

This section is a placeholder. There are no capabilities that require this register.

This register must be treated by software as RsvdP.

7.8.22. Slot Control 2 Register (Offset 38h)

This section is a placeholder. There are no capabilities that require this register.

This register must be treated by software as RsvdP.

7.8.23. Slot Status 2 Register (Offset 3Ah)

This section is a placeholder. There are no capabilities that require this register.

This register must be treated by software as RsvdZ.

7.9. PCI Express Extended Capabilities

PCI Express Extended Capability registers are located in Configuration Space at offsets 256 or greater as shown in Figure 7-29 or in the Root Complex Register Block (RCRB). These registers when located in the Configuration Space are accessible using only the PCI Express Extended Configuration Space flat memory-mapped access mechanism.

PCI Express Extended Capability structures are allocated using a linked list of optional or required PCI Express Extended Capabilities following a format resembling PCI Capability structures. The first DWORD of the Capability structure identifies the Capability and version and points to the next Capability as shown in Figure 7-29.

Each Capability structure must be DWORD aligned.

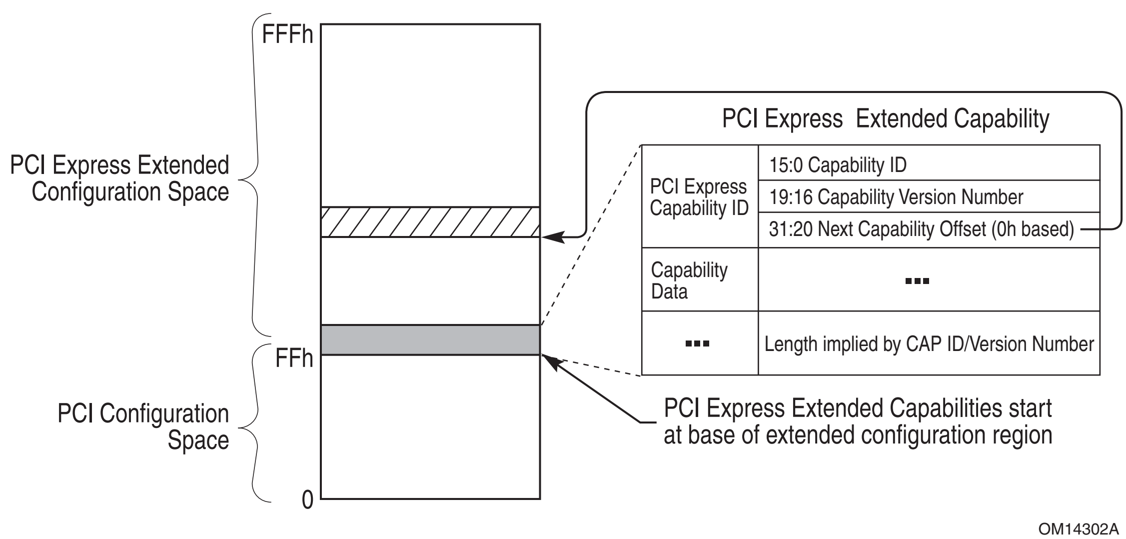

Figure 7-29: PCI Express Extended Configuration Space Layout

7.9.1. Extended Capabilities in Configuration Space

Extended Capabilities in Configuration Space always begin at offset 100h with a PCI Express Extended Capability header (Section 7.9.3). Absence of any Extended Capabilities is required to be indicated by an Extended Capability header with a Capability ID of 0000h, a Capability Version of 0h, and a Next Capability Offset of 0h.

7.9.2. Extended Capabilities in the Root Complex Register Block

Extended Capabilities in a Root Complex Register Block always begin at offset 0h with a PCI Express Extended Capability header (Section 7.9.3). Absence of any Extended Capabilities is required to be indicated by an Extended Capability header with a Capability ID of FFFFh and a Next Capability Offset of 0h.

7.9.3. PCI Express Extended Capability Header

All PCI Express Extended Capabilities must begin with a PCI Express Extended Capability header. Figure 7-30 details the allocation of register fields of a PCI Express Extended Capability header; Table 7-28 provides the respective bit definitions.

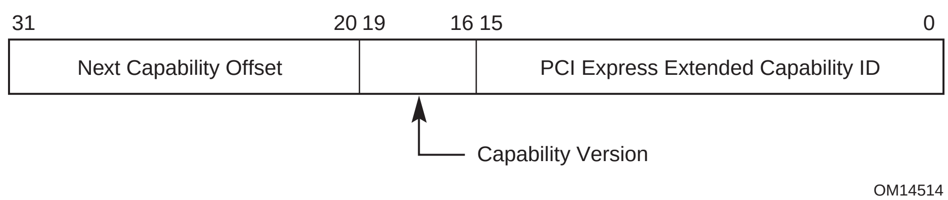

Figure 7-30: PCI Express Extended Capability Header

Table 7-28: PCI Express Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. A version of the specification that changes the Extended Capability in a way that is not otherwise identifiable (e.g., through a new Capability field) is permitted to increment this field. All such changes to the Capability structure must be software-compatible. Software must check for Capability Version numbers that are greater than or equal to the highest number defined when the software is written, as Functions reporting any such Capability Version numbers will contain a Capability structure that is compatible with that piece of software.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits.	RO

7.10. Advanced Error Reporting Capability

The PCI Express Advanced Error Reporting Capability is an optional Extended Capability that may be implemented by PCI Express device Functions supporting advanced error control and reporting. The Advanced Error Reporting Capability structure definition has additional interpretation for Root Ports and Root Complex Event Collectors; software must interpret the Device/Port Type field in the PCI Express Capabilities register to determine the availability of additional registers for Root Ports and Root Complex Event Collectors.

Figure 7-31 shows the PCI Express Advanced Error Reporting Capability structure.

Note that if an error reporting bit field is marked as optional in the error registers, the bits must be implemented or not implemented as a group across the Status, Mask and Severity registers. In other words, a Function is required to implement the same error bit fields in corresponding Status, Mask and Severity registers. Bits corresponding to bit fields that are not implemented must be hardwired to 0, unless otherwise specified.

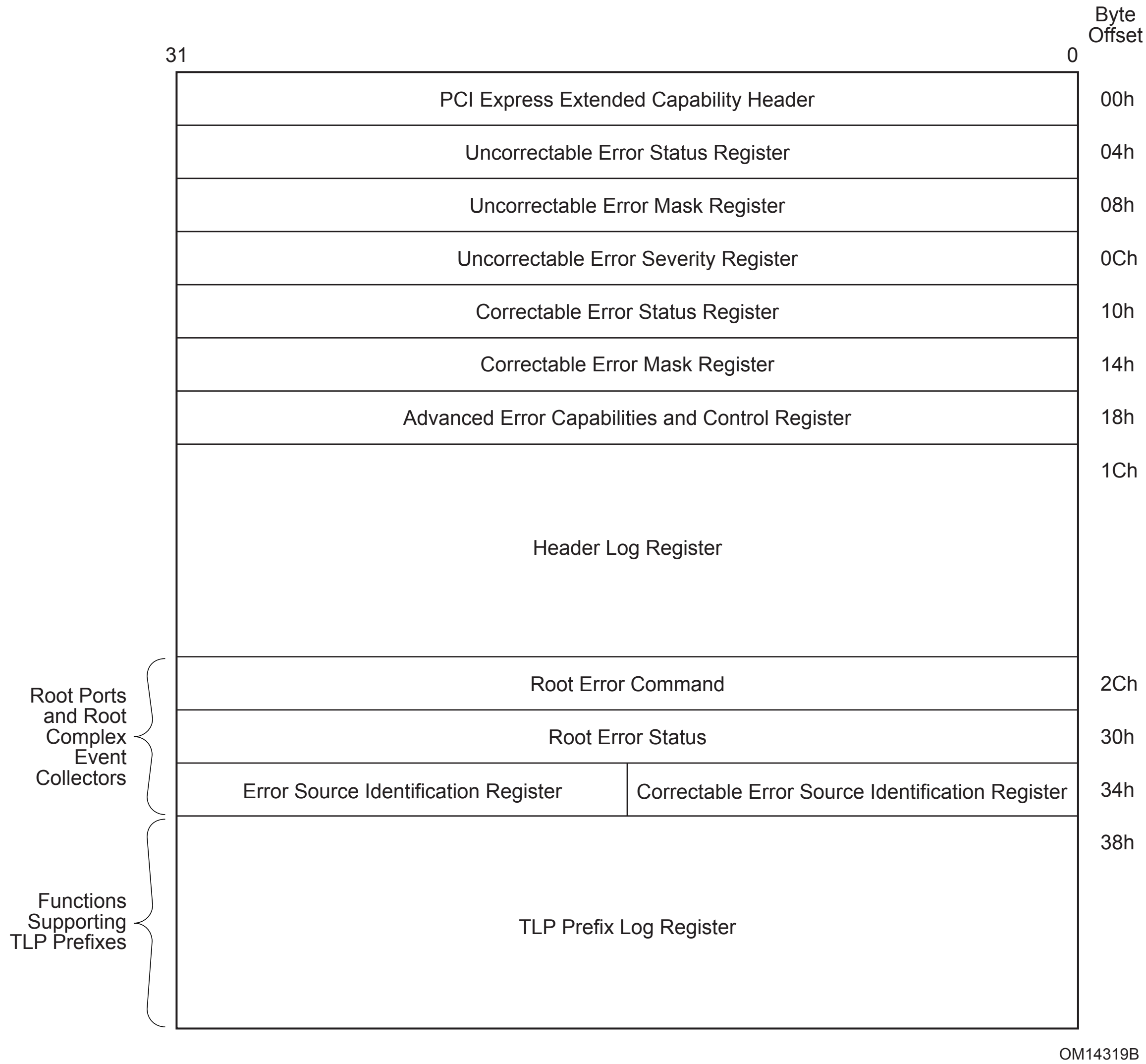

Figure 7-31: PCI Express Advanced Error Reporting Extended Capability Structure

7.10.1. Advanced Error Reporting Extended Capability Header (Offset 00h)

Figure 7-32 details the allocation of register fields of an Advanced Error Reporting Extended Capability header; Table 7-29 provides the respective bit definitions.

Refer to Section 7.9.3 for a description of the PCI Express Extended Capability header. The Extended Capability ID for the Advanced Error Reporting Capability is 0001h.

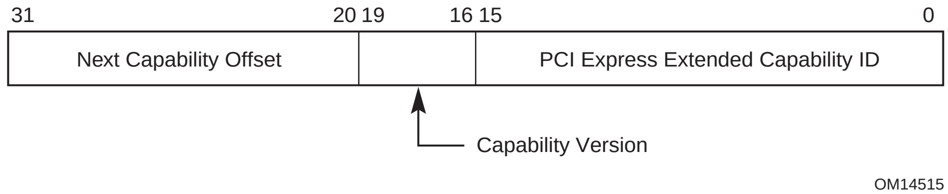

Figure 7-32: Advanced Error Reporting Extended Capability Header

Table 7-29: Advanced Error Reporting Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Advanced Error Reporting Capability is 0001h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 2h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.10.2. Uncorrectable Error Status Register (Offset 04h)

The Uncorrectable Error Status register indicates error detection status of individual errors on a PCI Express device Function. An individual error status bit that is Set indicates that a particular error was detected; software may clear an error status by writing a 1b to the respective bit. Refer to Section 6.2 for further details. Register bits not implemented by the Function are hardwired to 0b. Figure 7-33 details the allocation of register fields of the Uncorrectable Error Status register; Table 7-30 provides the respective bit definitions.

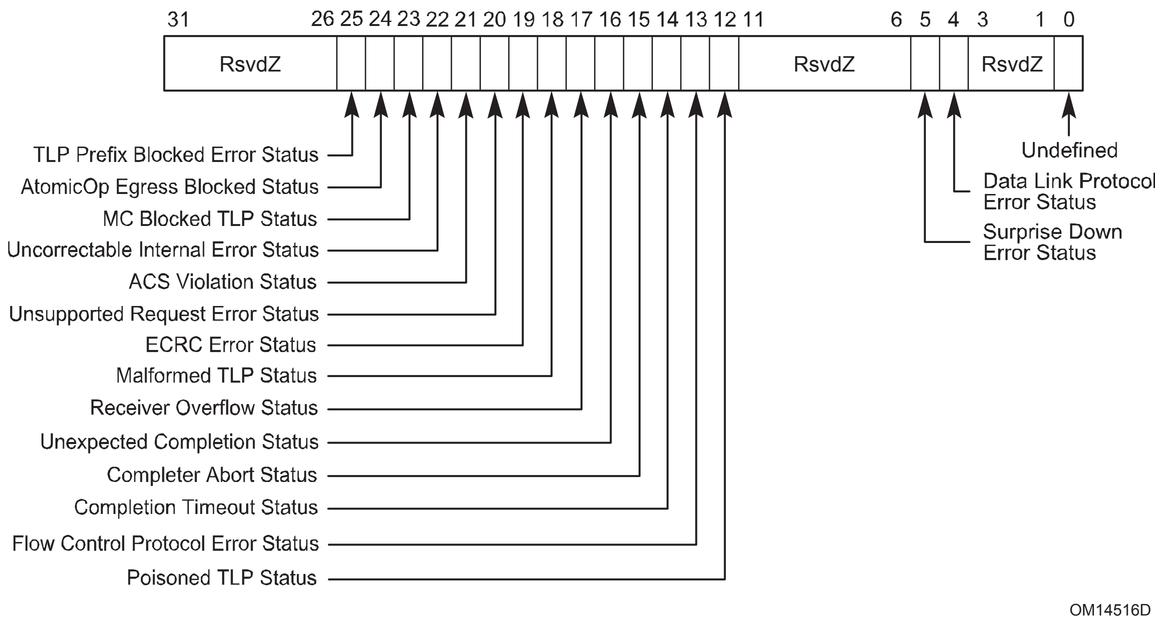

Figure 7-33: Uncorrectable Error Status Register

Table 7-30: Uncorrectable Error Status Register

Bit Location	Register Description	Attributes	Default
0	**Undefined** – The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate a Link Training Error. System software must ignore the value read from this bit. System software is permitted to write any value to this bit.	Undefined	Undefined
4	**Data Link Protocol Error Status**	RW1CS	0b
5	**Surprise Down Error Status** (Optional)	RW1CS	0b
12	**Poisoned TLP Status**	RW1CS	0b
13	**Flow Control Protocol Error Status** (Optional)	RW1CS	0b
14	**Completion Timeout Status**[98]	RW1CS	0b
15	**Completer Abort Status** (Optional)	RW1CS	0b
16	**Unexpected Completion Status**	RW1CS	0b
17	**Receiver Overflow Status** (Optional)	RW1CS	0b
18	**Malformed TLP Status**	RW1CS	0b
19	**ECRC Error Status** (Optional)	RW1CS	0b
20	**Unsupported Request Error Status**	RW1CS	0b

[98] For Switch Ports, required if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such Requests generated by other devices). If the Switch Port does not issue such Requests, then the Completion Timeout mechanism is not applicable and this bit must be hardwired to 0b.

Bit Location	Register Description	Attributes	Default
21	**ACS Violation Status** (Optional)	RW1CS	0b
22	**Uncorrectable Internal Error Status** (Optional)	RW1CS	0b
23	**MC Blocked TLP Status** (Optional)	RW1CS	0b
24	**AtomicOp Egress Blocked Status** (Optional)	RW1CS	0b
25	**TLP Prefix Blocked Error Status** (Optional)	RW1CS	0b

7.10.3. Uncorrectable Error Mask Register (Offset 08h)

The Uncorrectable Error Mask register controls reporting of individual errors by the device Function to the PCI Express Root Complex via a PCI Express error Message. A masked error (respective bit Set in the mask register) is not recorded or reported in the Header Log register, does not update the First Error Pointer, and is not reported to the PCI Express Root Complex by this Function. Refer to Section 6.2 for further details. There is a mask bit per error bit of the Uncorrectable Error Status register. Register fields for bits not implemented by the Function are hardwired to 0b. Figure 7-34 details the allocation of register fields of the Uncorrectable Error Mask register; Table 7-31 provides the respective bit definitions.

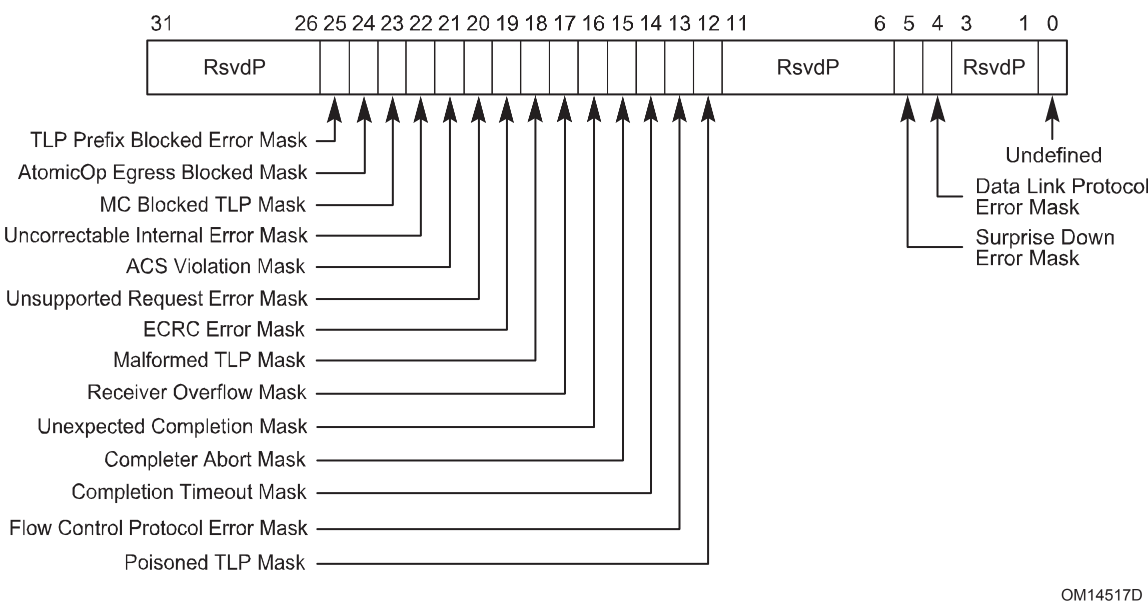

Figure 7-34: Uncorrectable Error Mask Register

Table 7-31: Uncorrectable Error Mask Register

Bit Location	Register Description	Attributes	Default
0	**Undefined** – The value read from this bit is undefined. In previous versions of this specification, this bit was used to mask a Link Training Error. System software must ignore the value read from this bit. System software must only write a value of 1b to this bit.	Undefined	Undefined
4	**Data Link Protocol Error Mask**	RWS	0b
5	**Surprise Down Error Mask** (Optional)	RWS	0b
12	**Poisoned TLP Mask**	RWS	0b
13	**Flow Control Protocol Error Mask** (Optional)	RWS	0b
14	**Completion Timeout Mask**[99]	RWS	0b
15	**Completer Abort Mask** (Optional)	RWS	0b
16	**Unexpected Completion Mask**	RWS	0b
17	**Receiver Overflow Mask** (Optional)	RWS	0b
18	**Malformed TLP Mask**	RWS	0b
19	**ECRC Error Mask** (Optional)	RWS	0b
20	**Unsupported Request Error Mask**	RWS	0b
21	**ACS Violation Mask** (Optional)	RWS	0b
22	**Uncorrectable Internal Error Mask** (Optional)	RWS	1b
23	**MC Blocked TLP Mask** (Optional)	RWS	0b
24	**AtomicOp Egress Blocked Mask** (Optional)	RWS	0b
25	**TLP Prefix Blocked Error Mask** (Optional)	RWS	0b

[99] For Switch Ports, required if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such Requests generated by other devices). If the Switch Port does not issue such Requests, then the Completion Timeout mechanism is not applicable and this bit must be hardwired to 0b.

7.10.4. Uncorrectable Error Severity Register (Offset 0Ch)

The Uncorrectable Error Severity register controls whether an individual error is reported as a Non-fatal or Fatal error. An error is reported as fatal when the corresponding error bit in the severity register is Set. If the bit is Clear, the corresponding error is considered non-fatal. Refer to Section 6.2 for further details. Register fields for bits not implemented by the Function are hardwired to the specified default value. Figure 7-35 details the allocation of register fields of the Uncorrectable Error Severity register; Table 7-32 provides the respective bit definitions.

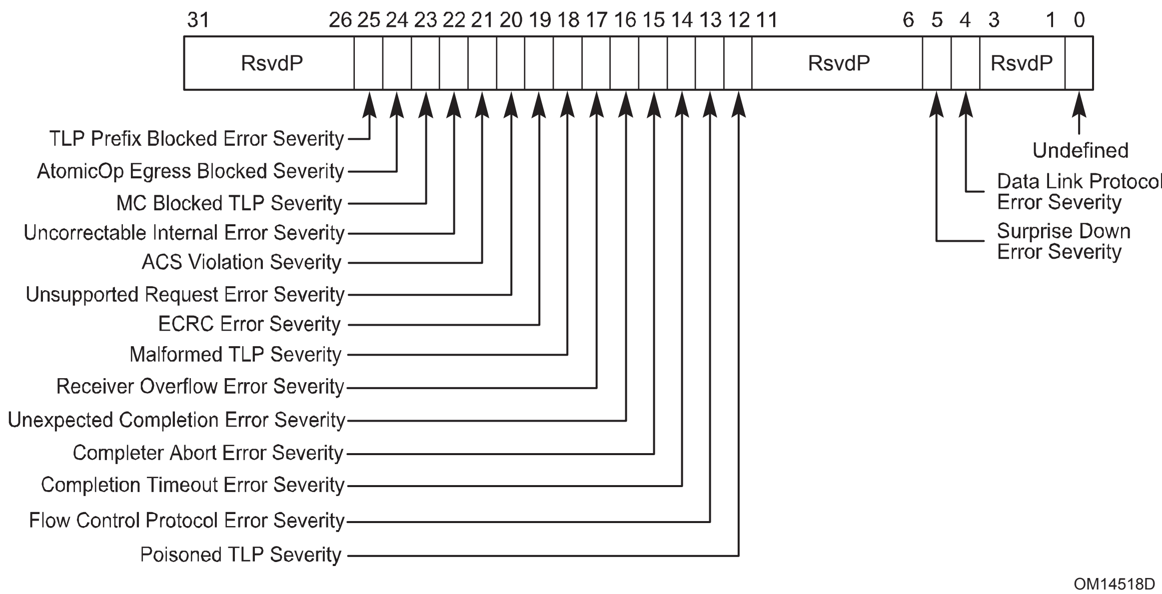

Figure 7-35: Uncorrectable Error Severity Register

Table 7-32: Uncorrectable Error Severity Register

Bit Location	Register Description	Attributes	Default
0	**Undefined** – The value read from this bit is undefined. In previous versions of this specification, this bit was used to Set the severity of a Link Training Error. System software must ignore the value read from this bit. System software is permitted to write any value to this bit.	Undefined	Undefined
4	**Data Link Protocol Error Severity**	RWS	1b
5	**Surprise Down Error Severity** (Optional)	RWS	1b
12	**Poisoned TLP Severity**	RWS	0b
13	**Flow Control Protocol Error Severity** (Optional)	RWS	1b
14	**Completion Timeout Error Severity**[100]	RWS	0b
15	**Completer Abort Error Severity** (Optional)	RWS	0b
16	**Unexpected Completion Error Severity**	RWS	0b
17	**Receiver Overflow Error Severity** (Optional)	RWS	1b
18	**Malformed TLP Severity**	RWS	1b
19	**ECRC Error Severity** (Optional)	RWS	0b
20	**Unsupported Request Error Severity**	RWS	0b
21	**ACS Violation Severity** (Optional)	RWS	0b
22	**Uncorrectable Internal Error Severity** (Optional)	RWS	1b
23	**MC Blocked TLP Severity** (Optional)	RWS	0b
24	**AtomicOps Egress Blocked Severity** (Optional)	RWS	0b
25	**TLP Prefix Blocked Error Severity** (Optional)	RWS	0b

[100] For Switch Ports, required if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such Requests generated by other devices). If the Switch Port does not issue such Requests, then the Completion Timeout mechanism is not applicable and this bit must be hardwired to 0b.

7.10.5. Correctable Error Status Register (Offset 10h)

The Correctable Error Status register reports error status of individual correctable error sources on a PCI Express device Function. When an individual error status bit is Set, it indicates that a particular error occurred; software may clear an error status by writing a 1b to the respective bit. Refer to Section 6.2 for further details. Register bits not implemented by the Function are hardwired to 0b. Figure 7-36 details the allocation of register fields of the Correctable Error Status register; Table 7-33 provides the respective bit definitions.

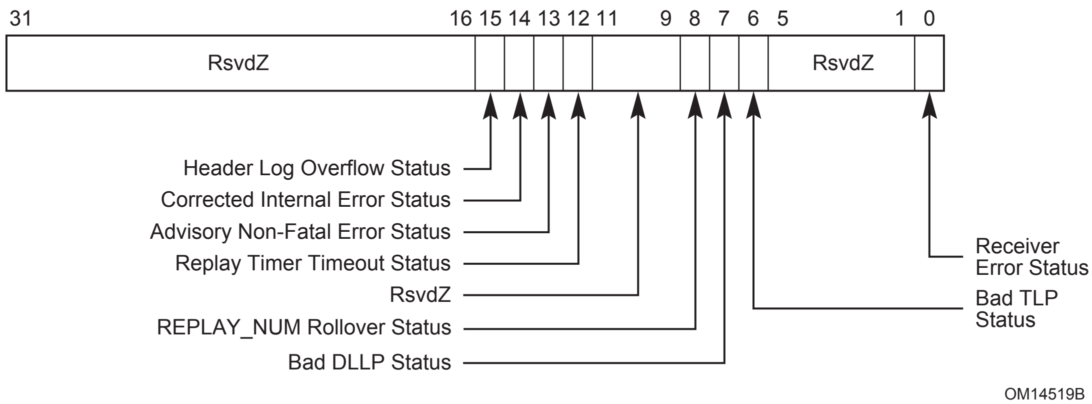

Figure 7-36: Correctable Error Status Register

Table 7-33: Correctable Error Status Register

Bit Location	Register Description	Attributes	Default
0	**Receiver Error Status**[101]	RW1CS	0b
6	**Bad TLP Status**	RW1CS	0b
7	**Bad DLLP Status**	RW1CS	0b
8	**REPLAY_NUM Rollover Status**	RW1CS	0b
12	**Replay Timer Timeout Status**	RW1CS	0b
13	**Advisory Non-Fatal Error Status**	RW1CS	0b
14	**Corrected Internal Error Status** (Optional)	RW1CS	0b
15	**Header Log Overflow Status** (Optional)	RW1CS	0b

[101] For historical reasons, implementation of this bit is optional. If not implemented, this bit must be RsvdZ, and bit 0 of the Correctable Error Mask Register must also not be implemented. Note that some checking for Receiver Errors is required in all cases (see Sections 4.2.1.3, 4.2.4.6, and 4.2.6).

