



PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	MFVC
DATE:	December 19, 2003
AFFECTED DOCUMENT:	PCI Express Base Specification, Revision 1.0a
SPONSOR:	Joe Cowan; Hewlett-Packard Company

Part I

1. Summary of the Functional Changes

Define a new Multi-function VC (MFVC) Capability structure to permit enhanced QoS management in a multi-function device (MFD), including TC/VC mapping, optional VC arbitration, and optional function arbitration for upstream requests. In addition, permit each function to contain an optional VC capability structure for QoS management within that function.

2. Benefits as a Result of the Changes

1. Architected QoS management between the functions within an MFD for upstream requests, with less complexity and logic than required for an integrated switch approach.
2. Ability to implement multiple elements (including switches) as different functions within an MFD, without losing the ability for each to contain its own VC Capability structure.
3. Increased ease of integrating existing single-function device cores with VC Capability structures into an MFD.

3. Assessment of the Impact

No impact to systems or peripherals that conform to the PCI Express 1.0a specification.

4. Analysis of the Hardware Implications

New implementations are permitted to expose the new functionality, all of which is optional. Complete specification changes are detailed in subsequent pages.

5. Analysis of the Software Implications

Existing software does not recognize the new capabilities, and operates as before. New software can optionally recognize the new capabilities, and take advantage of the new functionality. The SW management paradigm for the new MFVC Capability and potentially multiple VC Capability structures in an MFD is highly leveraged from the existing SW management paradigms for: (1) MFDs with a single VC Capability structure, and (2) switches.

Part II

Detailed Description of the change

Section 2.2.6.6, p. 58, change text as shown:

Together with the PCI Express Virtual Channel support, the TC mechanism is a fundamental element for enabling differentiated traffic servicing. Every PCI Express Transaction Layer Packet uses TC information as an invariant label that is carried end to end within the PCI Express fabric. As the packet traverses across the fabric, this information is used at every Link and within each Switch element to make decisions with regards to proper servicing of the traffic. A key aspect of servicing is the routing of the packets based on their TC labels through corresponding Virtual Channels. Section [2.5.2.4.2](#) covers the details of the VC mechanism.

Section 2.5, p. 95, change text as shown:

2.5. Virtual Channel (VC) Mechanism

The PCI Express Virtual Channel (VC) mechanism provides support for carrying throughout the PCI Express fabric traffic that is differentiated using TC labels. The foundation of VCs are independent fabric resources (queues/buffers and associated control logic). These resources are used to move information across PCI Express Links with fully independent flow-control between different VCs. This is key to solving the problem of flow-control induced blocking where a single traffic flow may create a bottleneck for all traffic within the system.

Traffic is associated with VCs by mapping packets with particular TC labels to their corresponding VCs. The PCI Express VC [and MFVC mechanisms](#) allows flexible mapping of TCs onto the VCs. In the simplest form, TCs can be mapped to VCs on a 1:1 basis. To allow performance/cost tradeoffs, PCI Express provides the capability of mapping multiple TCs onto a single VC. Section 2.5.2 covers details of TC to VC mapping.

A Virtual Channel is established when one or multiple TCs are associated with physical VC resource designated by VC ID. This process is controlled by the PCI Express configuration software as described in Sections 6.3, [and 7.11](#), [and 7.15](#).