7.10.6. Correctable Error Mask Register (Offset 14h)

The Correctable Error Mask register controls reporting of individual correctable errors by this Function to the PCI Express Root Complex via a PCI Express error Message. A masked error (respective bit Set in the mask register) is not reported to the PCI Express Root Complex by this Function. Refer to Section 6.2 for further details. There is a mask bit per error bit in the Correctable Error Status register. Register fields for bits not implemented by the Function are hardwired to 0b. Figure 7-37 details the allocation of register fields of the Correctable Error Mask register; Table 7-34 provides the respective bit definitions.

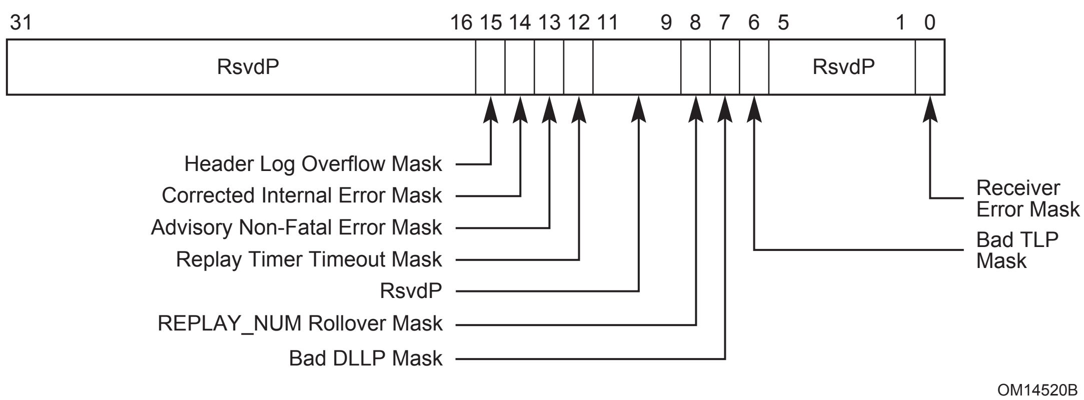

Figure 7-37: Correctable Error Mask Register

Table 7-34: Correctable Error Mask Register

Bit Location	Register Description	Attributes	Default
0	**Receiver Error Mask**[102]	RWS	0b
6	**Bad TLP Mask**	RWS	0b
7	**Bad DLLP Mask**	RWS	0b
8	**REPLAY_NUM Rollover Mask**	RWS	0b
12	**Replay Timer Timeout Mask**	RWS	0b
13	**Advisory Non-Fatal Error Mask** – This bit is Set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.	RWS	1b
14	**Corrected Internal Error Mask** (Optional)	RWS	1b
15	**Header Log Overflow Mask** (Optional)	RWS	1b

[102] For historical reasons, implementation of this bit is optional. If not implemented, this bit must be RsvdP, and bit 0 of the Correctable Error Status Register must also not be implemented. Note that some checking for Receiver Errors is required in all cases (see Sections 4.2.1.3, 4.2.4.6, and 4.2.6).

7.10.7. Advanced Error Capabilities and Control Register (Offset 18h)

Figure 7-38 details allocation of register fields in the Advanced Error Capabilities and Control register; Table 7-35 provides the respective bit definitions. Handling of multiple errors is discussed in Section 6.2.4.2.

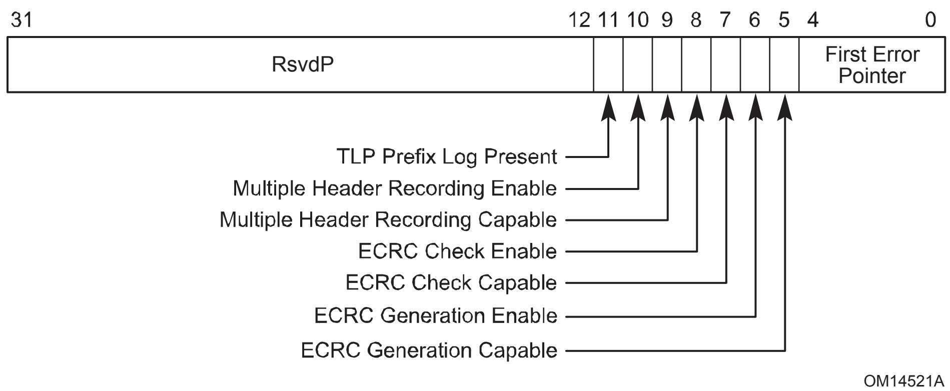

Figure 7-38: Advanced Error Capabilities and Control Register

Table 7-35: Advanced Error Capabilities and Control Register

Bit Location	Register Description	Attributes
4:0	**First Error Pointer** – The First Error Pointer is a field that identifies the bit position of the first error reported in the Uncorrectable Error Status register. Refer to Section 6.2 for further details.	ROS
5	**ECRC Generation Capable** – If Set, this bit indicates that the Function is capable of generating ECRC (see Section 2.7).	RO
6	**ECRC Generation Enable** – When Set, ECRC generation is enabled (see Section 2.7). Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RWS
7	**ECRC Check Capable** – If Set, this bit indicates that the Function is capable of checking ECRC (see Section 2.7).	RO
8	**ECRC Check Enable** – When Set, ECRC checking is enabled (see Section 2.7). Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RWS

Bit Location	Register Description	Attributes
9	**Multiple Header Recording Capable** – If Set, this bit indicates that the Function is capable of recording more than one error header. Refer to Section 6.2 for further details.	RO
10	**Multiple Header Recording Enable** – When Set, this bit enables the Function to record more than one error header. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b.	RWS
11	**TLP Prefix Log Present** – If Set and the First Error Pointer is valid, indicates that the TLP Prefix Log register contains valid information. If Clear or if First Error Pointer is invalid, the TLP Prefix Log register is undefined. Default value of this bit is 0. This bit is RsvdP if the End-End TLP Prefix Supported bit is Clear.	ROS

7.10.8. Header Log Register (Offset 1Ch)

The Header Log register contains the header for the TLP corresponding to a detected error; refer to Section 6.2 for further details. Section 6.2 also describes the conditions where the packet header is recorded. This register is 16 bytes and adheres to the format of the headers defined throughout this specification.

The header is captured such that, when read using DW accesses, the fields of the header are laid out in the same way the headers are presented in this document. Therefore, byte 0 of the header is located in byte 3 of the Header Log register, byte 1 of the header is in byte 2 of the Header Log register and so forth. For 12-byte headers, only bytes 0 through 11 of the Header Log register are used and values in bytes 12 through 15 are undefined.

In certain cases where a Malformed TLP is reported, the Header Log Register may contain TLP Prefix information. See Section 6.2.4.4 for details.

Figure 7-39 details allocation of register fields in the Header Log register; Table 7-36 provides the respective bit definitions.

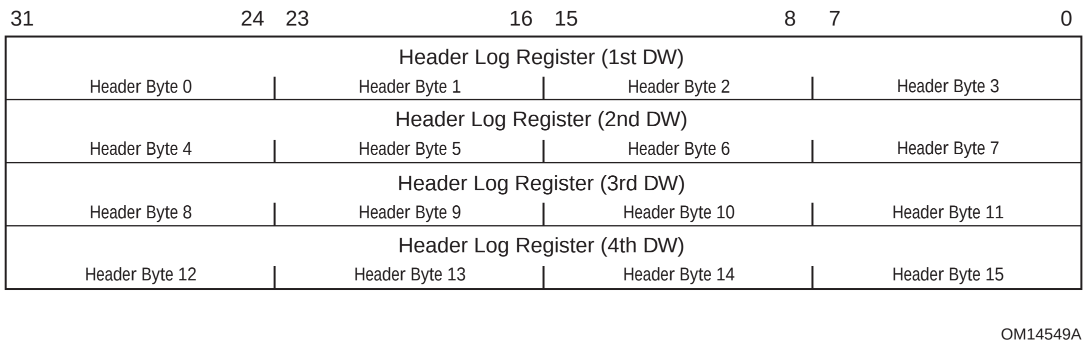

Figure 7-39: Header Log Register

Table 7-36: Header Log Register

Bit Location	Register Description	Attributes	Default
127:0	Header of TLP associated with error	ROS	0

7.10.9. Root Error Command Register (Offset 2Ch)

The Root Error Command register allows further control of Root Complex response to Correctable, Non-Fatal, and Fatal error Messages than the basic Root Complex capability to generate system errors in response to error Messages (either received or internally generated). Bit fields (see Figure 7-40) enable or disable generation of interrupts (claimed by the Root Port or Root Complex Event Collector) in addition to system error Messages according to the definitions in Table 7-37.

For both Root Ports and Root Complex Event Collectors, in order for a received error Message or an internally generated error Message to generate an interrupt enabled by this register, the error Message must be enabled for “transmission” by the Root Port or Root Complex Event Collector (see Section 6.2.4.1 and Section 6.2.8.1).

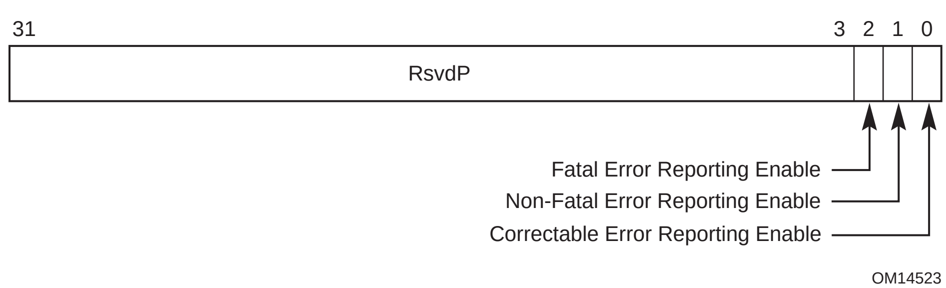

Figure 7-40: Root Error Command Register

Table 7-37: Root Error Command Register

Bit Location	Register Description	Attributes	Default
0	**Correctable Error Reporting Enable** – When Set, this bit enables the generation of an interrupt when a correctable error is reported by any of the Functions in the hierarchy associated with this Root Port. Root Complex Event Collectors provide support for the above described functionality for Root Complex Integrated Endpoints. Refer to Section 6.2 for further details.	RW	0b
1	**Non-Fatal Error Reporting Enable** – When Set, this bit enables the generation of an interrupt when a Non-fatal error is reported by any of the Functions in the hierarchy associated with this Root Port. Root Complex Event Collectors provide support for the above described functionality for Root Complex Integrated Endpoints. Refer to Section 6.2 for further details.	RW	0b
2	**Fatal Error Reporting Enable** – When Set, this bit enables the generation of an interrupt when a Fatal error is reported by any of the Functions in the hierarchy associated with this Root Port. Root Complex Event Collectors provide support for the above described functionality for Root Complex Integrated Endpoints. Refer to Section 6.2 for further details.	RW	0b

System error generation in response to PCI Express error Messages may be turned off by system software using the PCI Express Capability structure described in Section 7.8 when advanced error reporting via interrupts is enabled. Refer to Section 6.2 for further details.

7.10.10. Root Error Status Register (Offset 30h)

The Root Error Status register reports status of error Messages (ERR_COR, ERR_NONFATAL, and ERR_FATAL) received by the Root Port, and of errors detected by the Root Port itself (which are treated conceptually as if the Root Port had sent an error Message to itself). In order to update this register, error Messages received by the Root Port and/or internally generated error Messages must be enabled for "transmission" by the primary interface of the Root Port. ERR_NONFATAL and ERR_FATAL Messages are grouped together as uncorrectable. Each correctable and uncorrectable (Non-fatal and Fatal) error source has a first error bit and a next error bit associated with it respectively. When an error is received by a Root Complex, the respective first error bit is Set and the Requester ID is logged in the Error Source Identification register. A Set individual error status bit indicates that a particular error category occurred; software may clear an error status by writing a 1b to the respective bit. If software does not clear the first reported error before another error Message is received of the same category (correctable or uncorrectable), the corresponding next error status bit will be set but the Requester ID of the subsequent error Message is discarded. The next error status bits may be cleared by software by writing a 1b to the respective bit as well. Refer to Section 6.2 for further details. This register is updated regardless of the settings of the Root Control register and the Root Error Command register. Figure 7-41 details allocation of register fields in the Root Error Status register; Table 7-38 provides the respective bit definitions. Root Complex Event Collectors provide support for the above-described functionality for Root Complex Integrated Endpoints (and for the Root Complex Event Collector itself). In order to update this register, error Messages received by the Root Complex Event Collector from its associated Root Complex Integrated Endpoints and/or internally generated error Messages must be enabled for "transmission" by the Root Complex Event Collector.

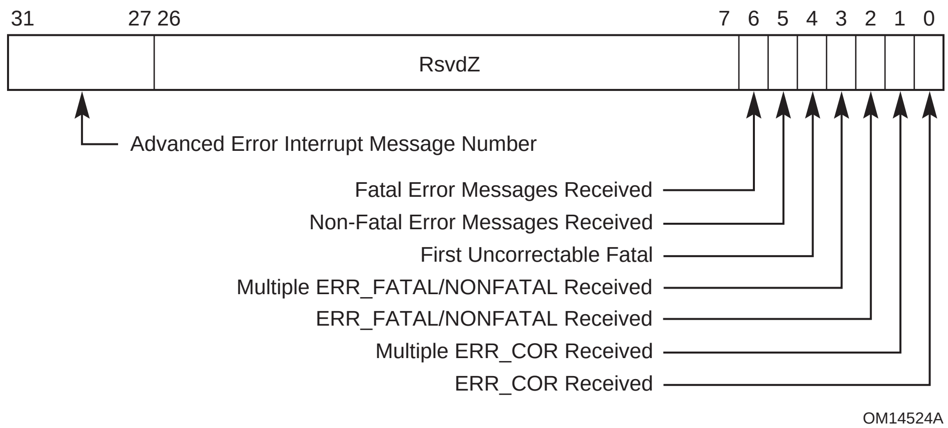

Figure 7-41: Root Error Status Register

Table 7-38: Root Error Status Register

Bit Location	Register Description	Attributes
0	**ERR_COR Received** – Set when a Correctable error Message is received and this bit is not already Set. Default value of this bit is 0b.	RW1CS
1	**Multiple ERR_COR Received** – Set when a Correctable error Message is received and ERR_COR Received is already Set. Default value of this bit is 0b.	RW1CS
2	**ERR_FATAL/NONFATAL Received** – Set when either a Fatal or a Non-fatal error Message is received and this bit is not already Set. Default value of this bit is 0b.	RW1CS
3	**Multiple ERR_FATAL/NONFATAL Received** – Set when either a Fatal or a Non-fatal error is received and **ERR_FATAL/NONFATAL Received** is already Set. Default value of this bit is 0b.	RW1CS

Bit Location	Register Description	Attributes
4	**First Uncorrectable Fatal** – Set when the first Uncorrectable error Message received is for a Fatal error. Default value of this field is 0b.	RW1CS
5	**Non-Fatal Error Messages Received** – Set when one or more Non-Fatal Uncorrectable error Messages have been received. Default value of this bit is 0b.	RW1CS
6	**Fatal Error Messages Received** – Set when one or more Fatal Uncorrectable error Messages have been received. Default value of this bit is 0b.	RW1CS
31:27	**Advanced Error Interrupt Message Number** – This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability. For MSI, the value in this register indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register. For MSI-X, the value in this register indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If MSI-X is enabled, the value in this register must indicate the vector for MSI-X. If MSI is enabled or neither is enabled, the value in this register must indicate the vector for MSI. If software enables both MSI and MSI-X at the same time, the value in this register is undefined.	RO

7.10.11. Error Source Identification Register (Offset 34h)

The Error Source Identification register identifies the source (Requester ID) of first correctable and uncorrectable (Non-fatal/Fatal) errors reported in the Root Error Status register. Refer to Section 6.2 for further details. This register is updated regardless of the settings of the Root Control register and the Root Error Command register. Figure 7-42 details allocation of register fields in the Error Source Identification register; Table 7-39 provides the respective bit definitions.

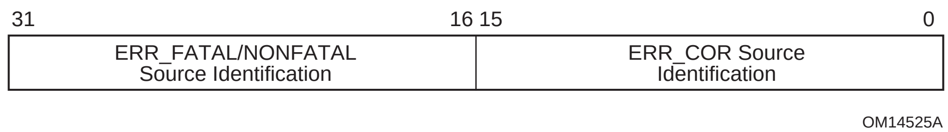

Figure 7-42: Error Source Identification Register

Table 7-39: Error Source Identification Register

Bit Location	Register Description	Attributes
15:0	**ERR_COR Source Identification** – Loaded with the Requester ID indicated in the received ERR_COR Message when the ERR_COR Received bit is not already set. Default value of this field is 0000h.	ROS
31:16	**ERR_FATAL/NONFATAL Source Identification** – Loaded with the Requester ID indicated in the received ERR_FATAL or ERR_NONFATAL Message when the ERR_FATAL/NONFATAL Received bit is not already set. Default value of this field is 0000h.	ROS

7.10.12. TLP Prefix Log Register (Offset 38h)

The TLP Prefix Log register captures the End-End TLP Prefix(s) for the TLP corresponding to the detected error; refer to Section 6.2 for further details. The TLP Prefix Log register is only meaningful when the TLP Prefix Log Present bit is Set (see Section 7.10.7).

The TLP Prefixes are captured such that, when read using DW accesses, the fields of the TLP Prefix are laid out in the same way the fields of the TLP Prefix are described. Therefore, byte 0 of a TLP Prefix is located in byte 3 of the associated TLP Prefix Log register; byte 1 of a TLP Prefix is located in byte 2; and so forth.

The First TLP Prefix Log Register contains the first End-End TLP Prefix from the TLP (see Section 6.2.4.4). The Second TLP Prefix Log register contains the second End-End TLP Prefix and so forth. If the TLP contains fewer than four End-End TLP Prefixes, the remaining TLP Prefix Log Registers contain zero. A TLP that contains more End-End TLP Prefixes than are indicated by the Function's Max End-End TLP Prefixs field must be handled as a Malformed TLP (see Section 2.2.10.2). To allow software to detect this condition, the supported number of End-End

TLP Prefixes are logged in this register, the first overflow End-End TLP Prefix is logged in the first DW of the Header Log register and the remaining DWs of the Header Log Register are undefined (see Section 6.2.4.4).

The TLP Prefix Log Registers beyond the number supported by the Function are hardwired to zero. For example, if a Functions, Max End-End TLP Prefixes field contains 10b (indicating 2 DW of buffering) then the third and fourth TLP Prefix Log Registers are hardwired to zero.

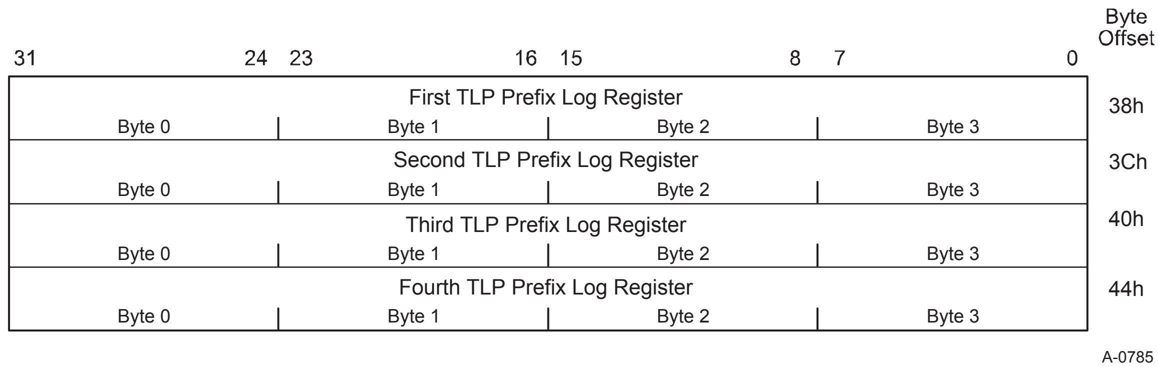

Figure 7-43: TLP Prefix Log Register

Table 7-40: TLP Prefix Log Register

Bit Location	Register Description	Attributes	Default
127:0	**TLP Prefix Log**	ROS	0

7.11. Virtual Channel Capability

The Virtual Channel (VC) Capability is an optional Extended Capability required for devices that have Ports (or for individual Functions) that support functionality beyond the default Traffic Class (TC0) over the default Virtual Channel (VC0). This may apply to devices with only one VC that support TC filtering or to devices that support multiple VCs. Note that a PCI Express device that supports only TC0 over VC0 does not require VC Extended Capability and associated registers. Figure 7-44 provides a high level view of the PCI Express Virtual Channel Capability structure. This structure controls Virtual Channel assignment for PCI Express Links and may be present in any device (or RCRB) that contains (controls) a Port, or any device that has a Multi-Function Virtual Change (MFVC) Capability structure. Some registers/fields in the PCI Express Virtual Channel Capability structure may have different interpretation for Endpoints, Switch Ports, Root Ports and RCRB. Software must interpret the Device/Port Type field in the PCI Express Capabilities register to determine the availability and meaning of these registers/fields.

The PCI Express VC Capability structure is permitted in the Extended Configuration Space of all single-Function devices or in RCRBs.

A multi-Function device at an Upstream Port is permitted to optionally contain a Multi-Function Virtual Channel (MFVC) Capability structure (see Section 7.18). If a multi-Function device contains an MFVC Capability structure, any or all of its Functions are permitted to contain a VC Capability structure. Per-Function VC Capability structures are also permitted for devices inside a Switch that contain only Switch Downstream Port Functions, or for Root Complex Integrated Endpoints. Otherwise, only Function 0 is permitted to contain a VC Capability structure.

To preserve software backward compatibility, two Extended Capability IDs are permitted for VC Capability structures: 0002h and 0009h. Any VC Capability structure in a device that also contains an MFVC Capability structure must use the Extended Capability ID 0009h. A VC Capability structure in a device that does not contain an MFVC Capability structure must use the Extended Capability ID 0002h.

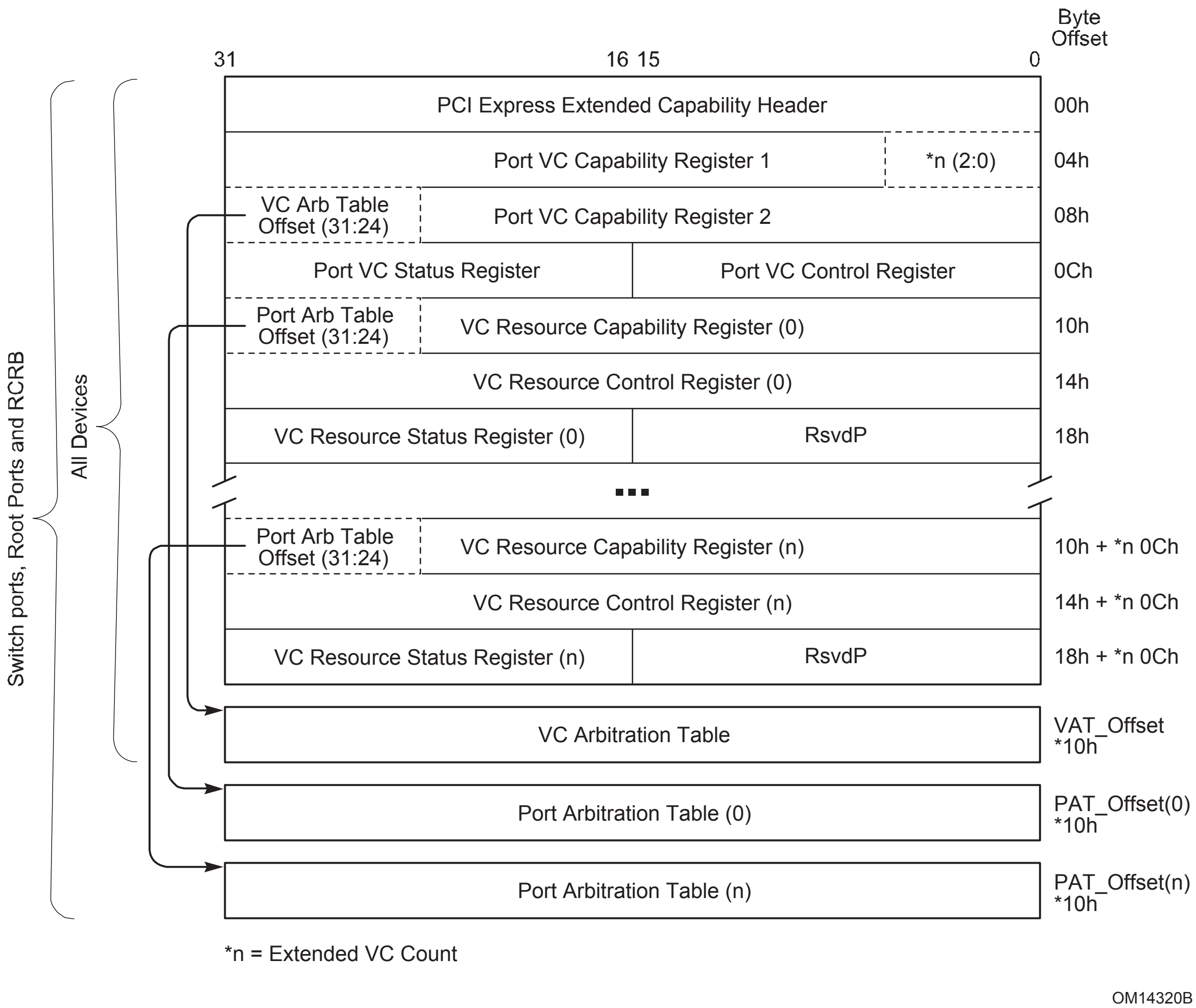

Figure 7-44: PCI Express Virtual Channel Capability Structure

The following sections describe the registers/fields of the PCI Express Virtual Channel Capability structure.

7.11.1. Virtual Channel Extended Capability Header

Refer to Section 7.9.3 for a description of the PCI Express Extended Capability header. A Virtual Channel Capability must use one of two Extended Capability IDs: 0002h or 0009h. Refer to Section 7.11 for rules governing when each should be used. Figure 7-45 details allocation of register fields in the Virtual Channel Extended Capability header; Table 7-41 provides the respective bit definitions.