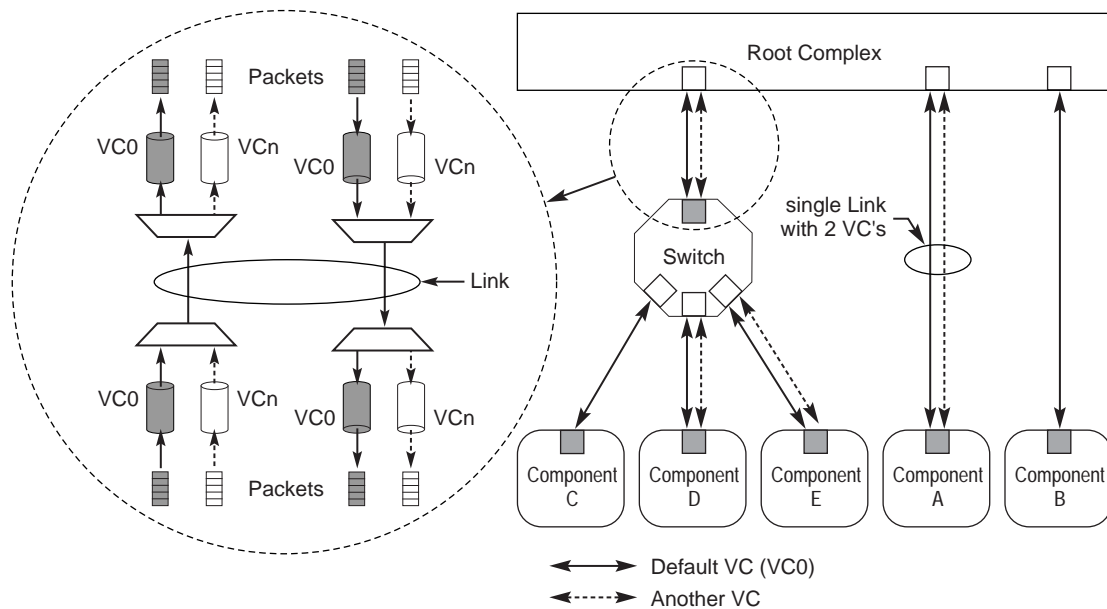
Support for TCs and VCs beyond default TC0/VC0 pair is optional. The association of TC0 with VC0 is fixed, i.e., “hardwired,” and must be supported by all PCI Express components. Therefore the baseline TC/VC setup does not require any VC-specific hardware or software configuration. In order to ensure interoperability, PCI Express components that do not implement the optional PCI Express Virtual Channel Capability structure must obey the following rules:

- ❑ A Requester must only generate requests with TC0 label. (Note that if the Requester initiates requests with a TC label other than TC0, the requests may be treated as malformed by the component on the other side of the Link that implements the extended VC capability and applies TC filtering.)

- ❑ A Completer must accept requests with TC label other than TC0, and must preserve the TC label, i.e., any completion that it generates must have the same TC label as the label of the request.
- ❑ A Switch must map all TCs to VC0 and must forward all transactions regardless of the TC label.

A PCI Express Endpoint or Root Complex that intends to be a Requester that can issue requests with TC label other than TC0 must implement the PCI Express Virtual Channel Capability structure, even if it only supports the default VC. This is required in order to enable mapping of TCs beyond the default configuration. It must follow the TC/VC mapping rules according to the software programming of the VC and MFVC Capability structures.

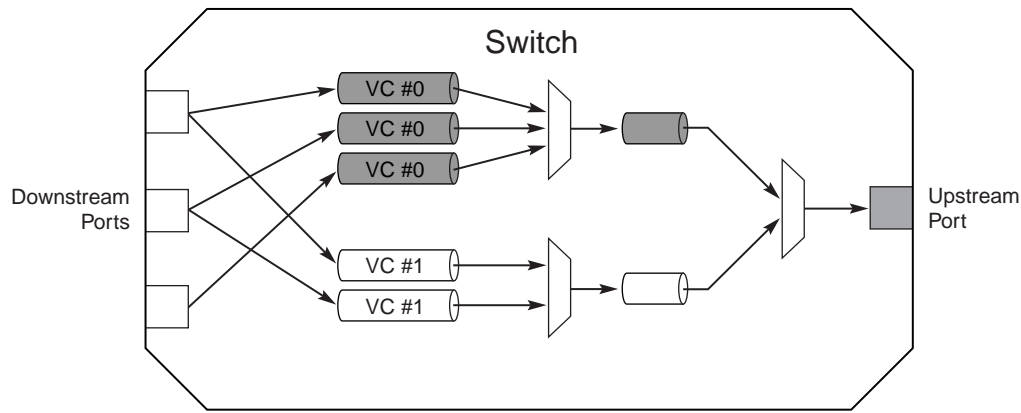
Figure 2-24 illustrates the concept of Virtual Channel. Conceptually, traffic that flows through VCs is multiplexed onto a common physical Link resource on the Transmit side and de-multiplexed into separate VC paths on the Receive side.