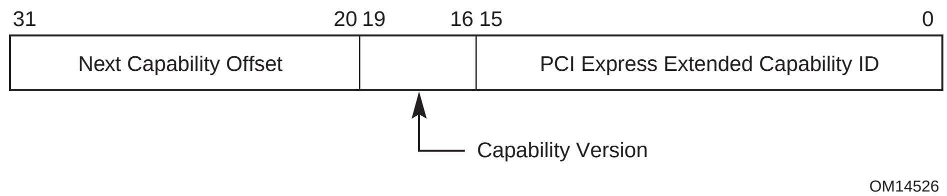

Figure 7-45: Virtual Channel Extended Capability Header

Table 7-41: Virtual Channel Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Virtual Channel Capability is either 0002h or 0009h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.11.2. Port VC Capability Register 1

The Port VC Capability Register 1 describes the configuration of the Virtual Channels associated with a PCI Express Port. Figure 7-46 details allocation of register fields in the Port VC Capability Register 1; Table 7-42 provides the respective bit definitions.

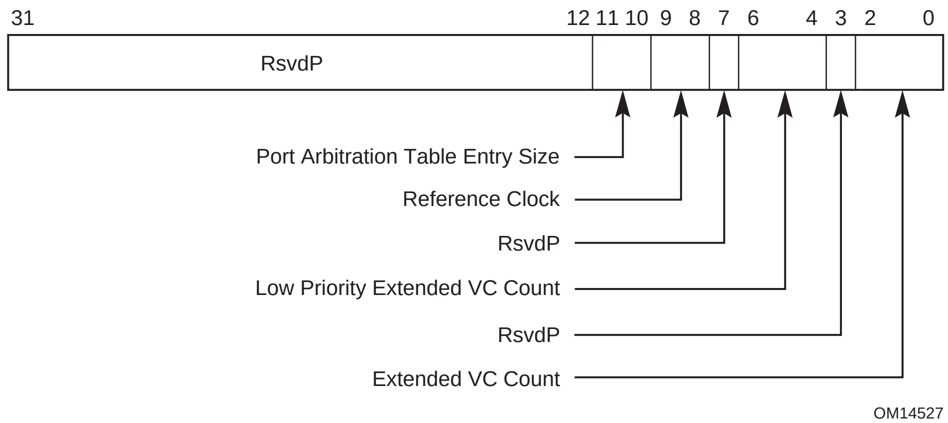

Figure 7-46: Port VC Capability Register 1

Table 7-42: Port VC Capability Register 1

Bit Location	Register Description	Attributes
2:0	**Extended VC Count** – Indicates the number of (extended) Virtual Channels in addition to the default VC supported by the device. This field is valid for all Functions. The minimum value of this field is 000b (for devices that only support the default VC). The maximum value is 7.	RO
6:4	**Low Priority Extended VC Count** – Indicates the number of (extended) Virtual Channels in addition to the default VC belonging to the low-priority VC (LPVC) group that has the lowest priority with respect to other VC resources in a strict-priority VC Arbitration. This field is valid for all Functions. The minimum value of this field is 000b and the maximum value is Extended VC Count.	RO
9:8	**Reference Clock** – Indicates the reference clock for Virtual Channels that support time-based WRR Port Arbitration. This field is valid for RCRBs, Switch Ports, and Root Ports that support peer-to-peer traffic. It is not valid for Root Ports that do not support peer-to-peer traffic, Endpoints, and Switches or Root Complexes not implementing WRR, and must be hardwired to 00b. Defined encodings are: 00b 100 ns reference clock 01b – 11b Reserved	RO

Bit Location	Register Description	Attributes
11:10	**Port Arbitration Table Entry Size** – Indicates the size (in bits) of Port Arbitration table entry in the Function. This field is valid only for RCRBs, Switch Ports, and Root Ports that support peer-to-peer traffic. It is not valid and must be hardwired to 00b for Root Ports that do not support peer-to-peer traffic and Endpoints. Defined encodings are: 00b The size of Port Arbitration table entry is 1 bit. 01b The size of Port Arbitration table entry is 2 bits. 10b The size of Port Arbitration table entry is 4 bits. 11b The size of Port Arbitration table entry is 8 bits.	RO

7.11.3. Port VC Capability Register 2

The Port VC Capability Register 2 provides further information about the configuration of the Virtual Channels associated with a PCI Express Port. Figure 7-47 details allocation of register fields in the Port VC Capability Register 2; Table 7-43 provides the respective bit definitions.

Figure 7-47: Port VC Capability Register 2

Table 7-43: Port VC Capability Register 2

Bit Location	Register Description	Attributes
7:0	**VC Arbitration Capability** – Indicates the types of VC Arbitration supported by the Function for the LPVC group. This field is valid for all Functions that report a Low Priority Extended VC Count field greater than 0. For all other Functions, this field must be hardwired to 00h. Each bit location within this field corresponds to a VC Arbitration Capability defined below. When more than 1 bit in this field is Set, it indicates that the Port can be configured to provide different VC arbitration services. Defined bit positions are: Bit 0 — Hardware fixed arbitration scheme, e.g., Round Robin Bit 1 — Weighted Round Robin (WRR) arbitration with 32 phases Bit 2 — WRR arbitration with 64 phases Bit 3 — WRR arbitration with 128 phases Bits 4-7 — Reserved	RO
31:24	**VC Arbitration Table Offset** – Indicates the location of the VC Arbitration Table. This field is valid for all Functions. This field contains the zero-based offset of the table in DQWORDS (16 bytes) from the base address of the Virtual Channel Capability structure. A value of 0 indicates that the table is not present.	RO

7.11.4. Port VC Control Register

Figure 7-48 details allocation of register fields in the Port VC Control register; Table 7-44 provides the respective bit definitions.

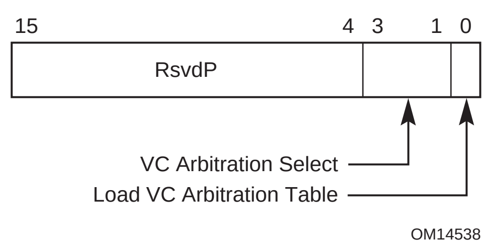

Figure 7-48: Port VC Control Register

Table 7-44: Port VC Control Register

Bit Location	Register Description	Attributes
0	**Load VC Arbitration Table** – Used by software to update the VC Arbitration Table. This bit is valid for all Functions when the selected VC Arbitration uses the VC Arbitration Table. Software sets this bit to request hardware to apply new values programmed into VC Arbitration Table; clearing this bit has no effect. Software checks the VC Arbitration Table Status bit to confirm that new values stored in the VC Arbitration Table are latched by the VC arbitration logic. This bit always returns 0b when read.	RW
3:1	**VC Arbitration Select** – Used by software to configure the VC arbitration by selecting one of the supported VC Arbitration schemes indicated by the VC Arbitration Capability field in the Port VC Capability Register 2. This field is valid for all Functions. The permissible values of this field are numbers corresponding to one of the asserted bits in the VC Arbitration Capability field. This field cannot be modified when more than one VC in the LPVC group is enabled.	RW

7.11.5. Port VC Status Register

The Port VC Status register provides status of the configuration of Virtual Channels associated with a Port. Figure 7-49 details allocation of register fields in the Port VC Status register; Table 7-45 provides the respective bit definitions.

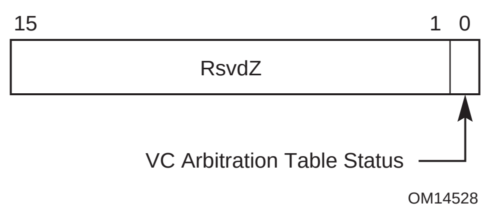

Figure 7-49: Port VC Status Register

Table 7-45: Port VC Status Register

Bit Location	Register Description	Attributes
0	**VC Arbitration Table Status** – Indicates the coherency status of the VC Arbitration Table. This field is valid for all Functions when the selected VC uses the VC Arbitration Table. This bit is Set by hardware when any entry of the VC Arbitration Table is written by software. This bit is cleared by hardware when hardware finishes loading values stored in the VC Arbitration Table after software sets the Load VC Arbitration Table field in the Port VC Control register. Default value of this field is 0b.	RO

7.11.6. VC Resource Capability Register

The VC Resource Capability register describes the capabilities and configuration of a particular Virtual Channel resource. Figure 7-50 details allocation of register fields in the VC Resource Capability register; Table 7-46 provides the respective bit definitions.

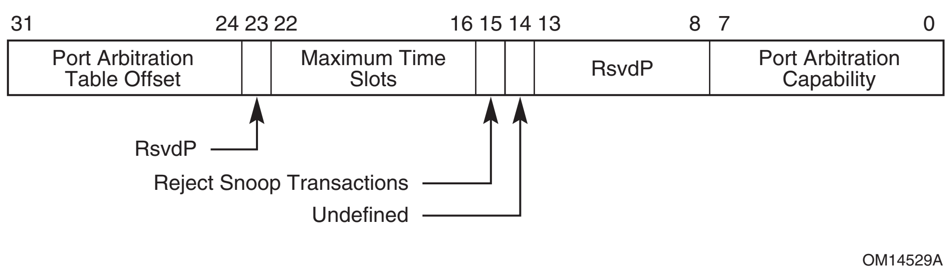

Figure 7-50: VC Resource Capability Register

Table 7-46: VC Resource Capability Register

Bit Location	Register Description	Attributes
7:0	**Port Arbitration Capability** – Indicates types of Port Arbitration supported by the VC resource. This field is valid for all Switch Ports, Root Ports that support peer-to-peer traffic, and RCRBs, but not for Endpoints or Root Ports that do not support peer-to-peer traffic. Each bit location within this field corresponds to a Port Arbitration Capability defined below. When more than 1 bit in this field is Set, it indicates that the VC resource can be configured to provide different arbitration services. Software selects among these capabilities by writing to the Port Arbitration Select field (see below). Defined bit positions are: Bit 0 Non-configurable hardware-fixed arbitration scheme, e.g., Round Robin (RR) Bit 1 Weighted Round Robin (WRR) arbitration with 32 phases Bit 2 WRR arbitration with 64 phases Bit 3 WRR arbitration with 128 phases Bit 4 Time-based WRR with 128 phases Bit 5 WRR arbitration with 256 phases Bits 6-7 Reserved	RO
14	**Undefined** – The value read from this bit is undefined. In previous versions of this specification, this bit was used to indicate Advanced Packet Switching. System software must ignore the value read from this bit.	RO
15	**Reject Snoop Transactions** – When Clear, transactions with or without the No Snoop bit Set within the TLP header are allowed on this VC. When Set, any transaction for which the No Snoop attribute is applicable but is not Set within the TLP header is permitted to be rejected as an Unsupported Request. Refer to Section 2.2.6.5 for information on where the No Snoop attribute is applicable. This field is valid for Root Ports and RCRB; it is not valid for Endpoints or Switch Ports.	HwInit
22:16	**Maximum Time Slots** – Indicates the maximum number of time slots (minus one) that the VC resource is capable of supporting when it is configured for time-based WRR Port Arbitration. For example, a value 000 0000b in this field indicates the supported maximum number of time slots is 1 and a value of 111 1111b indicates the supported maximum number of time slot is 128. This field is valid for all Switch Ports, Root Ports that support peer-to-peer traffic, and RCRBs, but is not valid for Endpoints or Root Ports that do not support peer-to-peer traffic. In addition, this field is valid only when the Port Arbitration Capability field indicates that the VC resource supports time-based WRR Port Arbitration.	HwInit

Bit Location	Register Description	Attributes
31:24	**Port Arbitration Table Offset** – Indicates the location of the Port Arbitration Table associated with the VC resource. This field is valid for all Switch Ports, Root Ports that support peer-to-peer traffic, and RCRBs, but is not valid for Endpoints or Root Ports that do not support peer-to-peer traffic. This field contains the zero-based offset of the table in DQWORDS (16 bytes) from the base address of the Virtual Channel Capability structure. A value of 00h indicates that the table is not present.	RO

7.11.7. VC Resource Control Register

Figure 7-51 details allocation of register fields in the VC Resource Control register; Table 7-47 provides the respective bit definitions.

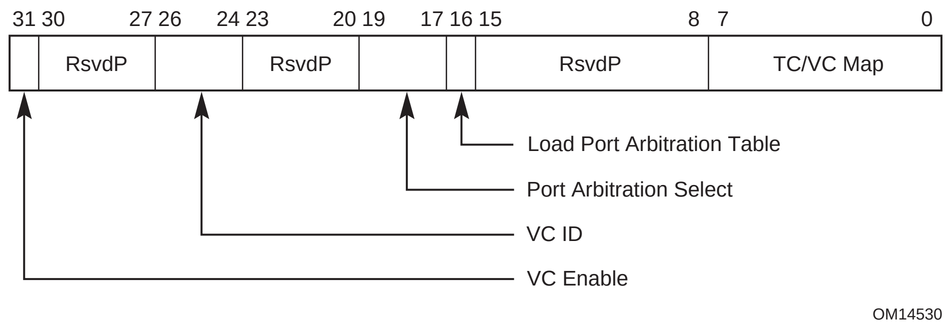

Figure 7-51: VC Resource Control Register

Table 7-47: VC Resource Control Register

Bit Location	Register Description	Attributes
7:0	**TC/VC Map** – This field indicates the TCs that are mapped to the VC resource. This field is valid for all Functions. Bit locations within this field correspond to TC values. For example, when bit 7 is Set in this field, TC7 is mapped to this VC resource. When more than 1 bit in this field is Set, it indicates that multiple TCs are mapped to the VC resource. In order to remove one or more TCs from the TC/VC Map of an enabled VC, software must ensure that no new or outstanding transactions with the TC labels are targeted at the given Link. Default value of this field is FFh for the first VC resource and is 00h for other VC resources. Note: Bit 0 of this field is read-only. It must be Set for the default VC0 and Clear for all other enabled VCs.	RW (see the note for exceptions)

Bit Location	Register Description	Attributes
16	**Load Port Arbitration Table** – When Set, this bit updates the Port Arbitration logic from the Port Arbitration Table for the VC resource. This bit is valid for all Switch Ports, Root Ports that support peer-to-peer traffic, and RCRBs, but is not valid for Endpoints or Root Ports that do not support peer-to-peer traffic. In addition, this bit is only valid when the Port Arbitration Table is used by the selected Port Arbitration scheme (that is indicated by a Set bit in the Port Arbitration Capability field selected by Port Arbitration Select). Software sets this bit to signal hardware to update Port Arbitration logic with new values stored in Port Arbitration Table; clearing this bit has no effect. Software uses the Port Arbitration Table Status bit to confirm whether the new values of Port Arbitration Table are completely latched by the arbitration logic. This bit always returns 0b when read. Default value of this bit is 0b.	RW
19:17	**Port Arbitration Select** – This field configures the VC resource to provide a particular Port Arbitration service. This field is valid for RCRBs, Root Ports that support peer-to-peer traffic, and Switch Ports, but is not valid for Endpoints or Root Ports that do not support peer-to-peer traffic. The permissible value of this field is a number corresponding to one of the asserted bits in the Port Arbitration Capability field of the VC resource.	RW
26:24	**VC ID** – This field assigns a VC ID to the VC resource (see note for exceptions). This field is valid for all Functions. This field cannot be modified when the VC is already enabled. Note: For the first VC resource (default VC), this field is read-only and must be hardwired to 000b.	RW

Bit Location	Register Description	Attributes
31	**VC Enable** – This bit, when Set, enables a Virtual Channel (see note 1 for exceptions). The Virtual Channel is disabled when this bit is cleared. This bit is valid for all Functions. Software must use the VC Negotiation Pending bit to check whether the VC negotiation is complete. Default value of this bit is 1b for the first VC resource and is 0b for other VC resource(s). Notes: 1. This bit is hardwired to 1b for the default VC (VC0), i.e., writing to this bit has no effect for VC0. 2. To enable a Virtual Channel, the VC Enable bits for that Virtual Channel must be Set in both components on a Link. 3. To disable a Virtual Channel, the VC Enable bits for that Virtual Channel must be cleared in both components on a Link. 4. Software must ensure that no traffic is using a Virtual Channel at the time it is disabled. 5. Software must fully disable a Virtual Channel in both components on a Link before re-enabling the Virtual Channel.	RW

7.11.8. VC Resource Status Register

Figure 7-52 details allocation of register fields in the VC Resource Status register; Table 7-48 provides the respective bit definitions.

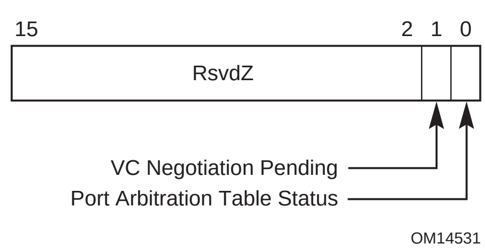

Figure 7-52: VC Resource Status Register

Table 7-48: VC Resource Status Register

Bit Location	Register Description	Attributes
0	**Port Arbitration Table Status** – This bit indicates the coherency status of the Port Arbitration Table associated with the VC resource. This bit is valid for RCRBs, Root Ports that support peer-to-peer traffic, and Switch Ports, but is not valid for Endpoints or Root Ports that do not support peer-to-peer traffic. In addition, this bit is valid only when the Port Arbitration Table is used by the selected Port Arbitration for the VC resource. This bit is Set by hardware when any entry of the Port Arbitration Table is written to by software. This bit is cleared by hardware when hardware finishes loading values stored in the Port Arbitration Table after software sets the Load Port Arbitration Table bit. Default value of this bit is 0b.	RO
1	**VC Negotiation Pending** –This bit indicates whether the Virtual Channel negotiation (initialization or disabling) is in pending state. This bit is valid for all Functions. The value of this bit is defined only when the Link is in the DL_Active state and the Virtual Channel is enabled (its VC Enable bit is Set). When this bit is Set by hardware, it indicates that the VC resource has not completed the process of negotiation. This bit is cleared by hardware after the VC negotiation is complete (on exit from the FC_INIT2 state). For VC0, this bit is permitted to be hardwired to 0b. Before using a Virtual Channel, software must check whether the VC Negotiation Pending bits for that Virtual Channel are Clear in both components on the Link.	RO

7.11.9. VC Arbitration Table

The VC Arbitration Table is a read-write register array that is used to store the arbitration table for VC Arbitration. This register array is valid for all Functions when the selected VC Arbitration uses a WRR table. If it exists, the VC Arbitration Table is located by the VC Arbitration Table Offset field.

The VC Arbitration Table is a register array with fixed-size entries of 4 bits. Figure 7-53 depicts the table structure of an example VC Arbitration Table with 32 phases. Each 4-bit table entry corresponds to a phase within a WRR arbitration period. The definition of table entry is depicted in Table 7-49. The lower 3 bits (bits 0-2) contain the VC ID value, indicating that the corresponding phase within the WRR arbitration period is assigned to the Virtual Channel indicated by the VC ID (must be a valid VC ID that corresponds to an enabled VC).

The highest bit (bit 3) of the table entry is reserved. The length of the table depends on the selected VC Arbitration as shown in Table 7-50.

When the VC Arbitration Table is used by the default VC Arbitration method, the default values of the table entries must be all zero to ensure forward progress for the default VC (with VC ID of 0).

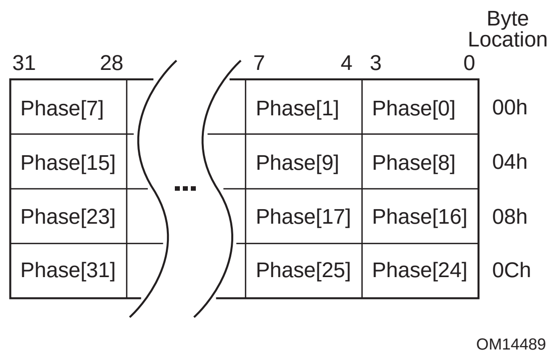

Figure 7-53: Example VC Arbitration Table with 32 Phases

Table 7-49: Definition of the 4-bit Entries in the VC Arbitration Table

Bit Location	Description	Attributes
2:0	VC ID	RW
3	RsvdP	RW

Table 7-50: Length of the VC Arbitration Table

VC Arbitration Select	VC Arbitration Table Length (in # of Entries)
001b	32
010b	64
011b	128

7.11.10. Port Arbitration Table

The Port Arbitration Table register is a read-write register array that is used to store the WRR or time-based WRR arbitration table for Port Arbitration for the VC resource. This register array is valid for all Switch Ports, Root Ports that support peer-to-peer traffic, and RCRBs, but is not valid for Endpoints or Root Ports that do not support peer-to-peer traffic. It is only present when one or more asserted bits in the Port Arbitration Capability field indicate that the component supports a Port Arbitration scheme that uses a programmable arbitration table. Furthermore, it is only valid when one of the above-mentioned bits in the Port Arbitration Capability field is selected by the Port Arbitration Select field.

The Port Arbitration Table represents one Port arbitration period. Figure 7-54 shows the structure of an example Port Arbitration Table with 128 phases and 2-bit table entries. Each table entry containing a Port Number corresponds to a phase within a Port arbitration period. For example, a table with 2-bit entries can be used by a Switch component with up to four Ports. A Port Number written to a table entry indicates that the phase within the Port Arbitration period is assigned to the selected PCI Express Port (the Port Number must be a valid one).

- When the WRR Port Arbitration is used for a VC of any Egress Port, at each arbitration phase, the Port Arbiter serves one transaction from the Ingress Port indicated by the Port Number of the current phase. When finished, it immediately advances to the next phase. A phase is skipped, i.e., the Port Arbiter simply moves to the next phase immediately if the Ingress Port indicated by the phase does not contain any transaction for the VC (note that a phase cannot contain the Egress Port's Port Number).
- When the Time-based WRR Port Arbitration is used for a VC of any given Port, at each arbitration phase aligning to a virtual timeslot, the Port Arbiter serves one transaction from the Ingress Port indicated by the Port Number of the current phase. It advances to the next phase at the next virtual timeslot. A phase indicates an "idle" timeslot, i.e., the Port Arbiter does not serve any transaction during the phase, if
    - the phase contains the Egress Port's Port Number, or
    - the Ingress Port indicated by the phase does not contain any transaction for the VC.

The Port Arbitration Table Entry Size field in the Port VC Capability Register 1 determines the table entry size. The length of the table is determined by the Port Arbitration Select field as shown in Table 7-51.

When the Port Arbitration Table is used by the default Port Arbitration for the default VC, the default values for the table entries must contain at least one entry for each of the other PCI Express Ports of the component to ensure forward progress for the default VC for each Port. The table may contain RR or RR-like fair Port Arbitration for the default VC.

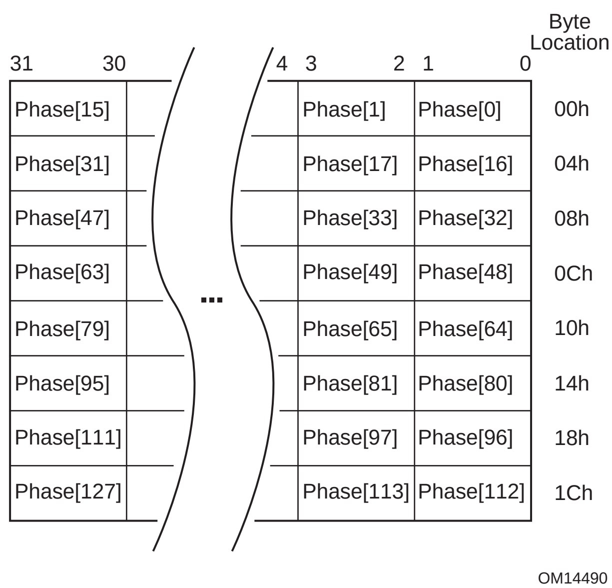

Figure 7-54: Example Port Arbitration Table with 128 Phases and 2-bit Table Entries

Table 7-51: Length of Port Arbitration Table

Port Arbitration Select	Port Arbitration Table Length (in Number of Entries)
001b	32
010b	64
011b	128
100b	128
101b	256

7.12. Device Serial Number Capability

The PCI Express Device Serial Number Capability is an optional Extended Capability that may be implemented by any PCI Express device Function. The Device Serial Number is a read-only 64-bit value that is unique for a given PCI Express device. Figure 7-55 details allocation of register fields in the PCI Express Capability structure.

All multi-Function devices that implement this Capability must implement it for Function 0; other Functions that implement this Capability must return the same Device Serial Number value as that reported by Function 0.

A PCI Express multi-device component such as a PCI Express Switch that implements this Capability must return the same Device Serial Number for each device.

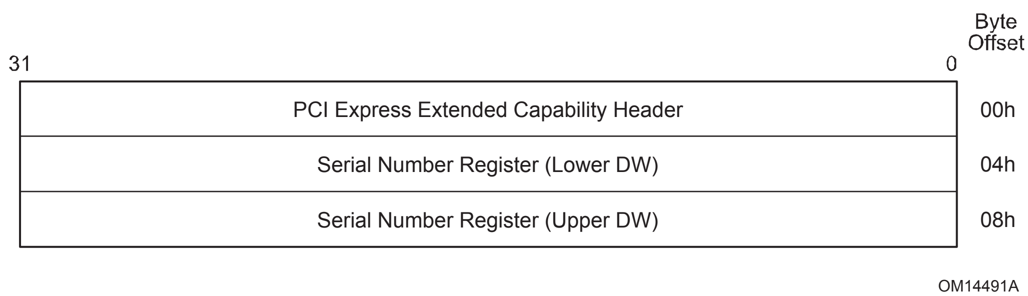

Figure 7-55: PCI Express Device Serial Number Capability Structure

7.12.1. Device Serial Number Extended Capability Header (Offset 00h)

Figure 7-56 details allocation of register fields in the Device Serial Number Extended Capability header; Table 7-52 provides the respective bit definitions. Refer to Section 7.9.3 for a description of the PCI Express Extended Capability header. The Extended Capability ID for the Device Serial Number Capability is 0003h.

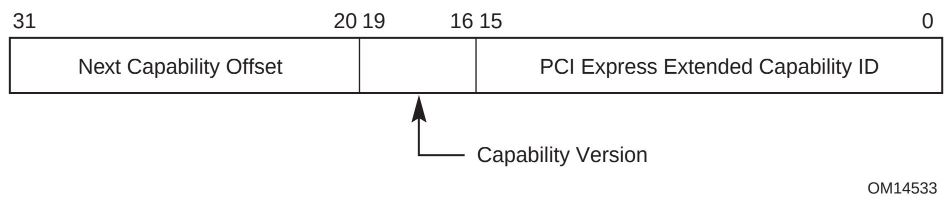

Figure 7-56: Device Serial Number Extended Capability Header

Table 7-52: Device Serial Number Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Device Serial Number Capability is 0003h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.12.2. Serial Number Register (Offset 04h)

The Serial Number register is a 64-bit field that contains the IEEE defined 64-bit extended unique identifier (EUI-64™). Figure 7-57 details allocation of register fields in the Serial Number register; Table 7-53 provides the respective bit definitions.

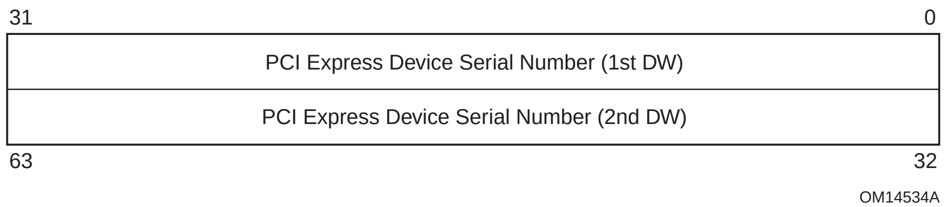

Figure 7-57: Serial Number Register

Table 7-53: Serial Number Register

Bit Location	Register Description	Attributes
63:0	**PCI Express Device Serial Number** – This field contains the IEEE defined 64-bit extended unique identifier (EUI-64™). This identifier includes a 24-bit company id value assigned by IEEE registration authority and a 40-bit extension identifier assigned by the manufacturer.	RO

7.13. PCI Express Root Complex Link Declaration Capability

The PCI Express Root Complex Link Declaration Capability is an optional Capability that is permitted to be implemented by Root Ports, Root Complex Integrated Endpoints, or RCRBs to declare a Root Complex's internal topology.

A Root Complex consists of one or more following elements:

- PCI Express Root Port
- A default system Egress Port or an internal sink unit such as memory (represented by an RCRB)
- Internal Data Paths/Links (represented by an RCRB on either side of an internal Link)
- Integrated devices
- Functions

A Root Complex Component is a logical aggregation of the above described Root Complex elements. No single element can be part of more than one Root Complex Component. Each Root Complex Component must have a unique Component ID.

A Root Complex is represented either as an opaque Root Complex or as a collection of one or more Root Complex Components.

The PCI Express Root Complex Link Declaration Capability is permitted to be present in a Root Complex element's Configuration Space or RCRB. It declares Links from the respective element to other elements of the same Root Complex Component or to an element in another Root Complex Component. The Links are required to be declared bidirectional such that each valid data path from one element to another has corresponding Link entries in the Configuration Space (or RCRB) of both elements.

The PCI Express Root Complex Link Declaration Capability is permitted to also declare an association between a Configuration Space element (Root Port or Root Complex Integrated Endpoint) and an RCRB Header Capability (see Section 7.20) contained in an RCRB that affects the behavior of the Configuration Space element. Note that an RCRB Header association is not declared bidirectional; the association is only declared by the Configuration Space element and not by the target RCRB.

IMPLEMENTATION NOTE

Topologies to Avoid

Topologies that create more than one data path between any two Root Complex elements (either directly or through other Root Complex elements) may not be able to support bandwidth allocation in a standard manner. The description of how traffic is routed through such a topology is implementation specific, meaning that general purpose-operating systems may not have enough information about such a topology to correctly support bandwidth allocation. In order to circumvent this problem, these operating systems may require that a single RCRB element (of type Internal Link) not declare more than one Link to a Root Complex Component other than the one containing the RCRB element itself.

The PCI Express Root Complex Link Declaration Capability, as shown in Figure 7-58, consists of the PCI Express Extended Capability header and Root Complex Element Self Description followed by one or more Root Complex Link Entries.

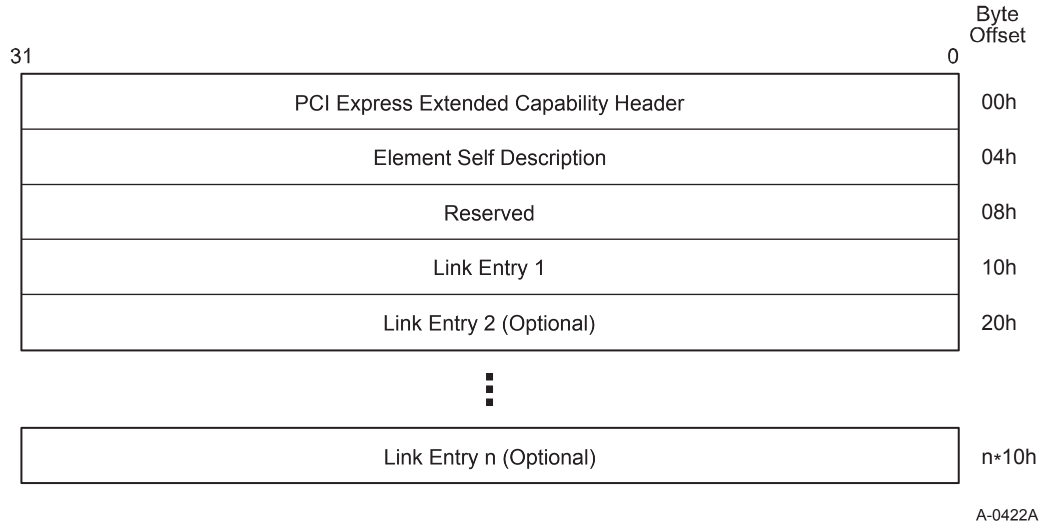

Figure 7-58: PCI Express Root Complex Link Declaration Capability

7.13.1. Root Complex Link Declaration Extended Capability Header

The Extended Capability ID for the Root Complex Link Declaration Capability is 0005h.

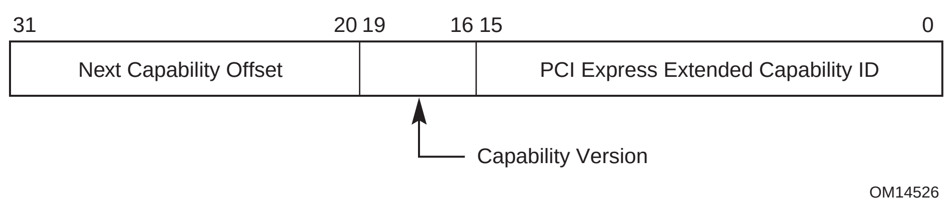

Figure 7-59: Root Complex Link Declaration Extended Capability Header

Table 7-54: Root Complex Link Declaration Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Link Declaration Capability is 0005h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO

Bit Location	Register Description	Attributes
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits.	RO

7.13.2. Element Self Description

The Element Self Description register provides information about the Root Complex element containing the Link Declaration Capability.

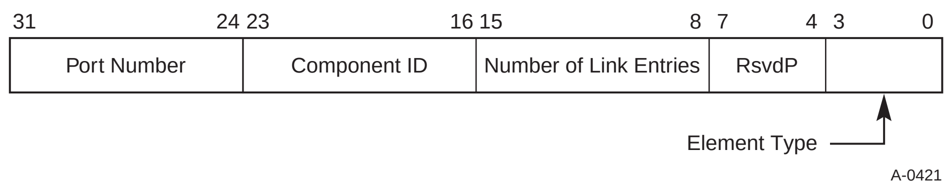

Figure 7-60: Element Self Description Register

Table 7-55: Element Self Description Register

Bit Location	Register Description	Attributes
3:0	**Element Type** – This field indicates the type of the Root Complex Element. Defined encodings are: 0h Configuration Space Element 1h System Egress Port or internal sink (memory) 2h Internal Root Complex Link 3h-15h Reserved	RO
15:8	**Number of Link Entries** – This field indicates the number of Link entries following the Element Self Description. This field must report a value of 01h or higher.	HwInit
23:16	**Component ID** – This field identifies the Root Complex Component that contains this Root Complex Element. Components IDs must start at 01h, as a value of 00h is reserved.	HwInit

Bit Location	Register Description	Attributes
31:24	**Port Number** – This field specifies the Port Number associated with this element with respect to the Root Complex Component that contains this element. An element with a Port Number of 00h indicates the default Egress Port to configuration software.	HwInit

7.13.3. Link Entries

Link Entries start at offset 10h of the PCI Express Root Complex Link Declaration Capability structure. Each Link Entry consists of a Link description followed by a 64-bit Link address at offset 08h from the start of Link entry identifying the target element for the declared Link. A Link Entry declares an internal Link to another Root Complex Element.

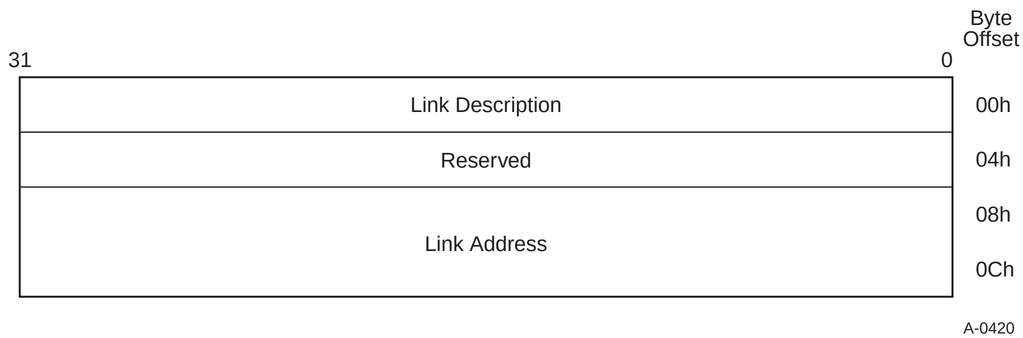

Figure 7-61: Link Entry

7.13.3.1. Link Description

The Link Description is located at offset 00h from the start of a Link Entry and is defined as follows:

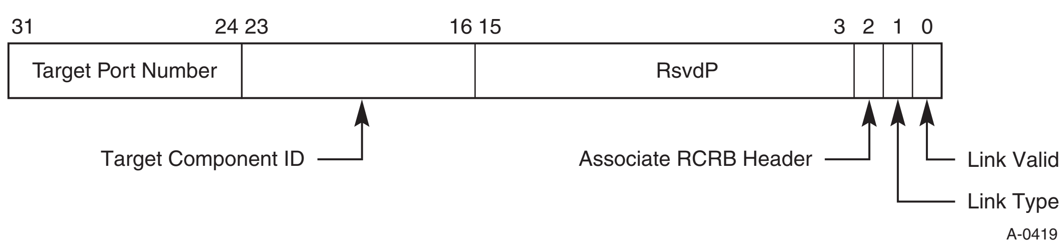

Figure 7-62: Link Description Register

Table 7-56: Link Description Register

Bit Location	Register Description	Attributes
0	**Link Valid** – When Set, this bit indicates that the Link Entry specifies a valid Link. Link entries that do not have either this bit Set or the Associate RCRB Header bit Set (or both) are ignored by software.	HwInit
1	**Link Type** – This bit indicates the target type of the Link and defines the format of the Link address field. Defined Link Type values are: 0b – Link points to memory-mapped space[103] (for RCRB). The Link address specifies the 64-bit base address of the target RCRB. 1b – Link points to Configuration Space (for a Root Port or Root Complex Integrated Endpoint). The Link address specifies the configuration address (PCI Segment Group, Bus, Device, Function) of the target element.	HwInit
2	**Associate RCRB Header** – When Set, this bit indicates that the Link Entry associates the declaring element with an RCRB Header Capability in the target RCRB. Link entries that do not have either this bit Set or the Link Valid bit Set (or both) are ignored by software. The Link Type bit must be Clear when this bit is Set.	HwInit
23:16	**Target Component ID** – This field identifies the Root Complex Component that is targeted by this Link entry. Components IDs must start at 01h, as a value of 00h is reserved	HwInit
31:24	**Target Port Number** – This field specifies the Port Number associated with the element targeted by this Link entry; the target Port Number is with respect to the Root Complex Component (identified by the Target Component ID) that contains the target element.	HwInit

7.13.3.2. Link Address

The Link address is a HwInit field located at offset 08h from the start of a Link Entry that identifies the target element for the Link entry. For a Link of Link Type 0 in its Link Description, the Link address specifies the memory-mapped base address of RCRB. For a Link of Link Type 1 in its Link Description, the Link address specifies the Configuration Space address of a PCI Express Root Port or a Root Complex Integrated Endpoint.

[103] The memory-mapped space for accessing an RCRB is not the same as Memory Space, and must not overlap with Memory Space.

7.13.3.2.1. Link Address for Link Type 0

For a Link pointing to a memory-mapped RCRB (Link Type bit = 0), the first DWORD specifies the lower 32 bits of the RCRB base address of the target element as shown below; bits 11:0 are hardwired to 000h and reserved for future use. The second DWORD specifies the high order 32 bits (63:32) of the base address of the target element.

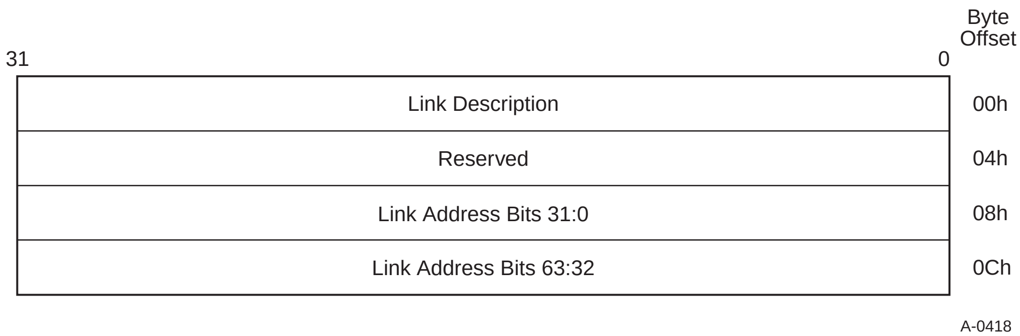

Figure 7-63: Link Address for Link Type 0

7.13.3.2.2. Link Address for Link Type 1

For a Link pointing to the Configuration Space of a Root Complex element (Link Type bit = 1), bits in the first DWORD specify the Bus, Device, and Function Number of the target element. As shown in Figure 7-62, bits 2:0 (N) encode the number of bits n associated with the Bus Number, with N = 000b specifying $n = 8$ and all other encodings specifying *n* = <value of N>. Bits 11:3 are reserved and hardwired to 0. Bits 14:12 specify the Function Number, and bits 19:15 specify the Device Number. Bits $(19 + n)$:20 specify the Bus Number, with $1 \leq n \leq 8$.

Bits 31:$(20 + n)$ of the first DWORD together with the second DWORD optionally identify the target element's hierarchy for systems implementing the PCI Express Enhanced Configuration Access Mechanism by specifying bits 63:$(20 + n)$ of the memory-mapped Configuration Space base address of the PCI Express hierarchy associated with the targeted element; single hierarchy systems that do not implement more than one memory mapped Configuration Space are allowed to report a value of zero to indicate default Configuration Space.

A Configuration Space base address [63:$(20 + n)$] equal to zero indicates that the Configuration Space address defined by bits $(19 + n)$:12 (Bus Number, Device Number, and Function Number) exists in the default PCI Segment Group; any non-zero value indicates a separate Configuration Space base address.

Software must not use *n* outside the context of evaluating the Bus Number and memory-mapped Configuration Space base address for this specific target element. In particular, *n* does not necessarily indicate the maximum Bus Number supported by the associated PCI Segment Group.

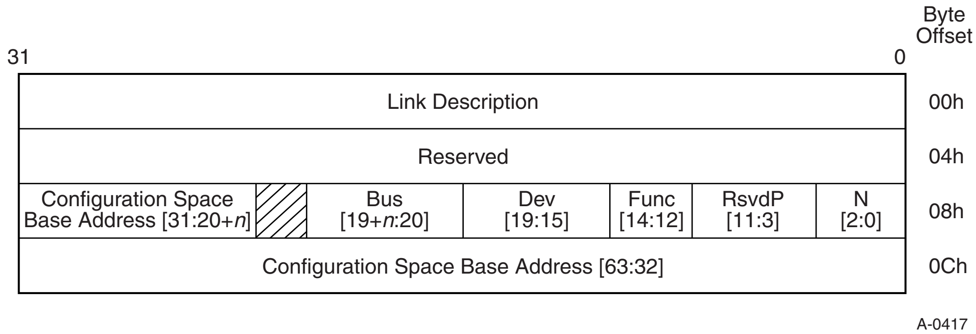

Figure 7-64: Link Address for Link Type 1

Table 7-57: Link Address for Link Type 1

Bit Location	Register Description	Attributes
2:0	**N** – Encoded number of Bus Number bits	HwInit
14:12	**Function Number**	HwInit
19:15	**Device Number**	HwInit
(19 + *n*):20	**Bus Number**	HwInit
63:(20 + *n*)	**PCI Express Configuration Space Base Address** ($1 \leq n \leq 8$) Note: A Root Complex that does not implement multiple Configuration Spaces is allowed to report this field as 0.	HwInit

7.14. PCI Express Root Complex Internal Link Control Capability

The PCI Express Root Complex Internal Link Control Capability is an optional Capability that controls an internal Root Complex Link between two distinct Root Complex Components. This Capability is valid for RCRBs that declare an Element Type field as Internal Root Complex Link in the Element Self-Description register of the Root Complex Link Declaration Capability structure.

The Root Complex Internal Link Control Capability structure is defined as shown in Figure 7-65.

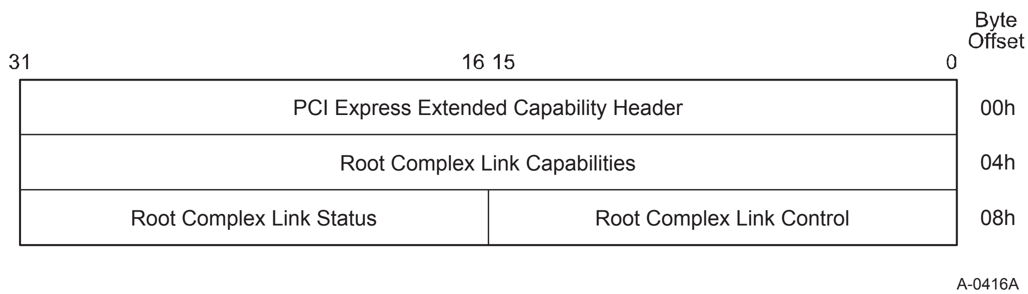

Figure 7-65: Root Complex Internal Link Control Capability

7.14.1. Root Complex Internal Link Control Extended Capability Header

The Extended Capability ID for the Root Complex Internal Link Control Capability is 0006h.

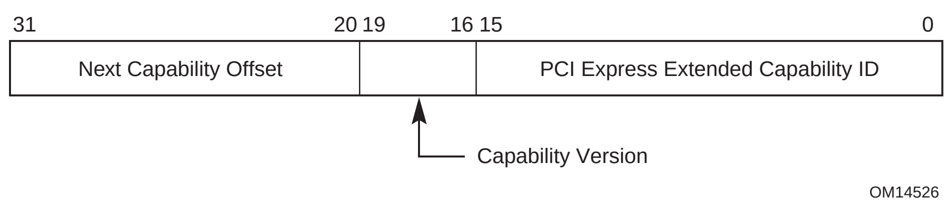

Figure 7-66: Root Internal Link Control Extended Capability Header

Table 7-58: Root Complex Internal Link Control Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Link Declaration Capability is 0006h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits.	RO

7.14.2. Root Complex Link Capabilities Register

The Root Complex Link Capabilities register identifies capabilities for this Link.

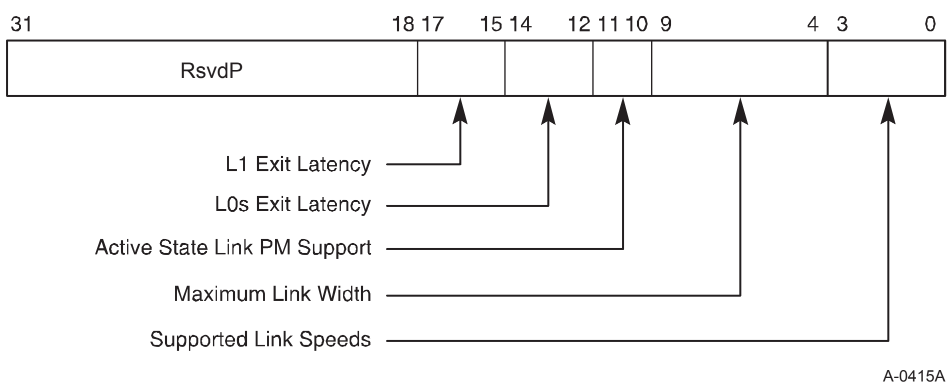

Figure 7-67: Root Complex Link Capabilities Register

Table 7-59: Root Complex Link Capabilities Register

Bit Location	Register Description	Attributes
3:0	**Supported Link Speeds** – This field indicates the supported Link speed(s) of the associated Link. Defined encodings are: 0001b 2.5 GT/s Link speed supported 0010b 5.0 GT/s and 2.5 GT/s Link speeds supported All other encodings are reserved. A Root Complex that does not support this feature must report 0000b in this field.	RO
9:4	**Maximum Link Width** – This field indicates the maximum width of the given Link. Defined encodings are: 00 0001b x1 00 0010b x2 00 0100b x4 00 1000b x8 00 1100b x12 01 0000b x16 10 0000b x32 All other encodings are reserved. A Root Complex that does not support this feature must report 00 0000b in this field.	RO
11:10	**Active State Power Management (ASPM) Support** – This field indicates the level of ASPM supported on the given Link. Defined encodings are: 00b No ASPM Support 01b L0s Entry Supported 10b L1 Entry Supported 11b L0s and L1 Supported	RO

Bit Location	Register Description	Attributes
14:12	**L0s Exit Latency** – This field indicates the L0s exit latency for the given Link. The value reported indicates the length of time this Port requires to complete transition from L0s to L0. Defined encodings are: 000b Less than 64 ns 001b 64 ns to less than 128 ns 010b 128 ns to less than 256 ns 011b 256 ns to less than 512 ns 100b 512 ns to less than 1 μs 101b 1 μs to less than 2 μs 110b 2 μs to 4 μs 111b More than 4 μs	RO
17:15	**L1 Exit Latency** – This field indicates the L1 exit latency for the given Link. The value reported indicates the length of time this Port requires to complete transition from L1 to L0. Defined encodings are: 000b Less than 1μs 001b 1 μs to less than 2 μs 010b 2 μs to less than 4 μs 011b 4 μs to less than 8 μs 100b 8 μs to less than 16 μs 101b 16 μs to less than 32 μs 110b 32 μs to 64 μs 111b More than 64 μs	RO

7.14.3. Root Complex Link Control Register

The Link Control register controls parameters for this internal Link.

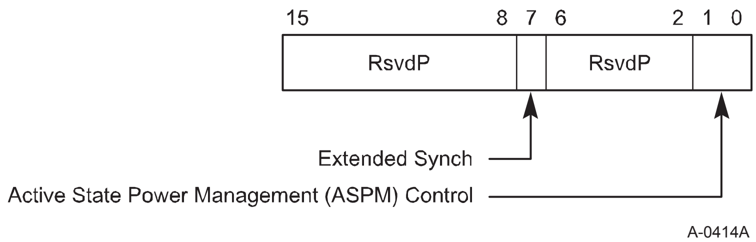

Figure 7-68: Root Complex Link Control Register

Table 7-60: Root Complex Link Control Register

Bit Location	Register Description	Attributes
1:0	**Active State Power Management (ASPM) Control** – This field controls the level of ASPM supported on the given Link. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled Note: "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). Default value of this field is implementation specific. ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. When disabling ASPM L1, software must disable ASPM L1 in the Downstream component on a Link prior to disabling ASPM L1 in the Upstream component on that Link. ASPM L1 must only be enabled on the Downstream component if both components on a Link support ASPM L1. A Root Complex that does not support this feature for the given internal Link must hardwire this field to 00b.	RW

Bit Location	Register Description	Attributes
7	**Extended Synch** – This bit when Set forces the transmission of additional Ordered Sets when exiting the L0s state (see Section 4.2.4.5) and when in the Recovery state (see Section 4.2.6.4.1). This mode provides external devices (e.g., logic analyzers) monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 state and resumes communication. A Root Complex that does not support this feature for the given internal Link must hardwire this bit to 0b. Default value for this bit is 0b.	RW

7.14.4. Root Complex Link Status Register

The Link Status register provides information about Link specific parameters.

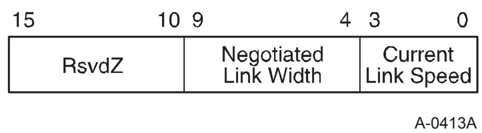

Figure 7-69: Root Complex Link Status Register

Table 7-61: Root Complex Link Status Register

Bit Location	Register Description	Attributes
3:0	**Current Link Speed** – This field indicates the negotiated Link speed of the given Link. Defined encodings are: 0001b 2.5 GT/s Link 0010b 5.0 GT/s Link All other encodings are reserved. The value in this field is undefined when the Link is not up. A Root Complex that does not support this feature must report 0000b in this field.	RO

Bit Location	Register Description	Attributes
9:4	**Negotiated Link Width** – This field indicates the negotiated width of the given Link. Defined encodings are: 00 0001b x1 00 0010b x2 00 0100b x4 00 1000b x8 00 1100b x12 01 0000b x16 10 0000b x32 All other encodings are reserved. The value in this field is undefined when the Link is not up. A Root Complex that does not support this feature must hardwire this field to 00 0000b.	RO

7.15. Power Budgeting Capability

The PCI Express Power Budgeting Capability allows the system to allocate power to devices that are added to the system at runtime. Through this Capability, a device can report the power it consumes on a variety of power rails, in a variety of device, power-management states, in a variety of operating conditions. The system uses this information to ensure that the system is capable of providing the proper power and cooling levels to the device. Failure to indicate proper device power consumption may risk device or system failure.

Implementation of the Power Budgeting Capability is optional for PCI Express devices that are implemented either in a form factor which does not require Hot-Plug support, or that are integrated on the system board. PCI Express form factor specifications may require support for power budgeting. Figure 7-70 details allocation of register fields in the Power Budgeting Capability structure.