OM13760

Figure 2-24: Virtual Channel Concept – An Illustration

Internal to the Switch, every Virtual Channel requires dedicated physical resources (queues/buffers and control logic) that support independent traffic flows inside the Switch. Figure 2-25 shows conceptually the VC resources within the Switch (shown in Figure 2-24) that are required to support traffic flow in the upstream direction.



OM13761

Figure 2-25: Virtual Channel Concept – Switch Internals (Upstream Flow)

[A multi-function device may implement Virtual Channel resources similar to a subset of those in a switch, for the purpose of managing the QoS for upstream requests from the different functions to the device's Upstream Egress Port.](#)



IMPLEMENTATION NOTE

VC and VC Buffering Considerations

1. The amount of buffering beyond the architectural minimums per supported VC is implementation-specific.
2. Buffering beyond the architectural minimums is not required to be identical across all VCs on a given Link, i.e., an implementation may provide greater buffer depth for selected VCs as a function of implementation usage models and other Link attributes, e.g., Link width and signaling.
3. Implementations may adjust their buffering per VC based on implementation-specific policies derived from PCI Express configuration and VC enablement, e.g., if a four VC implementation has only two VCs enabled, the implementation may assign the non-enabled VC buffering to the enabled VCs to improve fabric efficiency/performance by reducing the probability of fabric backpressure due to Link-level flow control.
4. The number of VCs supported, and the associated buffering per VC per Port, are not required to be the same for all Ports of a multi-Port component (a Switch or Root Complex).

2.5.1. Virtual Channel Identification (VC ID)

Individual PCI Express Ports on a Root Complex, Switch, or Device can support 1-8 Virtual Channels – each Port is independently configured/managed therefore allowing implementations to vary the number of VCs supported per Port based on usage model-specific requirements. These VCs are uniquely identified using the Virtual Channel Identification (VC ID) mechanism.

Note that [while DLLPs contain VC ID information for Flow Control accounting](#), TLPs do not [include VC ID information](#). The association of TLPs with a VC ID for the purpose of Flow Control accounting is done at each Port of the Link using TC to VC mapping as discussed in Section 2.5.2.

All PCI Express Ports that support more than VC0 must provide [the at least one](#) VC Capability structure according to the definition in Section 7.11. [A multi-function device is permitted to implement the MFVC Capability structure, as defined in Section 7.15](#). Providing [this these](#) extended structures is optional for Ports that support only the default TC0/VC0 configuration. PCI Express configuration software is responsible for configuring Ports on both sides of the Link for a matching number of VCs. This is accomplished by scanning the PCI Express hierarchy and using VC [or MFVC](#) Capability registers associated with Ports (that support more than default VC0) to establish [the](#) number of VCs for the Link. Rules for assigning VC ID to VC hardware resources within a PCI Express Port are as follows:

- VC ID assignment must be unique per PCI Express Port – The same VC ID cannot be assigned to different VC hardware resources within the same Port.
- VC ID assignment must be the same (matching in the terms of numbers of VCs and their IDs) for the two PCI Express Ports on both sides of a PCI Express Link.
- [If a multi-function device implements an MFVC Capability structure, its VC hardware resources are distinct from the VC hardware resources associated with any VC Capability structures of its Functions. The VC ID uniqueness requirement \(first bullet above\) still applies individually for the MFVC and any VC Capability structures. In addition, the VC ID cross-link matching requirement \(second bullet above\) applies for the MFVC Capability structure, but not the VC Capability structures of the Functions.](#)
- VC ID 0 is assigned and fixed to the default VC.

2.5.2. TC to VC Mapping

Every Traffic Class that is supported must be mapped to one of the Virtual Channels. The mapping of TC0 to VC0 is fixed.

The mapping of TCs other than TC0 is system software specific. However, the mapping algorithm must obey the following rules:

- One or multiple TCs can be mapped to a VC.
- One TC must not be mapped to multiple VCs in any PCI Express Port [or Endpoint Function](#).
- TC/VC mapping must be identical for PCI Express Ports on both sides of a PCI Express Link.

Section 2.5.3, p. 99, change text as shown:

2.5.3. VC and TC Rules

Here is a summary of key rules associated with the TC/VC mechanism:

- ❑ All PCI Express devices must support the general purpose I/O Traffic Class, i.e., TC0 and must implement the default VC0.
- ❑ Each Virtual Channel (VC) has independent Flow Control.
- ❑ There are no ordering relationships required between different TCs
- ❑ There are no ordering relationships required between different VCs
- ❑ A Switch's peer-to-peer capability applies to all Virtual Channels supported by the Switch.
- ❑ [A multi-function device's peer-to-peer capability between different functions applies to all Virtual Channels supported by the multi-function device.](#)
- ❑ Transactions with a TC that is not mapped to any enabled VC in a PCI Express Ingress Port are treated as malformed transactions by the receiving device.
- ❑ For Switches, transactions with a TC that is not mapped to any of enabled VCs in the target Egress Port are treated as malformed TLPs.
- ❑ [For multi-function devices with an MFVC Capability structure, any transaction with a TC that is not mapped to an enabled VC in the MFVC Capability structure is treated as a malformed TLP.](#)
- ❑ For a Root Port, transactions with a TC that is not mapped to any of enabled VCs in the target RCRB are treated as malformed TLPs.
- ❑ Switches must support independent TC/VC mapping configuration for each Port.
- ❑ Root Complex must support independent TC/VC mapping configuration for each RCRB and the associated Root Ports.

For more details on the VC and TC mechanisms, including configuration, mapping, and arbitration, refer to [Chapter Section 6.3](#).

Section 2.6, p. 100, change text as shown:

Each Virtual Channel maintains an independent Flow Control credit pool. The FC information is conveyed between two sides of the Link using DLLP ~~packets~~. The VC ID field of the DLLP is used to carry the Virtual Channel Identification that is required for proper flow-control credit accounting.

[Flow Control mechanisms used internally within a multi-function device are outside the scope of this specification.](#)

Flow Control is handled by the Transaction Layer in cooperation with the Data Link Layer. The Transaction Layer performs Flow Control accounting functions for Received TLPs and "gates" TLP Transmissions based on available credits for transmission.

Section 2.6.1, p. 102, change text as shown:

- ❑ When other Virtual Channels are enabled by software, each newly enabled VC will follow the Flow Control initialization protocol (see Section 3.3)
 - Software enables a Virtual Channel by setting the VC Enable bits for that Virtual Channel in both components on a Link (see Sections [7.11](#) and [7.15](#))

Section 3.3.1, p. 125, change text as shown:

- ❑ Rules for state FC_INIT1:
 - Entered when initialization of a VC (VCx) is required
 - ◆ Entrance to DL_Init state
 - ◆ When a VC is enabled by software (see Sections [7.11](#) and [7.15](#))

Section 6.3, p. 273, change text as shown:

6.3. Virtual Channel Support

6.3.1. Introduction and Scope

Virtual Channel mechanism provides a foundation for supporting differentiated services within the PCI Express fabric. It enables deployment of independent physical resources that together with traffic labeling are required for optimized handling of differentiated traffic. Traffic labeling is supported using Transaction Class TLP-level labels. Exact policy for traffic differentiation is determined by the TC/VC mapping and by the VC-based, [Port-based, and function-based](#) arbitration [mechanisms](#). The TC/VC mapping depends on the platform application requirements. These requirements drive the choice of ~~VC-the~~ arbitration [algorithms](#) and configurability/programmability of arbiters allows detailed tuning of the traffic servicing policy.

Basic definition of [the](#) Virtual Channel mechanism and associated Traffic Class labeling mechanism is covered in Chapter 2. The VC configuration/programming model is defined in Sections [7.11](#) and [7.15](#).