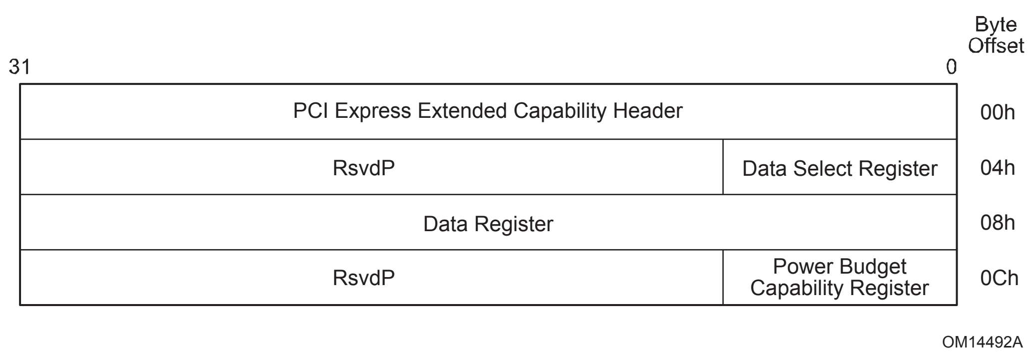

Figure 7-70: PCI Express Power Budgeting Capability Structure

7.15.1. Power Budgeting Extended Capability Header (Offset 00h)

Figure 7-71 details allocation of register fields in the Power Budgeting Extended Capability header; Table 7-62 provides the respective bit definitions. Refer to Section 7.9.3 for a description of the PCI Express Extended Capability header. The Extended Capability ID for the Power Budgeting Capability is 0004h.

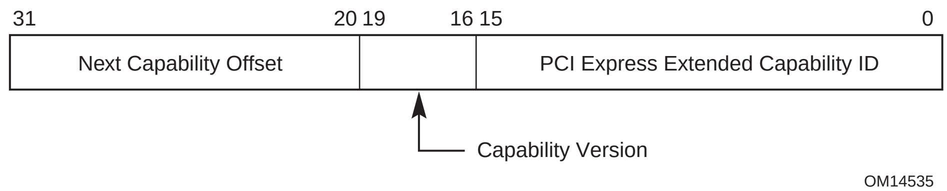

Figure 7-71: Power Budgeting Extended Capability Header

Table 7-62: Power Budgeting Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Power Budgeting Capability is 0004h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.15.2. Data Select Register (Offset 04h)

This read-write register indexes the Power Budgeting Data reported through the Data register and selects the DWORD of Power Budgeting Data that should appear in the Data register. Index values for this register start at zero to select the first DWORD of Power Budgeting Data; subsequent DWORDs of Power Budgeting Data are selected by increasing index values.

7.15.3. Data Register (Offset 08h)

This read-only register returns the DWORD of Power Budgeting Data selected by the Data Select register. Each DWORD of the Power Budgeting Data describes the power usage of the device in a particular operating condition. Power Budgeting Data for different operating conditions is not required to be returned in any particular order, as long as incrementing the Data Select register causes information for a different operating condition to be returned. If the Data Select register contains a value greater than or equal to the number of operating conditions for which the device provides power information, this register should return all zeros. Figure 7-72 details allocation of register fields in the Power Budgeting Data register; Table 7-63 provides the respective bit definitions.

The Base Power and Data Scale fields describe the power usage of the device; the Power Rail, Type, PM State, and PM Sub State fields describe the conditions under which the device has this power usage.

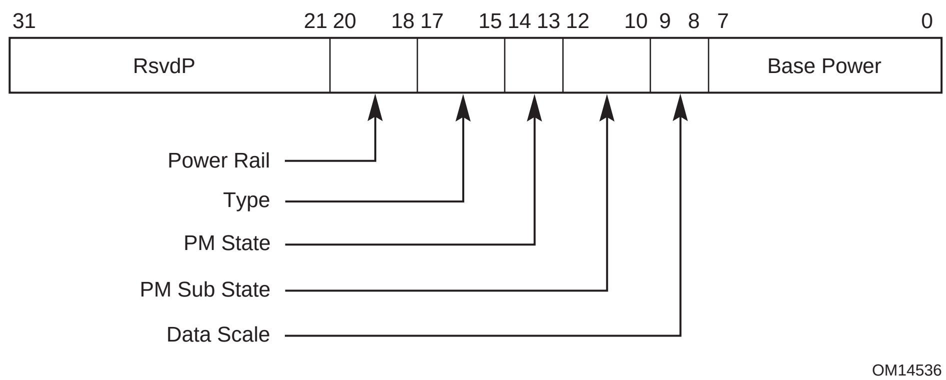

Figure 7-72: Power Budgeting Data Register

Table 7-63: Power Budgeting Data Register

Bit Location	Register Description	Attributes
7:0	**Base Power** – Specifies in watts the base power value in the given operating condition. This value must be multiplied by the data scale to produce the actual power consumption value except when the Data Scale field equals 00b (1.0x) and Base Power exceeds EFh, the following alternative encodings are used: F0h = 250 W Slot Power Limit F1h = 275 W Slot Power Limit F2h = 300 W Slot Power Limit F3h to FFh = reserved	RO

Bit Location	Register Description	Attributes
9:8	**Data Scale** – Specifies the scale to apply to the Base Power value. The power consumption of the device is determined by multiplying the contents of the Base Power field with the value corresponding to the encoding returned by this field, except as noted above. Defined encodings are: 00b 1.0x 01b 0.1x 10b 0.01x 11b 0.001x	RO
12:10	**PM Sub State** – Specifies the power management sub state of the operating condition being described. Defined encodings are: 000b Default Sub State 001b – 111b Device Specific Sub State	RO
14:13	**PM State** – Specifies the power management state of the operating condition being described. Defined encodings are: 00b D0 01b D1 10b D2 11b D3 A device returns 11b in this field and Aux or PME Aux in the Type register to specify the D3-Cold PM State. An encoding of 11b along with any other Type register value specifies the D3-Hot state.	RO
17:15	**Type** – Specifies the type of the operating condition being described. Defined encodings are: 000b PME Aux 001b Auxiliary 010b Idle 011b Sustained 111b Maximum All other encodings are reserved.	RO

Bit Location	Register Description	Attributes
20:18	**Power Rail** – Specifies the power rail of the operating condition being described. Defined encodings are: 000b Power (12V) 001b Power (3.3V) 010b Power (1.8V) 111b Thermal All other encodings are reserved.	RO

A device that implements the Power Budgeting Capability is required to provide data values for the D0 Max and D0 Sustained PM State/Type combinations for every power rail from which it consumes power; data for the D0 Max Thermal and D0 Sustained Thermal combinations must also be provided if these values are different from the values reported for D0 Max and D0 Sustained on the power rails.

Devices that support auxiliary power or PME from auxiliary power must provide data for the appropriate power type (Aux or PME Aux).

7.15.4. Power Budget Capability Register (Offset 0Ch)

This register indicates the power budgeting capabilities of a device. Figure 7-73 details allocation of register fields in the Power Budget Capability register; Table 7-64 provides the respective bit definitions.

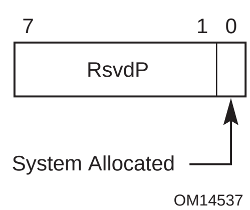

Figure 7-73: Power Budget Capability Register

Table 7-64: Power Budget Capability Register

Bit Location	Register Description	Attributes
0	**System Allocated** – When Set, this bit indicates that the power budget for the device is included within the system power budget. Reported Power Budgeting Data for this device must be ignored by software for power budgeting decisions if this bit is Set.	HwInit

7.16. ACS Extended Capability

The ACS Extended Capability is an optional capability that provides enhanced access controls (see Section 6.12). This capability may be implemented by a Root Port, a Switch Downstream Port, or a multi-Function device Function. It is never applicable to a PCI Express to PCI Bridge or Root Complex Event Collector. It is not applicable to a Switch Upstream Port unless that Switch Upstream Port is a Function in a multi-Function device.

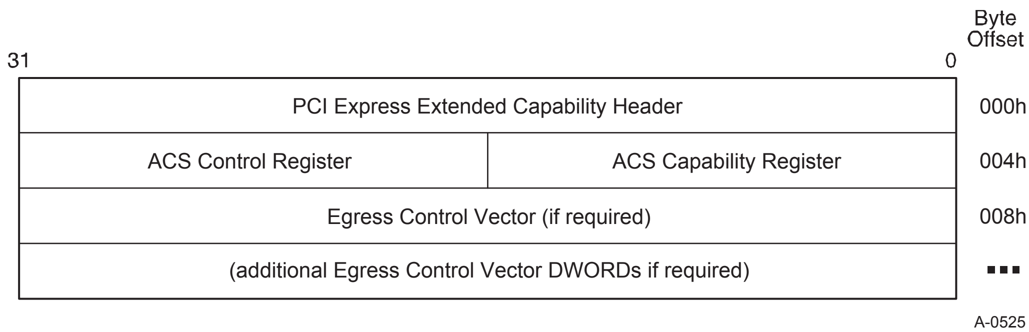

Figure 7-74: ACS Extended Capability

7.16.1. ACS Extended Capability Header (Offset 00h)

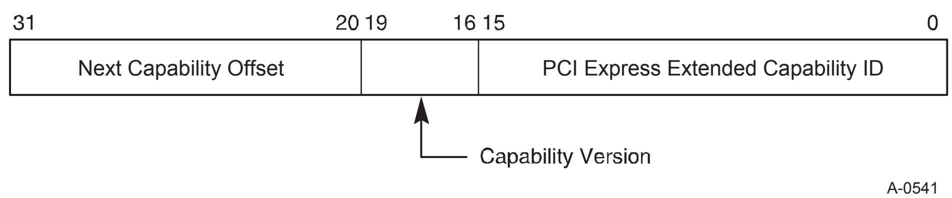

Figure 7-75: ACS Extended Capability Header

Table 7-65: ACS Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the ACS Extended Capability is 000Dh.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO

Bit Location	Register Description	Attributes
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities.	RO

7.16.2. ACS Capability Register (Offset 04h)

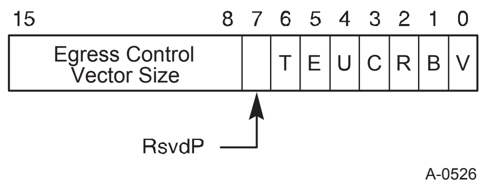

Figure 7-76: ACS Capability Register

Table 7-66: ACS Capability Register

Bit Location	Register Description	Attributes
0	**ACS Source Validation (V)** – Required for Root Ports and Switch Downstream Ports; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS Source Validation.	RO
1	**ACS Translation Blocking (B)** – Required for Root Ports and Switch Downstream Ports; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS Translation Blocking.	RO
2	**ACS P2P Request Redirect (R)** – Required for Root Ports that support peer-to-peer traffic with other Root Ports; required for Switch Downstream Ports; required for multi-Function device Functions that support peer-to-peer traffic with other Functions; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS P2P Request Redirect.	RO
3	**ACS P2P Completion Redirect (C)** – Required for all Functions that support ACS P2P Request Redirect; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS P2P Completion Redirect.	RO
4	**ACS Upstream Forwarding (U)** – Required for Root Ports if the RC supports Redirected Request Validation; required for Switch Downstream Ports; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS Upstream Forwarding.	RO
5	**ACS P2P Egress Control (E)** – Optional for Root Ports, Switch Downstream Ports, and multi-Function device Functions; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS P2P Egress Control.	RO

Bit Location	Register Description	Attributes
6	**ACS Direct Translated P2P (T)** – Required for Root Ports that support Address Translation Services (ATS) and also support peer-to-peer traffic with other Root Ports; required for Switch Downstream Ports; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS Direct Translated P2P.	RO
15:8	**Egress Control Vector Size** – Encodings 01h-FFh directly indicate the number of applicable bits in the Egress Control Vector; the encoding 00h indicates 256 bits. If the ACS P2P Egress Control (E) bit is 0b, the value of the size field is undefined, and the Egress Control Vector register is not required to be present.	HwInit

7.16.3. ACS Control Register (Offset 06h)

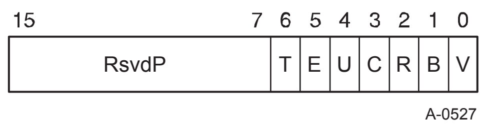

Figure 7-77: ACS Control Register

Table 7-67: ACS Control Register

Bit Location	Register Description	Attributes
0	**ACS Source Validation Enable (V)** – When Set, the component validates the Bus Number from the Requester ID of Upstream Requests against the secondary / subordinate Bus Numbers. Default value of this bit is 0b. Must be hardwired to 0b if the ACS Source Validation functionality is not implemented.	RW
1	**ACS Translation Blocking Enable (B)** – When Set, the component blocks all Upstream Memory Requests whose Address Translation (AT) field is not set to the default value. Default value of this bit is 0b. Must be hardwired to 0b if the ACS Translation Blocking functionality is not implemented.	RW

Bit Location	Register Description	Attributes
2	**ACS P2P Request Redirect Enable (R)** – In conjunction with ACS P2P Egress Control and ACS Direct Translated P2P mechanisms, determines when the component redirects peer-to-peer Requests Upstream (see Section 6.12.3). Note that with Downstream Ports, this bit only applies to Upstream Requests arriving at the Downstream Port, and whose normal routing targets a different Downstream Port. Default value of this bit is 0b. Must be hardwired to 0b if the ACS P2P Request Redirect functionality is not implemented.	RW
3	**ACS P2P Completion Redirect Enable (C)** – Determines when the component redirects peer-to-peer Completions Upstream; applicable only to Completions[104] whose Relaxed Ordering Attribute is clear. Default value of this bit is 0b. Must be hardwired to 0b if the ACS P2P Completion Redirect functionality is not implemented.	RW
4	**ACS Upstream Forwarding Enable (U)** – When Set, the component forwards Upstream any Request or Completion TLPs it receives that were redirected Upstream by a component lower in the hierarchy. Note that this bit only applies to Upstream TLPs arriving at a Downstream Port, and whose normal routing targets the same Downstream Port. Default value of this bit is 0b. Must be hardwired to 0b if the ACS Upstream Forwarding functionality is not implemented.	RW
5	**ACS P2P Egress Control Enable (E)** – In conjunction with the Egress Control Vector plus the ACS P2P Request Redirect and ACS Direct Translated P2P mechanisms, determines when to allow, disallow, or redirect peer-to-peer Requests (see Section 6.12.3). Default value of this bit is 0b. Must be hardwired to 0b if the ACS P2P Egress Control functionality is not implemented.	RW
6	**ACS Direct Translated P2P Enable (T)** – When Set, overrides the ACS P2P Request Redirect and ACS P2P Egress Control mechanisms with peer-to-peer Memory Requests whose Address Translation (AT) field indicates a Translated address (see Section 6.12.3). This bit is ignored if ACS Translation Blocking (B) is 1b. Default value of this bit is 0b. Must be hardwired to 0b if the ACS Direct Translated P2P functionality is not implemented.	RW

[104] This includes Read Completions, AtomicOp Completions, and other Completions with or without Data.

7.16.4. Egress Control Vector (Offset 08h)

The Egress Control Vector is a read-write register that contains a bit-array. The number of bits in the register is specified by the Egress Control Vector Size field, and the register spans multiple DWORDs if required. If the ACS P2P Egress Control bit in the ACS Capability register is 0b, the Egress Control Vector Size field is undefined and the Egress Control Vector register is not required to be present.

For the general case of an Egress Control Vector spanning multiple DWORDs, the DWORD offset and bit number within that DWORD for a given arbitrary bit *K* are specified by the formulas:

DWORD offset = 08h + (K div[105] 32) * 4

DWORD bit# = K mod[106] 32

Bits in a DWORD beyond those specified by the Egress Control Vector Size field are RsvdP.

For Root Ports and Switch Downstream Ports, each bit in the bit-array always corresponds to a Port Number. Otherwise, for Functions[107] within a multi-Function device, each bit in the bit-array corresponds to one or more Function Numbers, or a Function Group Number. For example, access to Function 2 is controlled by bit number 2 in the bit-array. For both Port Number cases and Function Number cases, the bit corresponding to the Function that implements this Extended Capability structure must be hardwired to 0b.[108]

If an ARI Device implements ACS Function Groups, its Egress Control Vector Size is required to be a power-of-2 between 8 and 256, and all of its implemented Egress Control Vector bits must be R/W. With ARI Devices, multiple Functions can be associated with a single bit, so for each Function, its associated bit determines how Requests from it targeting other Functions (if any) associated with the same bit are handled.

If ACS Function Groups are enabled in an ARI Device, the first 8 Egress Control Vector bits in each Function are associated with Function Group Numbers instead of Function Numbers. In this case, access control is enforced between Function Groups instead of Functions, and any implemented Egress Control Vector bits beyond the first 8 are unused.

Independent of whether an ARI Device implements ACS Function Groups, its Egress Control Vector Size is not required to cover the entire Function Number range of all Functions implemented by the Device. If ACS Function Groups are not enabled, Function Numbers are mapped to implemented Egress Control Vector bits by taking the modulo of the Egress Control Vector Size, which is constrained to be a power-of-2.

With RCs, some Port Numbers may refer to internal Ports instead of Root Ports. For Root Ports in such RCs, each bit in the bit-array that corresponds to an internal Port must be hardwired to 0b.

---

[105] Div is an integer divide with truncation.

[106] Mod is the remainder from an integer divide.

[107] Including Switch Upstream Ports.

[108] For ARI Devices, the bit must be R/W. See subsequent description.

Figure 7-78: Egress Control Vector Register

Table 7-68: Egress Control Vector

Bit Location	Register Description	Attributes
N-1:0	**Egress Control Vector** – An N-bit bit-array configured by software. When a given bit is set, peer-to-peer Requests targeting the associated Port, Function, or Function Group are blocked or redirected (if enabled) (see Section 6.12.3). Default value of this field is 0b.	RW

The following examples illustrate how the vector might be configured:

- For an 8-Port Switch, each Port will have a separate vector indicating which Downstream Egress Ports it may forward Requests to.
    - Port 1 being not allowed to communicate with any other Downstream Ports would be configured as: 1111 1100b with 0b indicating in bit 0 corresponds to the Upstream Port and a 0b in bit 1 represents the Ingress Port hardwired to 0b as well.
    - Port 2 being allowed to communicate with Ports 3, 5, and 7 would be configured as: 0101 0010b.
- For a 4-Function device, each Function will have a separate vector that indicates which Function it may forward Requests to.
    - Function 0 being not allowed to communicate with any other Functions would be configured as: 1110b with 0b in bit 0 corresponding to Function 0.
    - Function 1 being allowed to communicate with Functions 2 and 3 would be configured as: 0001b with a 0b in bit 1 corresponding to Function 1 hardwired to 0b as well.

7.17. PCI Express Root Complex Event Collector Endpoint Association Capability

The PCI Express Root Complex Event Collector Endpoint Association Capability is implemented by Root Complex Event Collectors.

It declares the Root Complex Integrated Endpoints supported by the Root Complex Event Collector on the same Logical Bus on which the Root Complex Event Collector is located. A Root Complex Event Collector must implement the Root Complex Event Collector Endpoint Association Capability; no other PCI Express device Function is permitted to implement this Capability.

The PCI Express Root Complex Event Collector Endpoint Association Capability, as shown in Figure 7-79, consists of the PCI Express Extended Capability header followed by a DWORD bitmap enumerating Root Complex Integrated Endpoints associated with the Root Complex Event Collector.

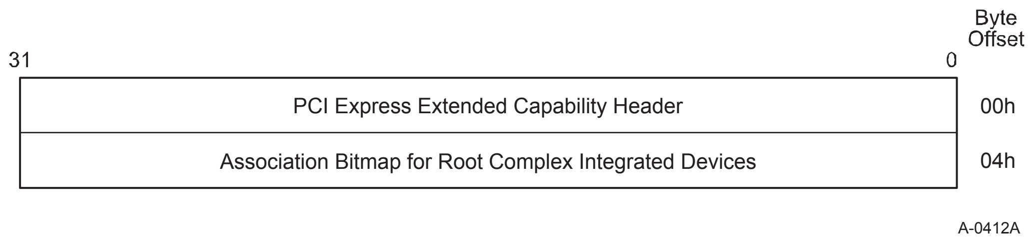

Figure 7-79: Root Complex Event Collector Endpoint Association Capability

7.17.1. Root Complex Event Collector Endpoint Association Extended Capability Header

The Extended Capability ID for the Root Complex Event Collector Endpoint Association Capability is 0007h. Figure 7-80 details allocation of fields in the Root Complex Event Collector Endpoint Association Extended Capability header; Table 7-69 provides the respective bit definitions.

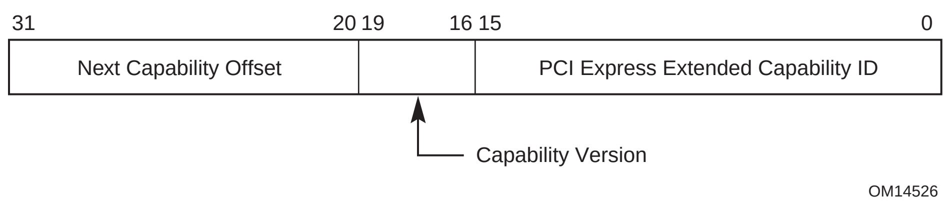

Figure 7-80: Root Complex Event Collector Endpoint Association Extended Capability Header

Table 7-69: Root Complex Event Collector Endpoint Association Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the Root Complex Event Collector Endpoint Association Capability is 0007h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO

Bit Location	Register Description	Attributes
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are reserved and must be implemented as 00b although software must mask them to allow for future uses of these bits.	RO

7.17.2. Association Bitmap for Root Complex Integrated Endpoints

The Association Bitmap for Root Complex Integrated Endpoints is a read-only register that sets the bits corresponding to the Device Numbers of Root Complex Integrated Endpoints supported by the Root Complex Event Collector on the same Logical Bus as the Event Collector itself. The bit corresponding to the Device Number of the Root Complex Event Collector must always be Set.

7.18. Multi-Function Virtual Channel Capability

The Multi-Function Virtual Channel (MFVC) Capability is an optional Extended Capability that permits enhanced QoS management in a multi-Function device, including TC/VC mapping, optional VC arbitration, and optional Function arbitration for Upstream Requests. When implemented, the MFVC Capability structure must be present in the Extended Configuration Space of Function 0 of the multi-Function device's Upstream Port. Figure 7-81 provides a high level view of the MFVC Capability structure. This MFVC Capability structure controls Virtual Channel assignment at the PCI Express Upstream Port of the multi-Function device, while a VC Capability structure, if present in a Function, controls the Virtual Channel assignment for that individual Function.

A multi-Function device is permitted to have an MFVC Capability structure even if none of its Functions have a VC Capability structure. However, an MFVC Capability structure is permitted only in Function 0 in the Upstream Port of a multi-Function device.

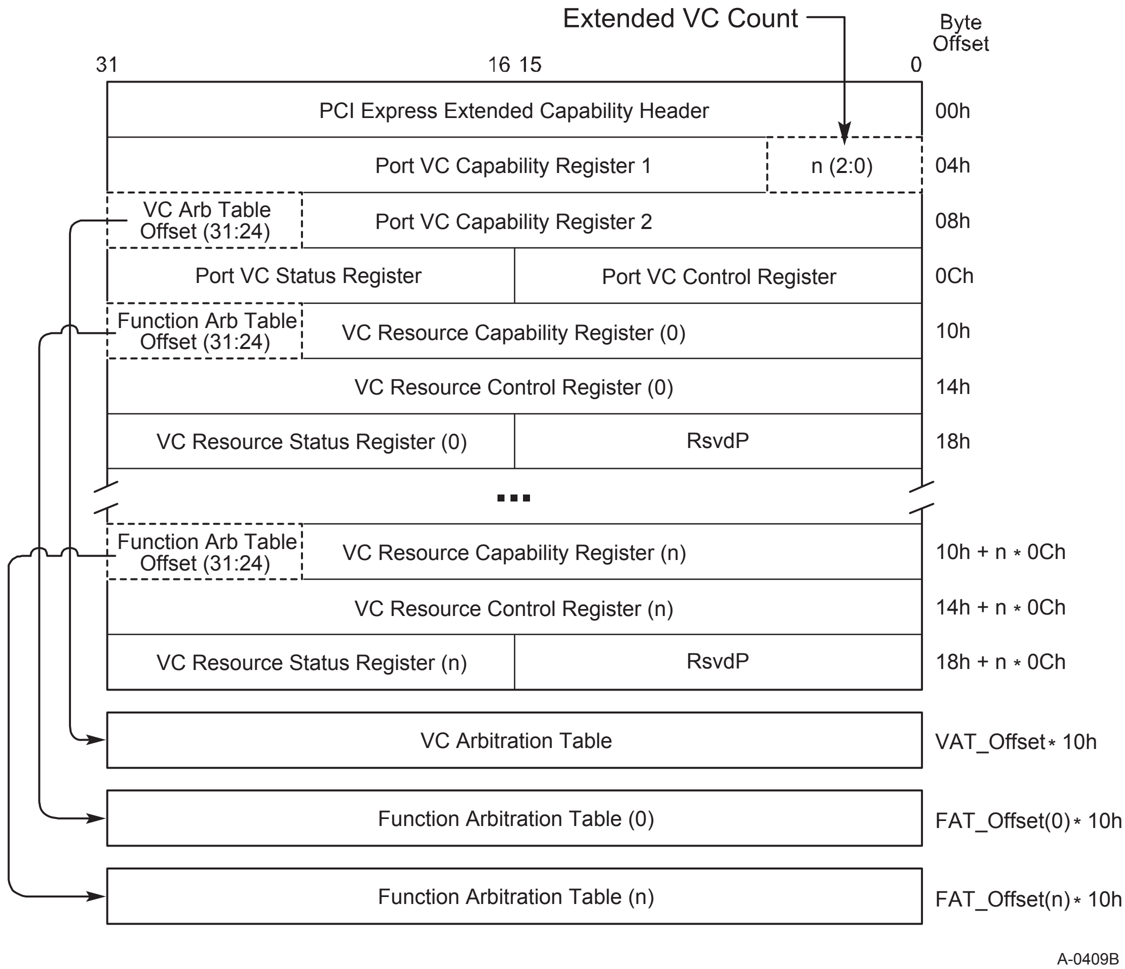

Figure 7-81: PCI Express MFVC Capability Structure

The following sections describe the registers/fields of the PCI Express MFVC Capability structure.

7.18.1. MFVC Extended Capability Header

Refer to Section 7.9.3 for a description of the PCI Express Extended Capability header. The Extended Capability ID for the MFVC Capability is 0008h. Figure 7-82 details allocation of register fields in the MFVC Extended Capability header; Table 7-70 provides the respective bit definitions.

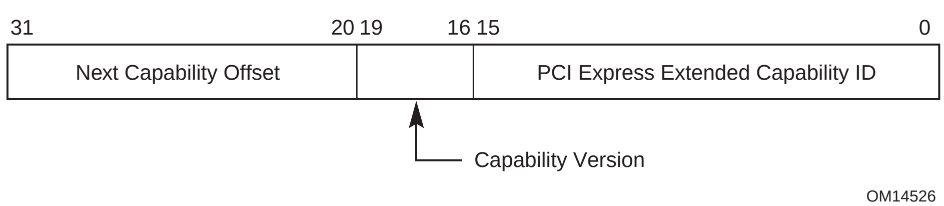

Figure 7-82: MFVC Extended Capability Header

Table 7-70: MFVC Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. The Extended Capability ID for the MFVC Capability is 0008h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.18.2. Port VC Capability Register 1

The Port VC Capability Register 1 describes the configuration of the Virtual Channels associated with a PCI Express Port of the multi-Function device. Figure 7-83 details allocation of register fields in the Port VC Capability Register 1; Table 7-71 provides the respective bit definitions.

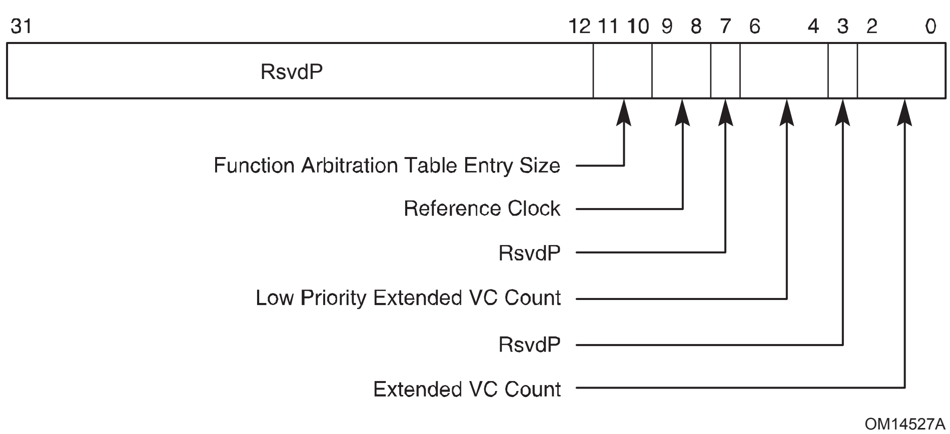

Figure 7-83: Port VC Capability Register 1

Table 7-71: Port VC Capability Register 1

Bit Location	Register Description	Attributes
2:0	**Extended VC Count** – Indicates the number of (extended) Virtual Channels in addition to the default VC supported by the device. The minimum value of this field is zero (for devices that only support the default VC). The maximum value is seven.	RO
6:4	**Low Priority Extended VC Count** – Indicates the number of (extended) Virtual Channels in addition to the default VC belonging to the low-priority VC (LPVC) group that has the lowest priority with respect to other VC resources in a strict-priority VC Arbitration. The minimum value of this field is 000b and the maximum value is Extended VC Count.	RO
9:8	**Reference Clock** – Indicates the reference clock for Virtual Channels that support time-based WRR Function Arbitration. Defined encodings are: 00b 100 ns reference clock 01b – 11b Reserved	RO
11:10	**Function Arbitration Table Entry Size** – Indicates the size (in bits) of Function Arbitration table entry in the device. Defined encodings are: 00b Size of Function Arbitration table entry is 1 bit 01b Size of Function Arbitration table entry is 2 bits 10b Size of Function Arbitration table entry is 4 bits 11b Size of Function Arbitration table entry is 8 bits	RO

7.18.3. Port VC Capability Register 2

The Port VC Capability Register 2 provides further information about the configuration of the Virtual Channels associated with a PCI Express Port of the multi-Function device. Figure 7-84 details allocation of register fields in the Port VC Capability Register 2; Table 7-72 provides the respective bit definitions.

Figure 7-84: Port VC Capability Register 2

Table 7-72: Port VC Capability Register 2

Bit Location	Register Description	Attributes
7:0	**VC Arbitration Capability** – Indicates the types of VC Arbitration supported by the device for the LPVC group. This field is valid for all devices that report a Low Priority Extended VC Count greater than 0. Each bit location within this field corresponds to a VC Arbitration Capability defined below. When more than 1 bit in this field is Set, it indicates that the device can be configured to provide different VC arbitration services. Defined bit positions are: Bit 0 – Hardware fixed arbitration scheme, e.g., Round Robin Bit 1 – Weighted Round Robin (WRR) arbitration with 32 phases Bit 2 – WRR arbitration with 64 phases Bit 3 – WRR arbitration with 128 phases Bits 4-7 – Reserved	RO
31:24	**VC Arbitration Table Offset** – Indicates the location of the VC Arbitration Table. This field contains the zero-based offset of the table in DQWORDS (16 bytes) from the base address of the MFVC Capability structure. A value of 00h indicates that the table is not present.	RO

7.18.4. Port VC Control Register

Figure 7-85 details allocation of register fields in the Port VC Control register; Table 7-73 provides the respective bit definitions.

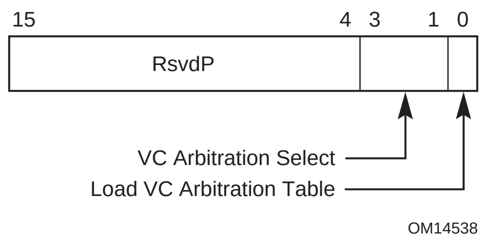

Figure 7-85: Port VC Control Register

Table 7-73: Port VC Control Register

Bit Location	Register Description	Attributes
0	**Load VC Arbitration Table** – Used by software to update the VC Arbitration Table. This bit is valid when the selected VC Arbitration uses the VC Arbitration Table. Software Sets this bit to request hardware to apply new values programmed into VC Arbitration Table; Clearing this bit has no effect. Software checks the VC Arbitration Table Status field to confirm that new values stored in the VC Arbitration Table are latched by the VC arbitration logic. This bit always returns 0b when read.	RW
3:1	**VC Arbitration Select** – Used by software to configure the VC arbitration by selecting one of the supported VC Arbitration schemes indicated by the VC Arbitration Capability field in the Port VC Capability Register 2. The permissible values of this field are numbers corresponding to one of the asserted bits in the VC Arbitration Capability field. This field cannot be modified when more than one VC in the LPVC group is enabled.	RW

7.18.5. Port VC Status Register

The Port VC Status register provides status of the configuration of Virtual Channels associated with a Port of the multi-Function device. Figure 7-86 details allocation of register fields in the Port VC Status register; Table 7-74 provides the respective bit definitions.

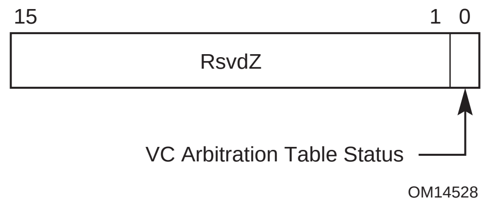

Figure 7-86: Port VC Status Register

Table 7-74: Port VC Status Register

Bit Location	Register Description	Attributes
0	**VC Arbitration Table Status** – Indicates the coherency status of the VC Arbitration Table. This bit is valid when the selected VC uses the VC Arbitration Table. This bit is Set by hardware when any entry of the VC Arbitration Table is written by software. This bit is Cleared by hardware when hardware finishes loading values stored in the VC Arbitration Table after software sets the Load VC Arbitration Table field in the Port VC Control register. Default value of this bit is 0b.	RO

7.18.6. VC Resource Capability Register

The VC Resource Capability register describes the capabilities and configuration of a particular Virtual Channel resource. Figure 7-87 details allocation of register fields in the VC Resource Capability register; Table 7-75 provides the respective bit definitions.

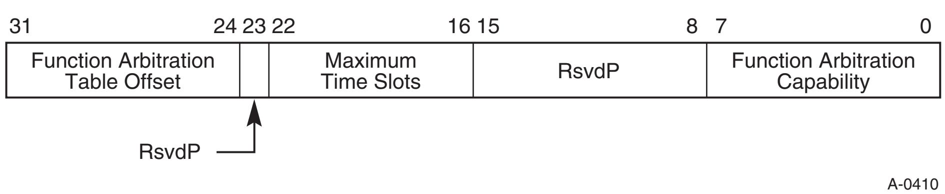

Figure 7-87: VC Resource Capability Register

Table 7-75: VC Resource Capability Register

Bit Location	Register Description	Attributes
7:0	**Function Arbitration Capability** – Indicates types of Function Arbitration supported by the VC resource. Each bit location within this field corresponds to a Function Arbitration Capability defined below. When more than 1 bit in this field is Set, it indicates that the VC resource can be configured to provide different arbitration services. Software selects among these capabilities by writing to the Function Arbitration Select field (see below). Defined bit positions are: Bit 0 — Non-configurable hardware-fixed arbitration scheme, e.g., Round Robin (RR) Bit 1 — Weighted Round Robin (WRR) arbitration with 32 phases Bit 2 — WRR arbitration with 64 phases Bit 3 — WRR arbitration with 128 phases Bit 4 — Time-based WRR with 128 phases Bit 5 — WRR arbitration with 256 phases Bits 6-7 — Reserved	RO
22:16	**Maximum Time Slots** – Indicates the maximum number of time slots (minus 1) that the VC resource is capable of supporting when it is configured for time-based WRR Function Arbitration. For example, a value of 000 0000b in this field indicates the supported maximum number of time slots is 1 and a value of 111 1111b indicates the supported maximum number of time slot is 128. This field is valid only when the Function Arbitration Capability indicates that the VC resource supports time-based WRR Function Arbitration.	HwInit
31:24	**Function Arbitration Table Offset** – Indicates the location of the Function Arbitration Table associated with the VC resource. This field contains the zero-based offset of the table in DQWORDS (16 bytes) from the base address of the MFVC Capability structure. A value of 00h indicates that the table is not present.	RO

7.18.7. VC Resource Control Register

Figure 7-88 details allocation of register fields in the VC Resource Control register; Table 7-76 provides the respective bit definitions.

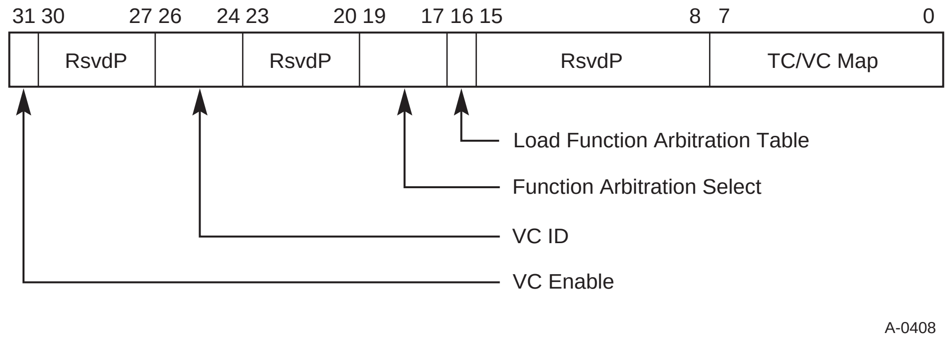

Figure 7-88: VC Resource Control Register

Table 7-76: VC Resource Control Register

Bit Location	Register Description	Attributes
7:0	**TC/VC Map** – This field indicates the TCs that are mapped to the VC resource. Bit locations within this field correspond to TC values. For example, when bit 7 is Set in this field, TC7 is mapped to this VC resource. When more than 1 bit in this field is Set, it indicates that multiple TCs are mapped to the VC resource. In order to remove one or more TCs from the TC/VC Map of an enabled VC, software must ensure that no new or outstanding transactions with the TC labels are targeted at the given Link. Default value of this field is FFh for the first VC resource and is 00h for other VC resources. Note: Bit 0 of this field is read-only. It must be hardwired to 1b for the default VC0 and hardwired to 0b for all other enabled VCs.	RW (see the note for exceptions)

Bit Location	Register Description	Attributes
16	**Load Function Arbitration Table** – When Set, this bit updates the Function Arbitration logic from the Function Arbitration Table for the VC resource. This field is only valid when the Function Arbitration Table is used by the selected Function Arbitration scheme (that is indicated by a Set bit in the Function Arbitration Capability field selected by Function Arbitration Select). Software sets this bit to signal hardware to update Function Arbitration logic with new values stored in the Function Arbitration Table; clearing this bit has no effect. Software uses the Function Arbitration Table Status bit to confirm whether the new values of Function Arbitration Table are completely latched by the arbitration logic. This bit always returns 0b when read. Default value of this bit is 0b.	RW
19:17	**Function Arbitration Select** – This field configures the VC resource to provide a particular Function Arbitration service. The permissible value of this field is a number corresponding to one of the asserted bits in the Function Arbitration Capability field of the VC resource.	RW
26:24	**VC ID** – This field assigns a VC ID to the VC resource (see note for exceptions). This field cannot be modified when the VC is already enabled. Note: For the first VC resource (default VC), this field is a read-only field that must be hardwired to 000b.	RW

Bit Location	Register Description	Attributes
31	**VC Enable** – When Set, this bit enables a Virtual Channel (see note 1 for exceptions). The Virtual Channel is disabled when this bit is cleared. Software must use the VC Negotiation Pending bit to check whether the VC negotiation is complete. Default value of this bit is 1b for the first VC resource and 0b for other VC resource(s). Notes: 1. This bit is hardwired to 1b for the default VC (VC0), i.e., writing to this field has no effect for VC0. 2. To enable a Virtual Channel, the VC Enable bits for that Virtual Channel must be Set in both components on a Link. 3. To disable a Virtual Channel, the VC Enable bits for that Virtual Channel must be Cleared in both components on a Link. 4. Software must ensure that no traffic is using a Virtual Channel at the time it is disabled. 5. Software must fully disable a Virtual Channel in both components on a Link before re-enabling the Virtual Channel.	RW

7.18.8. VC Resource Status Register

Figure 7-89 details allocation of register fields in the VC Resource Status register; Table 7-77 provides the respective bit definitions.

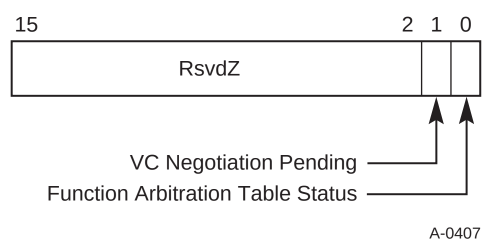

Figure 7-89: VC Resource Status Register

Table 7-77: VC Resource Status Register

Bit Location	Register Description	Attributes
0	**Function Arbitration Table Status** – This bit indicates the coherency status of the Function Arbitration Table associated with the VC resource. This bit is valid only when the Function Arbitration Table is used by the selected Function Arbitration for the VC resource. This bit is Set by hardware when any entry of the Function Arbitration Table is written to by software. This bit is cleared by hardware when hardware finishes loading values stored in the Function Arbitration Table after software sets the Load Function Arbitration Table bit. Default value of this bit is 0b.	RO
1	**VC Negotiation Pending** – This bit indicates whether the Virtual Channel negotiation (initialization or disabling) is in pending state. When this bit is Set by hardware, it indicates that the VC resource is still in the process of negotiation. This bit is cleared by hardware after the VC negotiation is complete. For a non-default Virtual Channel, software may use this bit when enabling or disabling the VC. For the default VC, this bit indicates the status of the process of Flow Control initialization. Before using a Virtual Channel, software must check whether the VC Negotiation Pending bits for that Virtual Channel are Clear in both components on a Link.	RO

7.18.9. VC Arbitration Table

The definition of the VC Arbitration Table in the MFVC Capability structure is identical to that in the VC Capability structure (see Section 7.11.9).

7.18.10. Function Arbitration Table

The Function Arbitration Table register in the MFVC Capability structure takes the same form as the Port Arbitration Table register in the VC Capability structure (see Section 7.11.10).

The Function Arbitration Table register is a read-write register array that is used to store the WRR or time-based WRR arbitration table for Function Arbitration for the VC resource. It is only present when one or more asserted bits in the Function Arbitration Capability field indicate that the multi-Function device supports a Function Arbitration scheme that uses a programmable arbitration table. Furthermore, it is only valid when one of the above-mentioned bits in the Function Arbitration Capability field is selected by the Function Arbitration Select field.

The Function Arbitration Table represents one Function arbitration period. Each table entry containing a Function Number or Function Group[109] Number corresponds to a phase within a Function Arbitration period. The table entry size requirements are as follows:

- The table entry size for non-ARI devices must support enough values to specify all implemented Functions plus at least one value that does not correspond to an implemented Function. For example, a table with 2-bit entries can be used by a multi-Function device with up to three Functions.
- The table entry size for ARI Devices must be either 4 bits or 8 bits.
  - If MFVC Function Groups are enabled, each entry maps to a single Function Group. Arbitration between multiple Functions within a Function Group is implementation specific, but must guarantee forward progress.
  - If MFVC Function Groups are not enabled and 4-bit entries are implemented, a given entry maps to all Functions whose Function Number modulo 8 matches its value. Similarly, if 8-bit entries are implemented, a given entry maps to all Functions whose Function Number modulo 128 matches its value. If a given entry maps to multiple Functions, arbitration between those Functions is implementation specific, but must guarantee forward progress.

A Function Number or Function Group Number written to a table entry indicates that the phase within the Function Arbitration period is assigned to the selected Function or Function Group (the Function Number or Function Group Number must be a valid one).

- When the WRR Function Arbitration is used for a VC of the Egress Port of the multi-Function device, at each arbitration phase the Function Arbiter serves one transaction from the Function or Function Group indicated by the Function Number or Function Goup Number of the current phase. When finished, it immediately advances to the next phase. A phase is skipped, i.e., the Function Arbiter simply moves to the next phase immediately if the Function or Function Group indicated by the phase does not contain any transaction for the VC.
- When the Time-based WRR Function Arbitration is used for a VC of the Egress Port of the multi-Function device, at each arbitration phase aligning to a virtual timeslot, the Function Arbiter serves one transaction from the Function or Function Group indicated by the Function Number or Function Group Number of the current phase. It advances to the next phase at the next virtual timeslot. A phase indicates an "idle" timeslot, i.e., the Function Arbiter does not serve any transaction during the phase, if
  - the phase contains the Number of a Function or a Function Group that does not exist, or
  - the Function or Function Group indicated by the phase does not contain any transaction for the VC.

The Function Arbitration Table Entry Size field in the Port VC Capability Register 1 determines the table entry size. The length of the table is determined by the Function Arbitration Select field as shown in Table 7-78.

When the Function Arbitration Table is used by the default Function Arbitration for the default VC, the default values for the table entries must contain at least one entry for each of the active

[109] If an ARI Device supports MFVC Function Groups capability and ARI-aware software enables it, arbitration is based on Function Groups instead of Functions. See Section 7.23.

Functions or Function Groups in the multi-Function device to ensure forward progress for the default VC for the multi-Function device's Upstream Port. The table may contain RR or RR-like fair Function Arbitration for the default VC.

Table 7-78: Length of Function Arbitration Table

Function Arbitration Select	Function Arbitration Table Length (in Number of Entries)
001b	32
010b	64
011b	128
100b	128
101b	256

7.19. Vendor-Specific Capability

The PCI Express Vendor-Specific Extended Capability (VSEC) is an optional Extended Capability that is permitted to be implemented by any PCI Express Function or RCRB. This allows PCI Express component vendors to use the Extended Capability mechanism to expose vendor-specific registers.

A single PCI Express Function or RCRB is permitted to contain multiple VSEC structures.

An example usage is a set of vendor-specific features that are intended to go into an on-going series of components from that vendor. A VSEC structure can tell vendor-specific software which features a particular component supports, including components developed after the software was released.

Figure 7-90details allocation of register fields in the VSEC structure. The structure of the PCI Express Extended Capability header and the Vendor-Specific header is architected by this specification.

With a PCI Express Function, the structure and definition of the Vendor-Specific Registers area is determined by the vendor indicated by the Vendor ID field located at byte offset 00h in PCI-compatible Configuration Space. With an RCRB, a VSEC is permitted only if the RCRB also contains an RCRB Header Capability structure, which contains a Vendor ID field indicating the vendor.

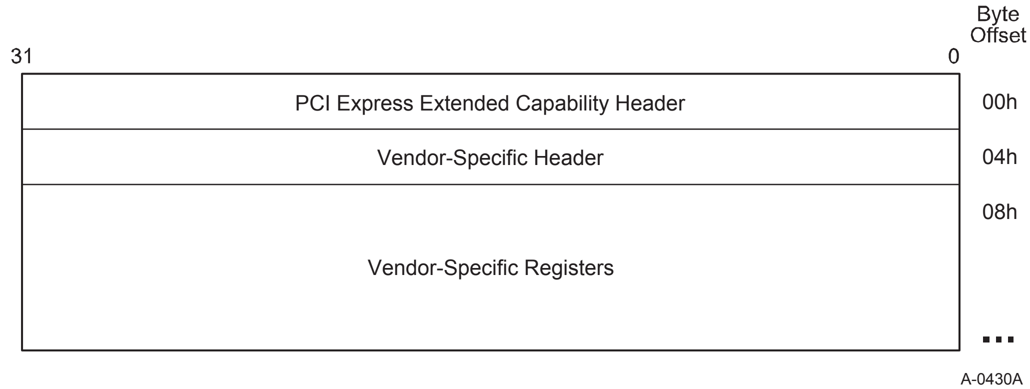

Figure 7-90: PCI Express VSEC Structure

7.19.1. Vendor-Specific Extended Capability Header (Offset 00h)

Figure 7-91 details allocation of register fields in the Vendor-Specific Extended Capability header; Table 7-79 provides the respective bit definitions. Refer to Section 7.9.3 for a description of the PCI Express Extended Capability header. The Extended Capability ID for the Vendor-Specific Capability is 000Bh.

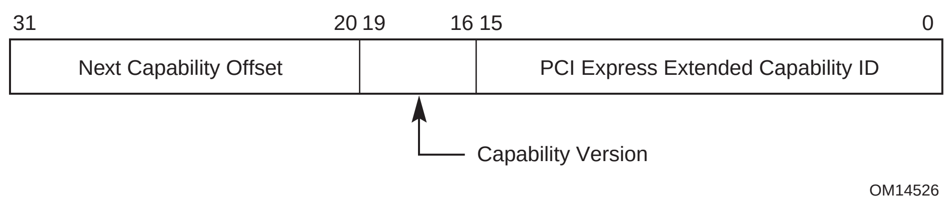

Figure 7-91: Vendor-Specific Extended Capability Header

Table 7-79: Vendor-Specific Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the Vendor-Specific Capability is 000Bh.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO

Bit Location	Register Description	Attributes
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.19.2. Vendor-Specific Header (Offset 04h)

Figure 7-92 details allocation of register fields in the Vendor-Specific header; Table 7-80 provides the respective bit definitions.

Vendor-specific software must qualify the associated Vendor ID of the PCI Express Function or RCRB before attempting to interpret the values in the VSEC ID or VSEC Rev fields.

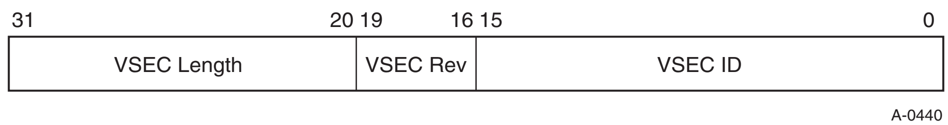

Figure 7-92: Vendor-Specific Header

Table 7-80: Vendor-Specific Header

Bit Location	Register Description	Attributes
15:0	**VSEC ID** – This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.	RO
19:16	**VSEC Rev** – This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.	RO
31:20	**VSEC Length** – This field indicates the number of bytes in the entire VSEC structure, including the PCI Express Extended Capability header, the Vendor-Specific header, and the Vendor-Specific Registers.	RO

7.20. RCRB Header Capability

The PCI Express RCRB Header Capability is an optional Extended Capability that may be implemented in an RCRB to provide a Vendor ID and Device ID for the RCRB and to permit the management of parameters that affect the behavior of Root Complex functionality associated with the RCRB.

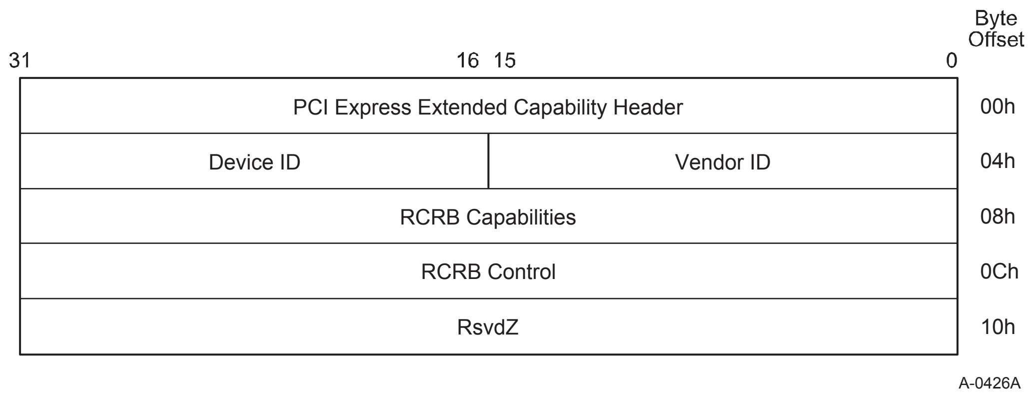

Figure 7-93: Root Complex Features Capability Structure

7.20.1. RCRB Header Extended Capability Header (Offset 00h)

Figure 7-94 details allocation of register fields in the RCRB Header Extended Capability header. Table 7-81 provides the respective bit definitions. Refer to Section 7.9.3 for a description of the PCI Express Enhanced Capabilities header. The Extended Capability ID for the RCRB Header Capability is 000Ah.

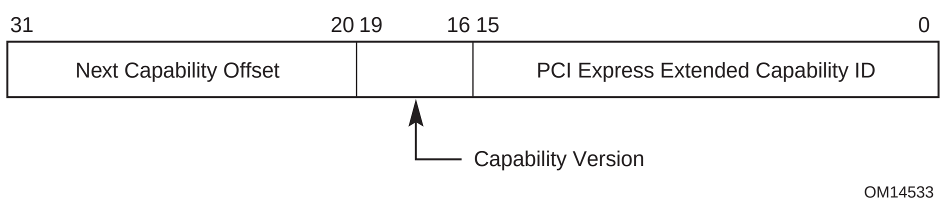

Figure 7-94: RCRB Header Extended Capability Header

Table 7-81: RCRB Header Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. Extended Capability ID for the RCRB Header Capability is 000Ah.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh.	RO

7.20.2. Vendor ID (Offset 04h) and Device ID (Offset 06h)

Figure 7-95 details allocation of register fields in the RCRB Capabilities register; Table 7-82 provides the respective bit definitions.