The remaining sections of this chapter cover VC mechanisms from the system perspective. They address the next level details on:

- ❑ Supported TC/VC configurations
- ❑ VC-based arbitration – algorithms and rules
- ❑ Traffic ordering considerations
- ❑ Isochronous support as a specific usage model

6.3.2. TC/VC Mapping and Example Usage

A Virtual Channel is established when one or more TC labels are associated with a physical resource designated by a VC ID. Every Traffic Class that is supported on a given path within the fabric must be mapped to one of the enabled Virtual Channels. Every Port must support the default TC0/VC0 pair – this is “hardwired.” Any additional TC label mapping or additional VC resource enablement is optional and is controlled by system software using the programming model described in Sections [7.11](#) and [7.15](#).

The number of VC resources provisioned within a component or enabled within a given fabric may vary due to implementation and usage model requirements, due to Hot-Plug of disparate components with varying resource capabilities, or due to system software restricting what resources may be enabled on a given path within the fabric.

Some examples to illustrate:

- ❑ A set of components (Root Complex, Endpoints, Switches) may only support the mandatory VC0 resource that must have TC0 mapped to VC0. System software may, based on application usage requirements, map one or all non-zero TC labels to VC0 as well on any or all paths within the fabric.
- ❑ A set of components may support two VC resources, e.g., VC0 and VC1. System software must map TC0/VC0 and in addition, may map one or all non-zero TC labels to either VC0 or VC1. As above, these mappings may be enabled on any or all paths within the fabric. See the examples below for additional information.
- ❑ A Switch may be implemented with eight Ports – seven x1 Links with two VC resources and one x16 Link with one VC resource. System software may enable both VC resources on the x1 Links and assign one or more additional TC labels to either VC thus allowing the Switch to differentiate traffic flowing between any Ports. The x16 Link must be also configured to map any non-TC0 traffic to VC0 if such traffic is to flow on this Link. Note: multi-Port components (Switches and Root Complex) are required to support independent TC/VC mapping per PCI Express Port.

In any of the above examples, system software has the ability to map one, all, or a subset of the TC labels to a given VC. Should system software wish to restrict the number of traffic classes that may flow through a given Link, it may configure only a subset of the TC labels to the enabled VC resources. Any TLP that does not contain a TC label that has been mapped to an enabled VC resource shall be treated as a malformed TLP and dropped by the receiving Port. This is referred to as TC Filtering; [however, Flow Control credits will be lost, and an uncorrectable error will be generated, so software intervention will usually be required to restore proper operation after a TC Filtering event occurs.](#)

Section 6.3.3.1, p. 280, delete the following Implementation Note:



IMPLEMENTATION NOTE

Arbitration for Multi-Function Endpoints

~~The arbitration of data flows from different functions of a multi-function Endpoint is beyond the scope of this specification. Mapping of different data flows (within a multi-function Endpoint) to different TCs and VCs is implementation specific. Multi-function Endpoints, however, should support PCI Express VC-based arbitration control mechanisms if multiple VCs are implemented for the PCI Express Link.~~

~~When a common VC on the PCI Express Link is shared by multiple functions, the aggregated traffic over the VC is subject to the bandwidth and latency regulations for that VC on the PCI Express Link. The multi-function Endpoints should implement proper arbitration for data flows from different functions in order to share the Link resources and achieve desired end-to-end services.~~

Section 6.3.3.2, p. 281, change text as shown:

If strict priority arbitration is supported by the hardware for a subset of the VC resources, software can configure the VCs into two priority groups – a lower and an upper group. The upper group is treated as a strict priority arbitration group while the lower group that is arbitrated to only when there are no packets to process in the upper group. Figure 6-8 illustrates an example configuration that supports eight VCs separated into two groups – the lower group consisting of VC0-VC3 and the upper group consisting of VC4-VC7. The arbitration within the lower group can be configured to one of the supported arbitration methods. The Low Priority Extended VC Count field in the Port VC Capability register 1 indicates the size of this group. The arbitration methods are listed in the VC Arbitration Capability field in the Port VC Capability register 2. See Sections [7.11](#) and [7.15](#) for details. When the Low Priority Extended VC Count field is set to zero, all VCs are governed by the strict-priority VC arbitration; when the field is equal to the Extended VC Count, all VCs are governed by the VC arbitration indicated by the VC Arbitration Capability field.

Section 6.3.3.2.1, p. 281, change text as shown:

Strict priority arbitration enables minimal latency for high-priority transactions. However, there is potential danger of bandwidth starvation should it not be applied correctly. Using strict priority requires all high-priority traffic to be regulated in terms of maximum peak bandwidth and Link usage duration. Regulation must be applied either at the transaction injection Port/[Function](#) or within subsequent Egress Ports where data flows contend for a common Link. System software must configure traffic such that lower priority transactions will be serviced at a sufficient rate to avoid transactions timeouts.

Section 6.3.3.2.2, p. 282, change text as shown:

Although weights can be fixed (by hardware implementation) for certain applications, to provide more generic support for different applications, PCI Express components that support the WRR scheme are recommended to implement programmable WRR.

Programming of WRR is controlled using the software interface defined in Sections [7.11](#) [and 7.15](#).

Add the following new section prior to Section 6.3.4 on page 283

6.3.3.4. Multi-function Devices and Function Arbitration

The multi-function arbitration model defines an optional arbitration infrastructure and functionality within a multi-function device. This functionality is needed to support a set of arbitration policies that control traffic contention for the device's Upstream Egress Port from its multiple functions.

Figure 6-9 shows a conceptual model of a multi-function device highlighting resources and associated functionality. Note that each function optionally contains a VC Capability structure, which if present manages TC/VC mapping, optional Port Arbitration, and optional VC arbitration, all within the function. The MFVC Capability structure manages TC/VC mapping, optional Function Arbitration, and optional VC Arbitration for the device's Upstream Egress Port. Together these resources enable enhanced QoS management for *upstream* requests. However, in contrast to a complete switch with devices on its downstream Ports, the multi-function device model does not support full QoS management for *peer-to-peer* requests between functions or for *downstream* requests.