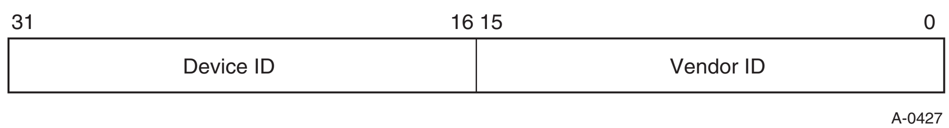

Figure 7-95: Vendor ID and Device ID

Table 7-82: Vendor ID and Device ID

Bit Location	Register Description	Attributes
15:0	**Vendor ID** – PCI-SIG assigned. Analogous to the equivalent field in PCI-compatible Configuration Space. This field provides a means to associate an RCRB with a particular vendor.	RO
31:16	**Device ID** – Vendor assigned. Analogous to the equivalent field in PCI-compatible Configuration Space. This field provides a means for a vendor to classify a particular RCRB.	RO

7.20.3. RCRB Capabilities (Offset 08h)

Figure 7-96 details allocation of register fields in the RCRB Capabilities register; Table 7-83 provides the respective bit definitions.

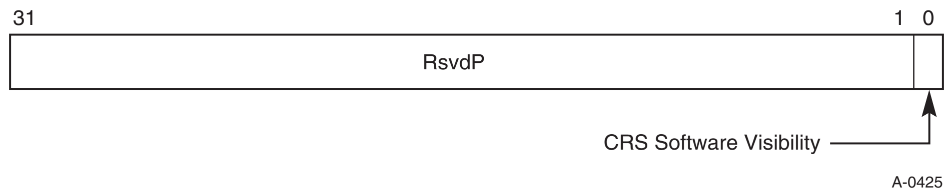

Figure 7-96: RCRB Capabilities

Table 7-83: RCRB Capabilities

Bit Location	Register Description	Attributes
0	**CRS Software Visibility** – When Set, this bit indicates that the Root Complex is capable of returning Configuration Request Retry Status (CRS) Completion Status to software for all Root Ports and integrated devices associated with this RCRB (see Section 2.3.1).	RO

7.20.4. RCRB Control (Offset 0Ch)

Figure 7-97 details allocation of register fields in the RCRB Control register; Table 7-84 provides the respective bit definitions.

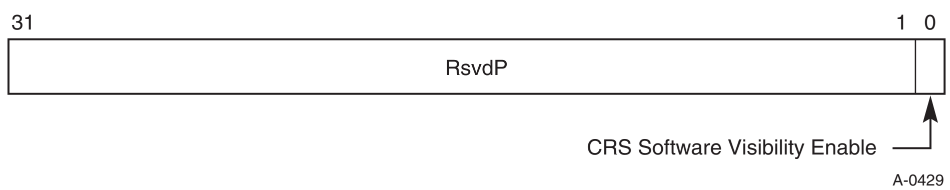

Figure 7-97: RCRB Control

Table 7-84: RCRB Control

Bit Location	Register Description	Attributes
0	**CRS Software Visibility Enable** – When Set, this bit enables the Root Complex to return Configuration Request Retry Status (CRS) Completion Status to software for all Root Ports and integrated devices associated with this RCRB (see Section 2.3.1). RCRBs that do not implement this capability must hardwire this bit to 0b. Default value of this bit is 0b.	RW

7.21. Multicast Capability

Multicast functionality is controlled by the Multicast Capability structure. The Multicast Capability is applicable to Root Ports, RCRBs, Switch Ports, Endpoint Functions, and Root Complex Integrated Endpoints. It is not applicable to PCI Express to PCI/PCI-X Bridges.

In the cases of a Switch or Root Complex or a component that contains multiple Functions, multiple copies of this Capability structure are required – one for each Endpoint Function, Switch Port, or Root Port that supports Multicast. To provide implementation efficiencies, certain fields within each of the Multicast Capability structures within a component must be programmed the same and results are indeterminate if this is not the case. The fields and registers that must be configured with the same values include MC_Enable, MC_Num_Group, MC_Base_Address and MC_Index_Position. These same fields in an Endpoint's Multicast Capability structure must match those configured into a Multicast Capability structure of the Switch or Root Complex above the Endpoint or in which the Root Complex Integrated Endpoint is integrated.

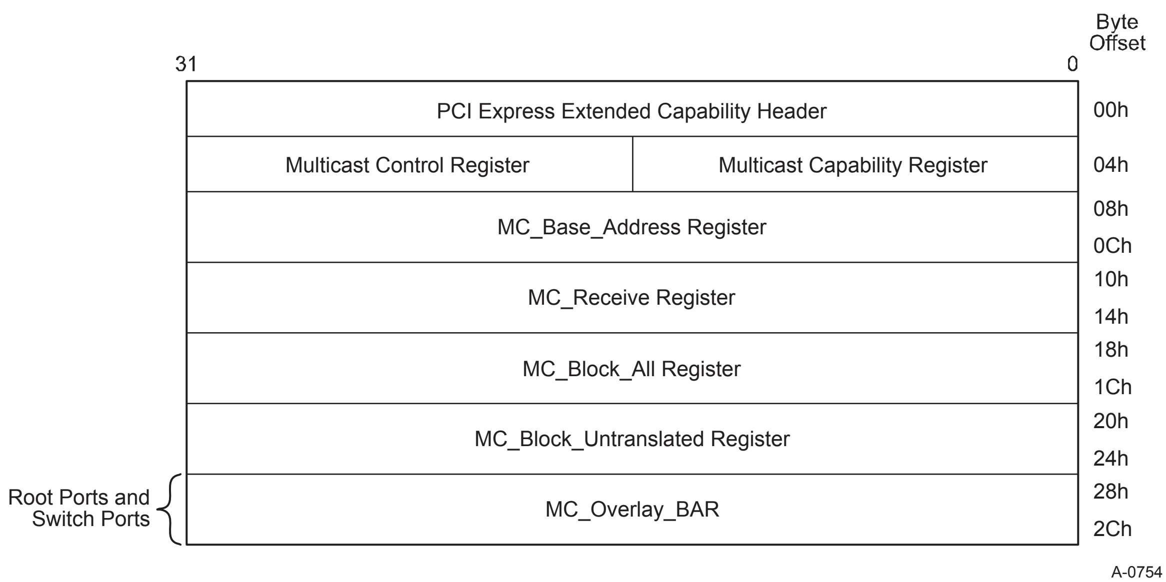

Figure 7-98: Multicast Extended Capability Structure

7.21.1. Multicast Extended Capability Header (Offset 00h)

Figure 7-99 details allocation of the fields in the Multicast Extended Capability Header and Table 7-85 provides the respective bit definitions.

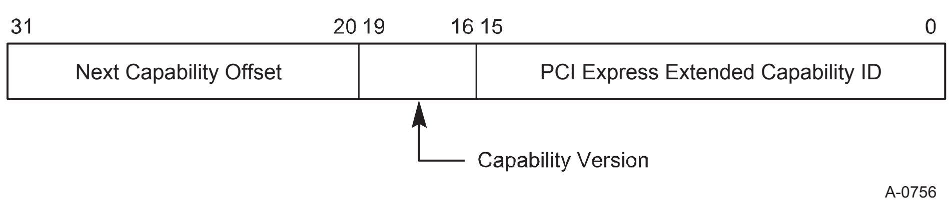

Figure 7-99: Multicast Extended Capability Header

Table 7-85: Multicast Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Multicast Capability is 0012h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities.	RO

7.21.2. Multicast Capability Register (Offset 04h)

Figure 7-100 details allocation of the fields in the Multicast Capability register and Table 7-86 provides the respective bit definitions.

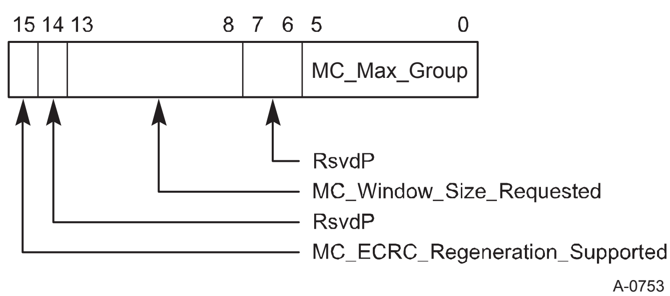

Figure 7-100: Multicast Capability Register

Table 7-86: Multicast Capability Register

Bit Location	Register Description	Attributes
5:0	**MC_Max_Group** – Value indicates the maximum number of Multicast Groups that the component supports, encoded as M-1. A value of 00h indicates that one Multicast Group is supported.	RO
13:8	**MC_Window_Size_Requested** – In Endpoints, the $\log_2$ of the Multicast Window size requested. RsvdP in Switch and Root Ports.	RO

Bit Location	Register Description	Attributes
15	**MC_ECRC_Regeneration_Supported** – If Set, indicates that ECRC regeneration is supported. This bit must not be Set unless the Function supports Advanced Error Reporting, and the ECRC Check Capable bit in the Advanced Error Capabilities and Control register is also Set. However, if ECRC regeneration is supported, its operation is not contingent upon the setting of the ECRC Check Enable bit in the Advanced Error Capabilities and Control register.	RO

7.21.3. Multicast Control Register (Offset 06h)

Figure 7-101 details allocation of the fields in the Multicast Control register and Table 7-87 provides the respective bit definitions.

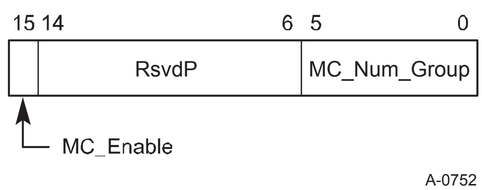

Figure 7-101: Multicast Control Register

Table 7-87: Multicast Control Register

Bit Location	Register Description	Attributes
5:0	**MC_Num_Group** – Value indicates the number of Multicast Groups configured for use, encoded as N-1. The default value of 00h indicates that one Multicast Group is configured for use. Behavior is undefined if value exceeds MC_Max_Group. This parameter indirectly defines the upper limit of the Multicast address range. This field is ignored if MC_Enable is Clear. Default is 0.	RW
15	**MC_Enable** – When Set, the Multicast mechanism is enabled for the component. Default is 0.	RW

7.21.4. Multicast Base Address Register (Offset 08h)

The MC_Base_Address register contains the MC_Base_Address and the MC_Index_Position. Figure 7-102 details allocation of the fields in the MC_Base_Address register and Table 7-88 provides the respective bit definitions.

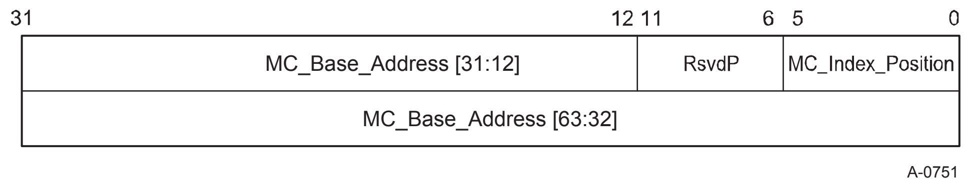

Figure 7-102: MC_Base_Address Register

Table 7-88: Multicast Base Address Register

Bit Location	Register Description	Attributes
5:0	**MC_Index_Position** – The location of the LSB of the Multicast Group number within the address. Behavior is undefined if this value is less than 12 and MC_Enable is Set. Default is 0.	RW
63:12	**MC_Base_Address** – The base address of the Multicast address range. The behavior is undefined if MC_Enable is Set and bits in this field corresponding to address bits that contain the Multicast Group number or address bits less than MC_Index_Position are non-zero. Default is 0.	RW

7.21.5. MC_Receive Register (Offset 10h)

The MC_Receive register provides a bit vector denoting which Multicast groups the Function should accept, or in the case of Switch and Root Complex Ports, forward Multicast TLPs. This register is required in all Functions that implement the MC Capability structure.

Figure 7-103 details allocation of the fields in the MC_Receive register and Table 7-89 provides the respective bit definitions.

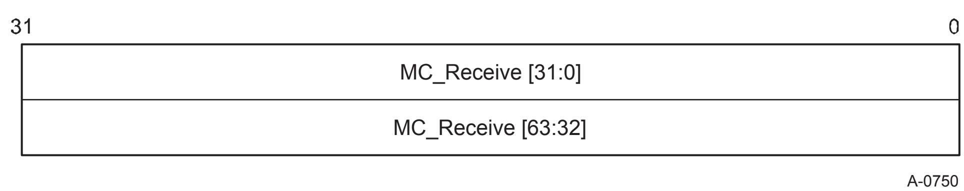

Figure 7-103: MC_Receive Register

Table 7-89: MC_Receive Register

Bit Location	Register Description	Attributes
MC_Max_Group:0	**MC_Receive** – For each bit that's Set, this Function gets a copy of any Multicast TLPs for the associated Multicast Group. Bits above MC_Num_Group are ignored by hardware. Default is 0.	RW
All other bits	Reserved	RsvdP

7.21.6. MC_Block_All Register (Offset 18h)

The MC_Block_All register provides a bit vector denoting which Multicast groups the Function should block. This register is required in all Functions that implement the MC Capability structure.

Figure 7-104 details allocation of the fields in the MC_Block_All Register and Table 7-90 provides the respective bit definitions.

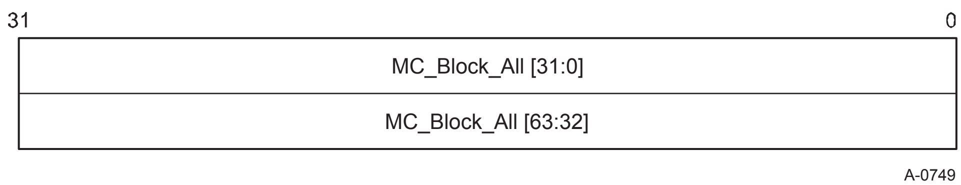

Figure 7-104: MC_Block_All Register

Table 7-90: MC_Block_All Register

Bit Location	Register Description	Attributes
MC_Max_Group:0	**MC_Block_All** – For each bit that is Set, this Function is blocked from sending TLPs to the associated Multicast Group. Bits above MC_Num_Group are ignored by hardware. Default is 0.	RW
All other bits	Reserved	RsvdP

7.21.7. MC_Block_Untranslated Register (Offset 20h)

The MC_Block_Untranslated register is used to determine whether or not a TLP that includes an Untranslated Address should be blocked. This register is required in all Functions that implement the MC Capability structure. However, an Endpoint Function that does not implement the ATS capability may implement this register as RsvdP.

Figure 7-105 details allocation of the fields in the MC_Block_Untranslated register and Table 7-91 provides the respective bit definitions.

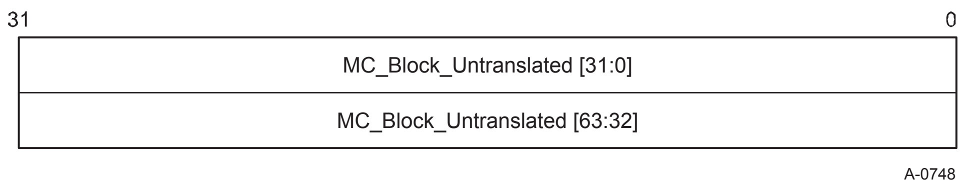

Figure 7-105: MC_Block_Untranslated Register

Table 7-91: MC_Block_Untranslated Register

Bit Location	Register Description	Attributes
MC_Max_Group:0	**MC_Block_Untranslated** – For each bit that is Set, this Function is blocked from sending TLPs containing Untranslated Addresses to the associated MCG. Bits above MC_Num_Group are ignored by hardware. Default is 0.	RW
All other bits	Reserved	RsvdP

7.21.8. MC_Overlay_BAR (Offset 28h)

The MC_Overlay_BAR is required in Switch and Root Complex Ports that support the Multicast Capability and not implemented in Endpoints. Software must interpret the Device/Port Type Field in the PCI Express Capabilities Register to determine if the MC_Overlay_BAR is present in a Function.

The MC_Overlay_BAR specifies the base address of a window in unicast space onto which Multicast TLPs going out an Egress Port are overlaid by a process of address replacement. This allows a single BAR in an Endpoint attached to the Switch or Root Port to be used for both unicast and Multicast traffic. At a Switch Upstream Port, it allows the Multicast address range, or a portion of it, to be overlayed onto host memory.

Figure 7-106 details allocation of the fields in the MC_Overlay BAR and Table 7-92 provides the respective bit definitions.

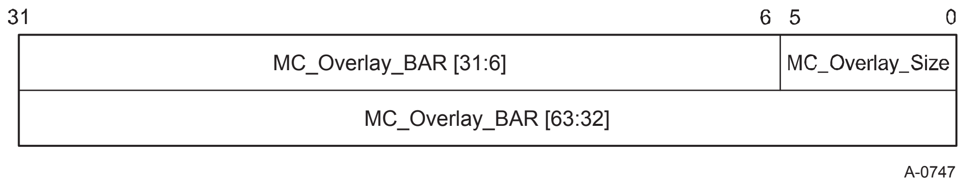

Figure 7-106: MC_Overlay_BAR

Table 7-92: MC Overlay BAR

Bit Location	Register Description	Attributes
5:0	**MC_Overlay_Size** – If 6 or greater, specifies the size in bytes of the overlay aperture as a power of 2. If less than 6, disables the overlay mechanism. Default is 0.	RW
63:6	**MC_Overlay_BAR** – Specifies the base address of the window onto which MC TLPs passing through this Function will be overlaid. Default is 0.	RW

7.22. Resizable BAR Capability

The Resizable BAR Capability is an optional capability that allows hardware to communicate resource sizes, and system software, after determining the optimal size, to communicate this optimal size back to the hardware. Hardware communicates the resource sizes that are acceptable for operation via the Resizable BAR Capability register. Software determines, through a proprietary mechanism, what the optimal size is for the resource, and programs that size via the BAR Size field of the Resizable BAR Control register. Hardware immediately reflects the size inference in the read-only bits of the appropriate Base Address register. Hardware must Clear any bits that change from R/W to read-only, so that subsequent reads return zero. Software must clear the Memory Space Enable bit in the Command register before writing the BAR Size field. After writing the BAR Size field, the contents of the corresponding BAR are undefined. To ensure that it contains a valid address after resizing the BAR, system software must reprogram the BAR, and Set the Memory Space Enable bit (unless the resource is not allocated).

The Resizable BAR Capability register is permitted to indicate the ability to operate at 4 GB or greater only if the associated BAR is a 64-bit BAR.

This capability is applicable to Functions that have Base Address registers only. It is strongly recommended that a Function not advertise any supported BAR sizes in its Resizable BAR Capability register that are larger than the space it would effectively utilize if allocated.

IMPLEMENTATION NOTE

Using the Capability During Resource Allocation

System software that allocates resources can use this capability to resize the resources inferred by the Function's BAR's read-only bits. Previous versions of this software determined the resource size by writing FFFFh to the BAR, reading back the value, and determining the size by the number of bits that are Set. Following this, the base address is written to the BAR.

System software uses this capability in place of the above mentioned method of determining the resource size, and prior to assigning the base address to the BAR. Potential usable resource sizes are reported by the Function, and read, from the Resizable BAR Capability registers. It is intended that the software allocate the largest of the reported sizes that it can, since allocating less address space than the largest reported size can result in lower performance. Software then writes the size to the Resizable BAR Command register for the appropriate BAR for the Function. Following this, the base address is written to the BAR.

For interoperability reasons, it is possible that hardware will set the default size of the BAR to a low size; that is, a size lower than the largest reported in the Resizable BAR Capability register. Software that does not use this capability to size resources will likely result in sub-optimal resource allocation, where the resources are smaller than desirable, or not allocatable because there is no room for them.

With the Resizable BAR capability, the amount of address space consumed by a device can change. In a resource constrained environment, the allocation of more address space to a device may result in allocation of less of the address space to other memory-mapped hardware, like system RAM. System software responsible for allocating resources in this kind of environment is recommended to distribute the limited address space appropriately.

The Resizable BAR Capability structure defines a PCI Express Extended Capability which is located in PCI Express Extended Configuration Space, that is, above the first 256 bytes, and is shown below in Figure 7-107. This structure allows devices with this capability to be identified and controlled. A Capability and a Control register is implemented for each BAR that is resizable. Since a maximum of six BARs may be implemented by any Function, the Resizable BAR Capability structure can range from 12 bytes long (for a single BAR) to 52 bytes long (for all six BARs).

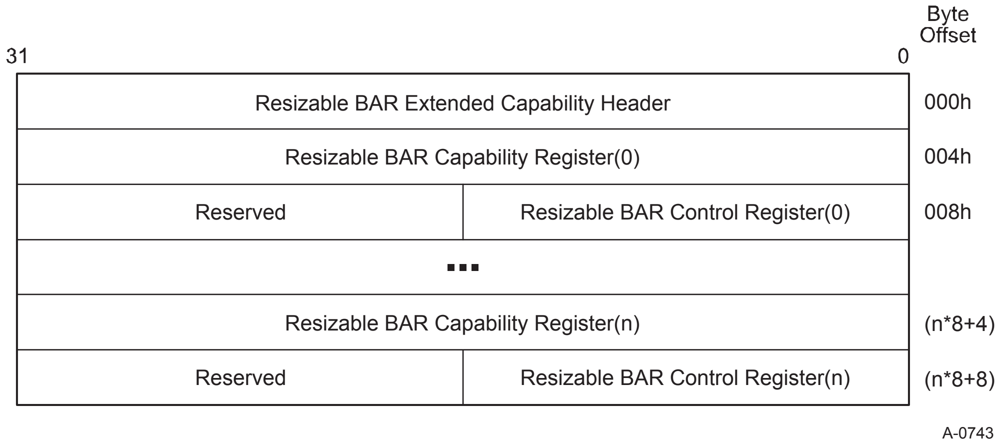

Figure 7-107: Resizable BAR Capability

7.22.1. Resizable BAR Extended Capability Header (Offset 00h)

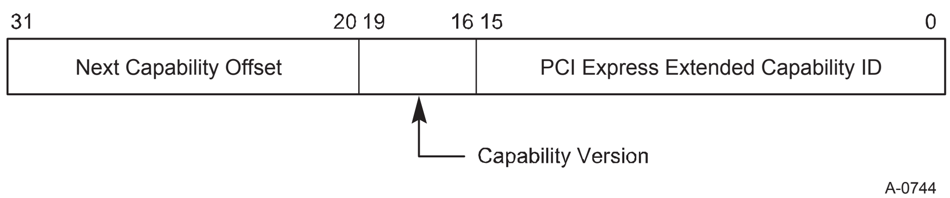

Figure 7-108: Resizable BAR Extended Capability Header

Table 7-93: Resizable BAR Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability. The PCI Express Extended Capability ID for the Resizable BAR Capability is 0015h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities.	RO

7.22.2. Resizable BAR Capability Register (Offset 04h)

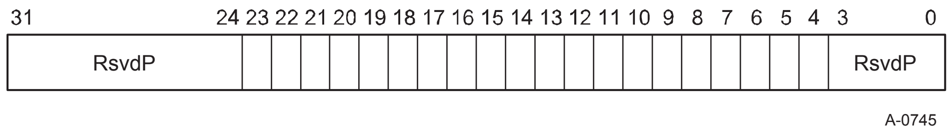

Figure 7-109: Resizable BAR Capability Register

Table 7-94: Resizable BAR Capability Register

Bit Location	Register Description	Attributes
4	When Set, indicates that the Function supports operating with the BAR sized to 1 MB	RO
5	When Set, indicates that the Function supports operating with the BAR sized to 2 MB	RO
6	When Set, indicates that the Function supports operating with the BAR sized to 4 MB	RO
7	When Set, indicates that the Function supports operating with the BAR sized to 8 MB	RO
8	When Set, indicates that the Function supports operating with the BAR sized to 16 MB	RO
9	When Set, indicates that the Function supports operating with the BAR sized to 32 MB	RO
10	When Set, indicates that the Function supports operating with the BAR sized to 64 MB	RO
11	When Set, indicates that the Function supports operating with the BAR sized to 128 MB	RO
12	When Set, indicates that the Function supports operating with the BAR sized to 256 MB	RO
13	When Set, indicates that the Function supports operating with the BAR sized to 512 MB	RO
14	When Set, indicates that the Function supports operating with the BAR sized to 1 GB	RO
15	When Set, indicates that the Function supports operating with the BAR sized to 2 GB	RO
16	When Set, indicates that the Function supports operating with the BAR sized to 4 GB	RO
17	When Set, indicates that the Function supports operating with the BAR sized to 8 GB	RO
18	When Set, indicates that the Function supports operating with the BAR sized to 16 GB	RO
19	When Set, indicates that the Function supports operating with the BAR sized to 32 GB	RO

Bit Location	Register Description	Attributes
20	When Set, indicates that the Function supports operating with the BAR sized to 64 GB	RO
21	When Set, indicates that the Function supports operating with the BAR sized to 128 GB	RO
22	When Set, indicates that the Function supports operating with the BAR sized to 256 GB	RO
23	When Set, indicates that the Function supports operating with the BAR sized to 512 GB	RO

7.22.3. Resizable BAR Control Register (Offset 08h)

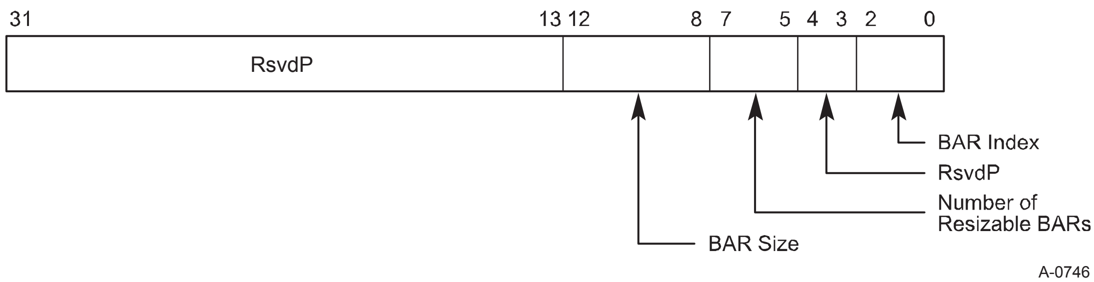

Figure 7-110: Resizable BAR Control Register

Table 7-95: Resizable BAR Control Register

Bit Location	Register Description	Attributes
2:0	**BAR Index** – This encoded value points to the beginning of the BAR. 0 = BAR located at offset 10h 1 = BAR located at offset 14h 2 = BAR located at offset 18h 3 = BAR located at offset 1Ch 4 = BAR located at offset 20h 5 = BAR located at offset 24h All other encodings are reserved. For a 64-bit Base Address register, the BAR Index indicates the lower DWORD. This value indicates which BAR supports a negotiable size.	RO
7:5	**Number of Resizable BARs** – Indicates the total number of resizable BARs in the capability structure for the Function. See Figure 7-107. The value of this field must be in the range of 01h to 06h. The field is valid in Resizable BAR Control register (0) (at offset 008h), and is RsvdP for all others.	RO/RsvdP

Bit Location	Register Description	Attributes
12:8	**BAR Size** – This is an encoded value. 0 = 1 MB 1 = 2 MB 2 = 4 MB 3 = 8 MB … 19 = 512 GB The default value of this field is equal to the default size of the address space that the BAR resource is requesting via the BAR's read-only bits. When this register field is programmed, the value is immediately reflected in the size of the resource, as encoded in the number of read-only bits in the BAR. Software must only write supported values that correspond to those reported in the Resizable BAR Capability register. Writing an unsupported value will produce undefined results.	R/W

7.23. ARI Capability

ARI is an optional capability. This capability must be implemented by each Function in an ARI Device. It is not applicable to a Root Port, a Switch Downstream Port, a Root Complex Integrated Endpoint, or a Root Complex Event Collector.