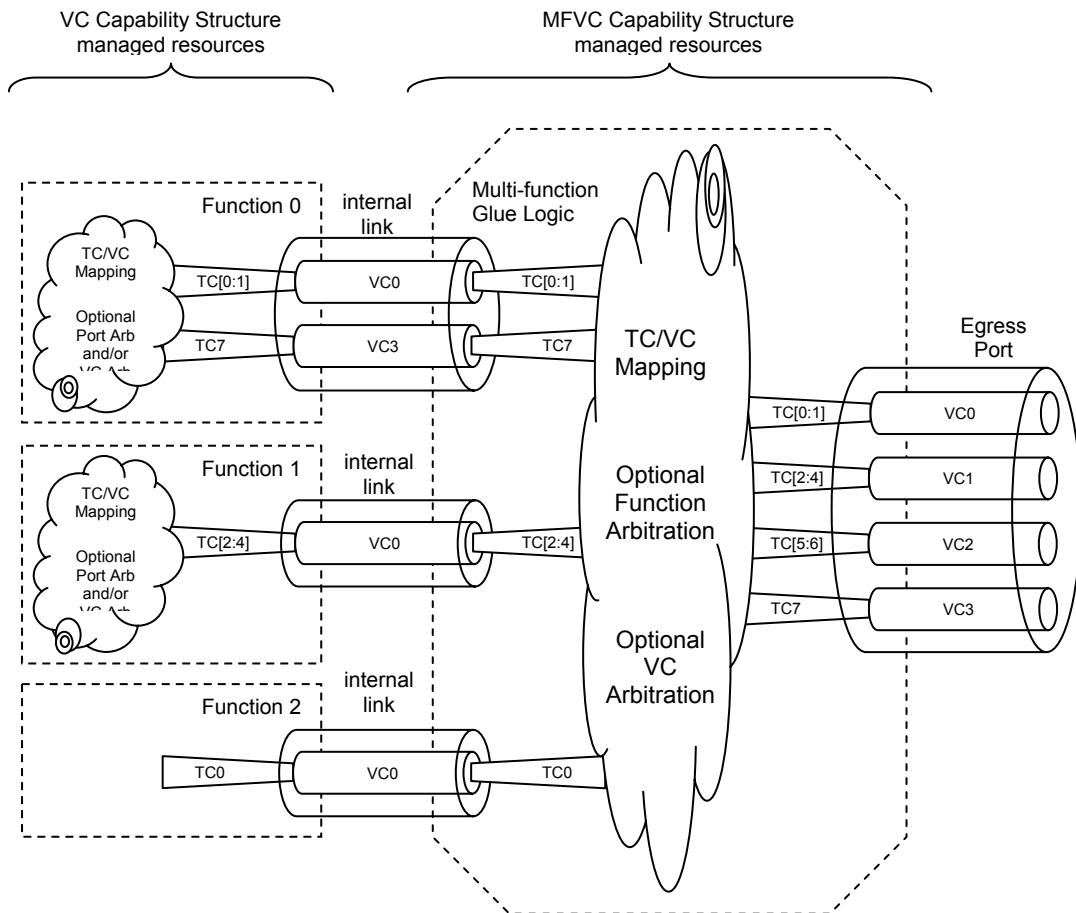


Figure 6-9: Multi-function Arbitration Model

QoS for an upstream request originating at a function is managed as follows. First, a function-specific mechanism applies a TC label to the request. For example, a device driver might configure a function to tag all its requests with TC7.

Next, if the function contains a VC Capability structure, it specifies the TC/VC mapping to one of the function's VC resources (perhaps the function's single VC resource). In addition, the VC Capability structure supports the enablement and configuration of the function's VC resources.

If the function is a switch and the target VC resource supports Port Arbitration, this mechanism governs how the switch's multiple Downstream Ingress Ports arbitrate for that VC resource. If the Port Arbitration mechanism supports time-based WRR, this also governs the injection rate of requests from each Downstream Ingress Port.

If the function supports VC arbitration, this mechanism manages how the function's multiple VC resources arbitrate for the conceptual internal link to the MFVC resources.

Once a request packet conceptually arrives at MFVC resources, address/routing information in the TLP header determines whether the request goes upstream or peer-to-peer to another function. For the case of peer-to-peer, QoS management is left to unarchitected device-specific mechanisms. For the case of upstream, TC/VC mapping in the MFVC Capability structure determines which VC resource the request will target. The MFVC Capability structure also supports enablement and configuration of the VC resources in the multi-function glue logic. If the target VC resource supports Function Arbitration, this mechanism governs how the multiple functions arbitrate for this VC resource. If the Function Arbitration mechanism supports time-based WRR, this governs the injection rate of requests for each function into this VC resource.

Finally, if the MFVC Capability structure supports VC Arbitration, this mechanism governs how the MFVC's multiple VCs compete for the device's Upstream Egress Port. Independent of VC arbitration policy, management/control logic associated with each VC must observe transaction ordering and flow control rules before it can make pending traffic visible to the arbitration mechanism.



IMPLEMENTATION NOTE

Multi-Function Devices without the MFVC Capability Structure

If a multi-function device lacks an MFVC Capability Structure, the arbitration of data flows from different functions of a multi-function device is beyond the scope of this specification. However, if a multi-function device supports TCs other than TC0 and does not implement an MFVC Capability structure, it must implement a single VC Capability structure in Function 0 to provide architected TC/VC mappings for the Link.

Section 6.3.4, p. 283, change text as shown:

6.3.4. Isochronous Support

Servicing isochronous data transfer requires a system to provide not only guaranteed data bandwidth but also deterministic service latency. The isochronous support mechanisms in PCI Express are defined to ensure that isochronous traffic receives its allocated bandwidth over a relevant period of time while also preventing starvation of the other traffic in the system. Isochronous support mechanisms apply to communication between Endpoint and Root Complex as well as to peer-to-peer communication.

Isochronous service is realized through proper use of PCI Express mechanisms such as TC transaction labeling, VC data-transfer protocol, and TC-to-VC mapping. End-to-end isochronous service requires software to set up proper configuration along the path between the Requester to Completer. This section describes the rules for software configuration and the rules hardware components must follow to provide end-to-end isochronous services. More information and background material regarding isochronous applications and isochronous service design guidelines can be found in Appendix A.