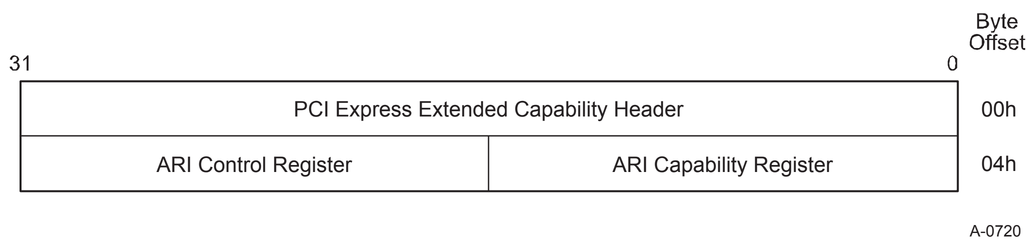

Figure 7-111: ARI Capability

7.23.1. ARI Capability Header (Offset 00h)

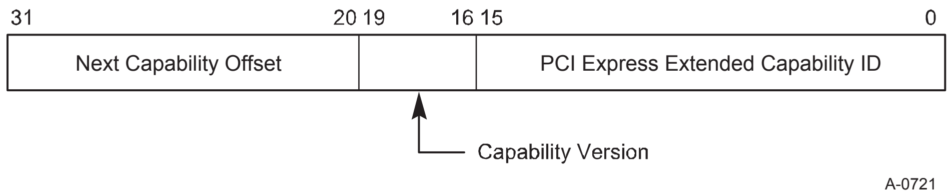

Figure 7-112: ARI Capability Header

Table 7-96: ARI Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability. PCI Express Extended Capability ID for the ARI Capability is 000Eh.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities.	RO

7.23.2. ARI Capability Register (Offset 04h)

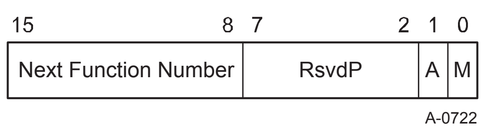

Figure 7-113: ARI Capability Register

Table 7-97: ARI Capability Register

Bit Location	Register Description	Attributes
0	**MFVC Function Groups Capability (M)** – Applicable only for Function 0; must be 0b for all other Functions. If 1b, indicates that the ARI Device supports Function Group level arbitration via its Multi-Function Virtual Channel (MFVC) Capability structure.	RO

Bit Location	Register Description	Attributes
1	**ACS Function Groups Capability (A)** – Applicable only for Function 0; must be 0b for all other Functions. If 1b, indicates that the ARI Device supports Function Group level granularity for ACS P2P Egress Control via its ACS Capability structures.	RO
15:8	**Next Function Number** – This field indicates the Function Number of the next higher numbered Function in the Device, or 00h if there are no higher numbered Functions. Function 0 starts this linked list of Functions.	RO

7.23.3. ARI Control Register (Offset 06h)

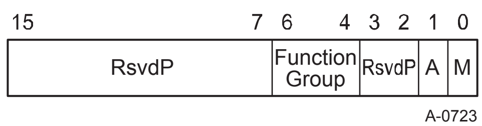

Figure 7-114: ARI Control Register

Table 7-98: ARI Control Register

Bit Location	Register Description	Attributes
0	**MFVC Function Groups Enable (M)** – Applicable only for Function 0; must be hardwired to 0b for all other Functions. When set, the ARI Device must interpret entries in its Function Arbitration Table as Function Group Numbers rather than Function Numbers. Default value of this field is 0b. Must be hardwired to 0b if the MFVC Function Groups Capability bit is 0b.	RW
1	**ACS Function Groups Enable (A)** – Applicable only for Function 0; must be hardwired to 0b for all other Functions. When set, each Function in the ARI Device must associate bits within its Egress Control Vector with Function Group Numbers rather than Function Numbers. Default value of this field is 0b. Must be hardwired to 0b if the ACS Function Groups Capability bit is 0b.	RW
6:4	**Function Group** – Assigns a Function Group Number to this Function. Default value of this field is 000b. Must be hardwired to 000b if in Function 0, the MFVC Function Groups Capability bit and ACS Function Groups Capability bit are both 0b.	RW

7.24. Dynamic Power Allocation (DPA) Capability

The DPA Capability structure is shown in Figure 7-115.

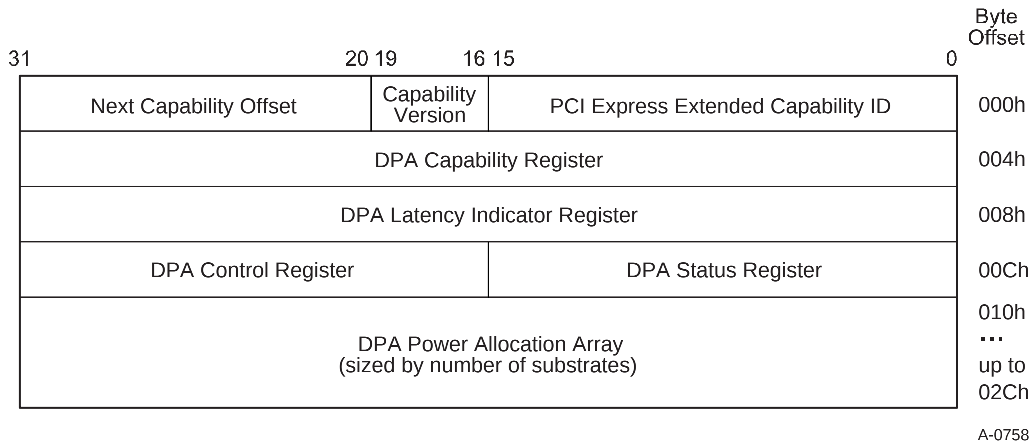

Figure 7-115: Dynamic Power Allocation Capability Structure

7.24.1. DPA Extended Capability Header (Offset 00h)

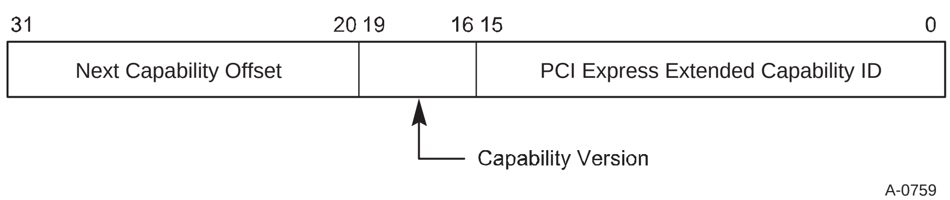

Figure 7-116: DPA Extended Capability Header

Table 7-99: DPA Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the DPA Extended Capability is 0016h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities.	RO

7.24.2. DPA Capability Register (Offset 04h)

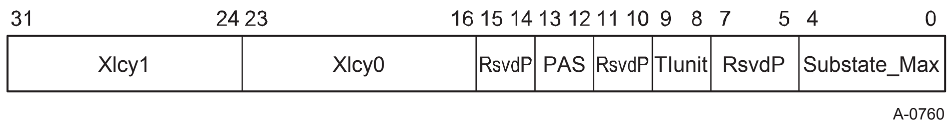

Figure 7-117: DPA Capability Register

Table 7-100: DPA Capability Register

Bit Location	Register Description	Attributes
4:0	**Substate_Max** – Value indicates the maximum substate number, which is the total number of supported substates minus one. A value of 00000b indicates support for one substate.	RO
9:8	**Transition Latency Unit (Tlunit)** – A substate's Transition Latency Value is multiplied by the Transition Latency Unit to determine the maximum Transition Latency for the substate. Defined encodings are 00b – 1 ms 01b – 10 ms 10b – 100 ms 11b – Reserved	RO
13:12	**Power Allocation Scale (PAS)** – The encodings provide the scale to determine power allocation per substate in Watts. The value corresponding to the substate in the Substate Power Allocation Register is multiplied by this field to determine the power allocation for the substate. Defined encodings are 00b – 10.0x 01b – 1.0x 10b – 0.1x 11b – 0.01x	RO
23:16	**Transition Latency Value 0 (Xlcy0)** – This value is multiplied by the Transition Latency Unit to determine the maximum Transition Latency for the substate	RO
31:24	**Transition Latency Value 1 (Xlcy1)** – This value is multiplied by the Transition Latency Unit to determine the maximum Transition Latency for the substate.	RO

7.24.3. DPA Latency Indicator Register (Offset 08h)

Figure 7-118: DPA Latency Indicator Register

Table 7-101: DPA Latency Indicator Register

Bit Location	Register Description	Attributes
Substate_Max:0	**Transition Latency Indicator Bits** – Each bit indicates which Transition Latency Value is associated with the corresponding substate. A value of 0b indicates Transition Latency Value 0; a value of 1b indicates Transition Latency Value 1.	RO
All other bits	Reserved	RsvdP

7.24.4. DPA Status Register (Offset 0Ch)

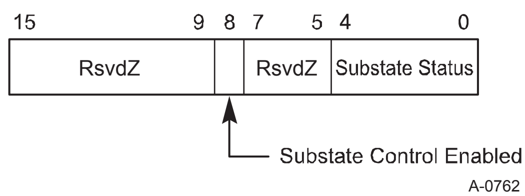

Figure 7-119: DPA Status Register

Table 7-102: DPA Status Register

Bit Location	Register Description	Attributes
4:0	**Substate Status** – Indicates current substate for this Function. Default is 00000b	RO
8	**Substate Control Enabled** – Used by software to disable the Substate Control field in the DPA Control register. Hardware sets this bit following a Conventional Reset or FLR. Software clears this bit by writing a 1b to it. Software is unable to set this bit directly. When this bit is Set, the Substate Control field determines the current substate. When this bit is Clear, the Substate Control field has no effect on the current substate. Default value is 1b.	RW1C

7.24.5. DPA Control Register (Offset 0Eh)

Figure 7-120: DPA Control Register

Table 7-103: DPA Control Register

Bit Location	Register Description	Attributes
4:0	**Substate Control** – Used by software to configure the Function substate. Software writes the substate value in this field to initiate a substate transition. When the Substate Control Enabled bit in the DPA Status register is Set, this field determines the Function substate. When the Substate Control Enabled bit in the DPA Status register is Clear, this field has no effect the Function substate. Default value is 00000b.	RW

7.24.6. DPA Power Allocation Array

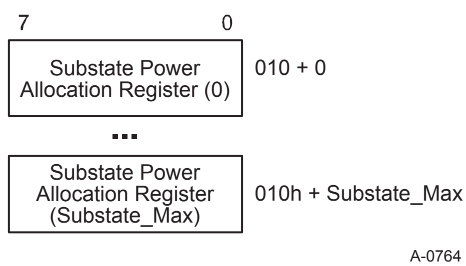

Figure 7-121: DPA Power Allocation Array

Each Substate Power Allocation Register indicates the power allocation value for its associated substate. The number of Substate Power Allocation Registers implemented must be equal to the number of substates supported by Function, which is Substate_Max plus one.

Table 7-104: Substate Power Allocation Register (0 to Substate_Max)

Bit Location	Register Description	Attributes
7:0	**Substate Power Allocation** – The value in this field is multiplied by the Power Allocation Scale to determine power allocation in Watts for the associated substate.	RO

7.25. Latency Tolerance Reporting (LTR) Capability

The PCI Express Latency Tolerance Reporting (LTR) Capability is an optional Extended Capability that allows software to provide platform latency information to components with Upstream Ports (Endpoints and Switches), and is required if the component supports the LTR mechanism. It is not applicable to Root Ports, Bridges, or Downstream Ports in a Switch.

For a multi-Function device associated with the Upstream Port of a component that implements the LTR mechanism, this Capability structure must be implemented only in Function 0, and must control the component's Link behavior on behalf of all the Functions of the device.

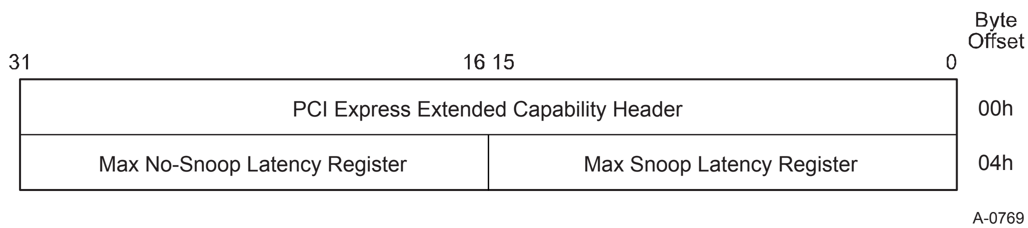

Figure 7-122: LTR Extended Capability Structure

7.25.1. LTR Extended Capability Header (Offset 00h)

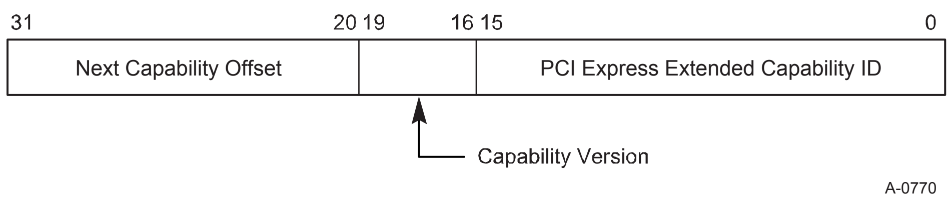

Figure 7-123: LTR Extended Capability Header

Table 7-105: LTR Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability for the LTR Extended Capability is 0018h.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities.	RO

7.25.2. Max Snoop Latency Register (Offset 04h)

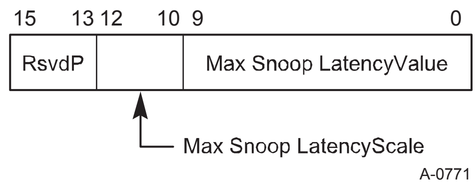

Figure 7-124: Max Snoop Latency Register

Table 7-106: Max Snoop Latency Register

Bit Location	Register Description	Attributes
9:0	**Max Snoop LatencyValue** – Along with the Max Snoop LatencyScale field, this register specifies the maximum snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. The default value for this field is 0.	RW
12:10	**Max Snoop LatencyScale** – This register provides a scale for the value contained within the Maximum Snoop LatencyValue field. Encoding is the same as the LatencyScale fields in the LTR Message. See Section 6.18. The default value for this field is 0. Hardware operation is undefined if software writes a Not Permitted value to this field.	RW

7.25.3. Max No-Snoop Latency Register (Offset 06h)

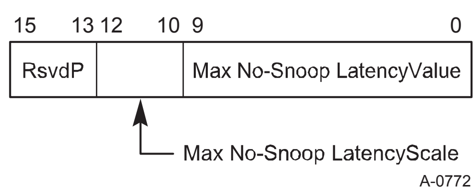

Figure 7-125: Max No-Snoop Latency Register

Table 7-107: Max No-Snoop Latency Register

Bit Location	Register Description	Attributes
9:0	**Max No-Snoop LatencyValue** – Along with the Max No-Snoop LatencyScale field, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. The default value for this field is 0.	RW
12:10	**Max No-Snoop LatencyScale** – This register provides a scale for the value contained within the Max No-Snoop LatencyValue field. Encoding is the same as the LatencyScale fields in the LTR Message. See Section 6.18 The default value for this field is 0. Hardware operation is undefined if software writes a Not Permitted value to this field.	RW

7.26. TPH Requester Capability

The TPH Requester Capability structure is required for all Functions that are capable of generating Request TLPs with TPH. For a multi-Function device, this capability must be present in each Function that is capable of generating Requests with TPH.

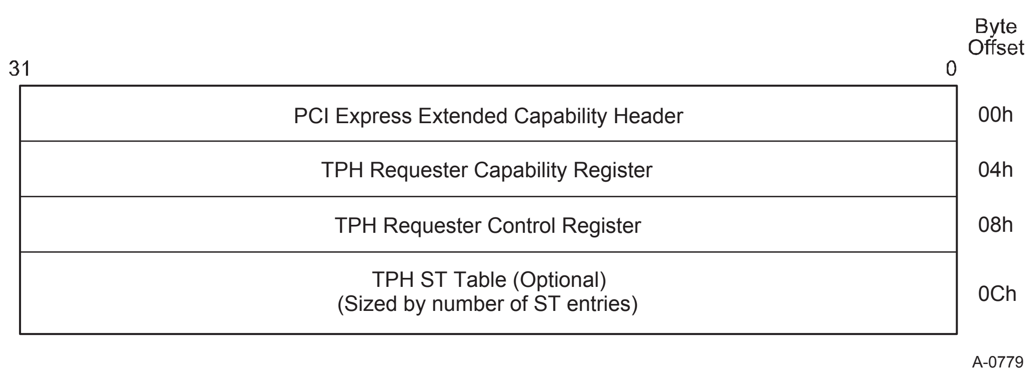

Figure 7-126: TPH Extended Capability Structure

7.26.1. TPH Requester Extended Capability Header (Offset 00h)

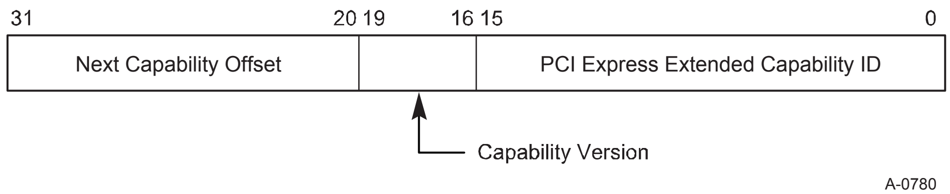

Figure 7-127: TPH Requester Extended Capability Header

Table 7-108: TPH Requester Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the TPH Requester Capability is 17h**.**	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of capabilities.	RO

7.26.2. TPH Requester Capability Register (Offset 04h)

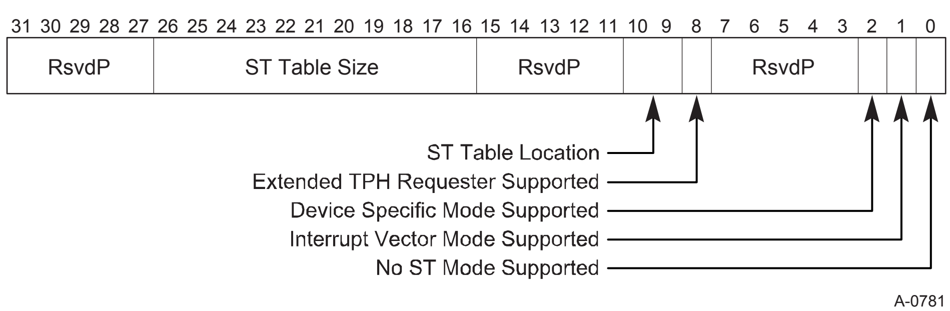

Figure 7-128: TPH Requester Capability Register

Table 7-109: TPH Requester Capability Register

Bit Location	Register Description	Attributes
0	**No ST Mode Supported** – If set indicates that the Function supports the No ST Mode of operation. This mode is required to be supported by all Functions that implement this Capability structure. This field must have a value of 1b.	RO
1	**Interrupt Vector Mode Supported** – If set indicates that the Function supports the Interrupt Vector Mode of operation.	RO
2	**Device Specific Mode Supported** – If set indicates that the Function supports the Device Specific Mode of operation.	RO
8	**Extended TPH Requester Supported** – If Set indicates that the Function is capable of generating Requests with a TPH TLP Prefix. See Section 2.2.7.1 for additional details.	RO
10:9	**ST Table Location** – Value indicates if and where the ST Table is located. Defined Encodings are: 00 – ST Table is not present. 01 – ST Table is located in the TPH Requester Capability structure. 10 – ST Table is located in the MSI-X Table structure. 11 – Reserved A Function that only supports the No ST Mode of operation must have a value of 00b in this field.	RO
26:16	**ST Table Size** – Software reads this field to determine the ST Table Size N, which is encoded as N-1. For example, a returned value of "00000000011" indicates a table size of 4. There is an upper limit of 64 entries when the ST Table is located in the TPH Requester Capability structure. There is an upper limit of 2 K entries when the ST Table is located in the MSI-X Table. This field is only applicable for Functions that implement an ST Table as indicated by the ST Table Location field. Otherwise, the value in this field is undefined.	RO

7.26.3. TPH Requester Control Register (Offset 08h)

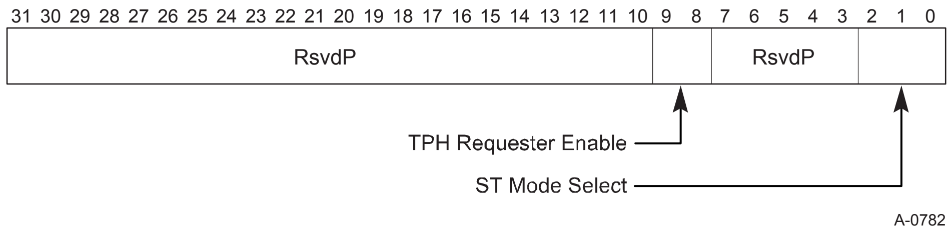

Figure 7-129: TPH Requester Control Register

Table 7-110: TPH Requester Control Register

Bit Location	Register Description	Attributes
2:0	**ST Mode Select** – selects the ST mode of operation. Defined encodings are: 000b – No ST Mode 001b – Interrupt Vector Mode 010b – Device Specific Mode 0thers – reserved for future use Functions that support only the No ST Mode of operation must hardwire this field to 000b. The default value of this field is 000b. See Section 6.17.2for details on ST modes of operation.	RW
9:8	**TPH Requester Enable** – defined encodings are: 00b – Function operating as a Requester is not permitted to issue Requests with TPH or Extended TPH. 01b – Function operating as a Requester is permitted to issue Requests with TPH and is not permitted to issue Requests with Extended TPH. 10b – Reserved. 11b – Function operating as a Requester is permitted to issue Requests with TPH and Extended TPH. The default value of this field is 00b.	RW

7.26.4. TPH ST Table (Starting from Offset 0Ch)

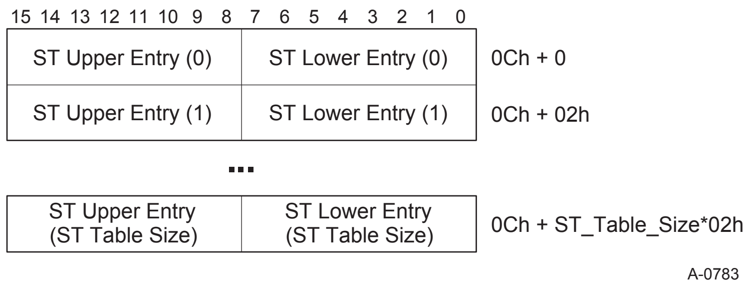

Figure 7-130: TPH ST Table

The ST Table must be implemented in the TPH Requester Capability Structure if the value of the ST Table Location field is 01b. For all other values, the ST Entry Registers must not be implemented. The number of ST Entry Registers implemented must be equal to the number of ST entries supported by the Function, which is the value of the ST Table Size field plus one.

Table 7-111: TPH ST Table

Bit Location	Register Description	Attributes
7:0	**ST Lower** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 01b, then this field contains the lower 8 bits of a Steering Tag. Default value of this field is 0h.	RW
15:8	**ST Upper** – If the Function implements a TPH Requester Capability structure, and the ST Table Location indicates a value of 01b, and the Extended TPH Requester Supported bit is Set, then this field contains the upper 8 bits of a Steering Tag. Otherwise, this field is RsvdP. Default value of this field is 0h.	RW

A

A. Isochronous Applications

A.1. Introduction

The design goal of isochronous mechanisms in PCI Express is to ensure that isochronous traffic receives its allocated bandwidth over a relevant time period while also preventing starvation of other non-isochronous traffic.

Furthermore, there may exist data traffic that requires a level of service falling in between what is required for bulk data traffic and isochronous data traffic. This type of traffic can be supported through the use of Port arbitration within Switches, the use of TC labels [1:7], and optional additional VC resources. Policies for assignment of TC labels and VC resources that are not isochronous-focused are outside the scope of the PCI Express specification.

Two paradigms of PCI Express communication are supported by the PCI Express isochronous mechanisms: Endpoint-to-Root-Complex communication model and peer-to-peer (Endpoint-to-Endpoint) communication model. In the Endpoint-to-Root-Complex communication model, the primary isochronous traffic is memory read and write requests to the Root Complex and read completions from the Root Complex. Figure A-1 shows an example of a simple system with both communication models. In the figure, devices A, B, called Requesters, are PCI Express Endpoints capable of issuing isochronous request transactions, while device C and Root Complex, called Completers, are capable of being the targets of isochronous request transactions. An Endpoint-to-Root-Complex communication is established between device A and the Root Complex, and a peer-to-peer communication is established between device B and device C. In the rest of this section, Requester and Completer will be used to make reference to PCI Express elements involved in transactions. The specific aspects of each communication model will be called out explicitly.