6.3.4.1. Rules for Software Configuration

System software must obey the following rules to configure PCI Express fabric for isochronous traffic:

- Software must designate one or more TC labels for isochronous transactions.
- ~~The Software must ensure that setting of~~ the Attribute fields of all isochronous requests targeting the same Completer ~~must be~~ fixed and identical.
- Software must configure all VC resources used to support isochronous traffic to be serviced (arbitrated) at the requisite bandwidth and latency to meet the application objectives. This may be accomplished using strict priority, WRR, or hardware-fixed arbitration.
- ~~For each Switch Egress Port and RCRB that supports isochronous data flows, the associated Port arbitration must support time based arbitration for the configured VC.~~
- Software should not intermix isochronous traffic with non-isochronous traffic on a given VC.
- Software must observe the Maximum Time Slots capability reported by the PCI Express Port or RCRB.
- Software must not assign all PCI Express Link capacity to isochronous traffic. This is required to ensure the requisite forward progress of other non-isochronous transactions to avoid false transaction timeouts.
- Software must limit the Max_Payload_Size for each path that supports isochronous to meet the isochronous latency. For example, all traffic flowing on a path from an isochronous capable device to the Root Complex should be limited to packets that do not exceed the Max_Payload_Size required to meet the isochronous latency requirements.

- ❑ Software must set Max_Read_Request_Size of an isochronous-configured device with a value that does not exceed the Max_Payload_Size set for the device.

6.3.4.2. Rules for Requesters

A Requester requiring isochronous services must obey the following rules:

- ❑ The value in the Length field of read requests must never exceed Max_Payload_Size.
- ❑ If isochronous traffic targets the Root Complex and the RCRB indicates it cannot meet the isochronous bandwidth and latency requirements without requiring all transactions to set the No Snoop attribute bit, indicated by setting the Reject Snoop Transactions field, then this bit must be set within the TLP header else the transaction will be rejected.

6.3.4.3. Rules for Completers

A Completer providing isochronous services must obey the following rules:

- ❑ A Completer should not apply flow control induced backpressure to uniformly injected isochronous requests under normal operating conditions.
- ❑ A Completer must report its isochronous bandwidth capability in the Maximum Time Slots field in the VC Resource Capability register. Note that a Completer must account for partial writes.
- ❑ A Completer must observe the maximum isochronous transaction latency.
- ❑ A Root Complex as a Completer must implement [at least one](#) RCRB and support time-based Port Arbitration [mechanism](#) for the associated VCs. Note that ~~the~~ time-based Port Arbitration only applies to request transactions.

6.3.4.4. Rules for Switch Components

A Switch component providing isochronous services must obey the following rules:

- ❑ An isochronous-configured Switch Port should not apply flow control induced backpressure to uniformly injected isochronous requests under normal operating conditions.
- ❑ An isochronous-configured Switch must observe the maximum isochronous transaction latency.
- ❑ A Switch component must support time-based Port Arbitration [mechanism](#) for each Port that supports one or more VCs capable of supporting isochronous traffic. Note that ~~the~~ time-based Port Arbitration [only](#) applies to request transactions but not to completion transactions.

6.3.4.5. Rules for Multi-function Devices

A multi-function device *that includes* an MFVC Capability structure providing isochronous services must obey the following rules:

- MFVC glue logic configured for isochronous operation should not apply backpressure to uniformly injected isochronous requests from its functions under normal operating conditions.
- The MFVC Capability structure must support time-based Function Arbitration for each VC capable of supporting isochronous traffic. Note that time-based Function Arbitration applies only to upstream request transactions; it does not apply to any downstream or peer-to-peer request transactions, nor to any completion transactions.

A multi-function device *that lacks* an MFVC Capability structure has no architected mechanism to provide isochronous services for its multiple functions concurrently.

Section 7.11, p. 377, change text as shown:

7.11. Virtual Channel Capability

The PCI Express Virtual Channel (VC) capability is an optional extended capability that is required to be implemented by PCI Express Ports of devices that support PCI Express functionality beyond the general purpose I/O traffic, i.e., the default Traffic Class 0 (TC0) over the default Virtual Channel 0 (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 7-38 provides a high level view of the PCI Express Virtual Channel Capability structure for all devices. This structure controls Virtual Channel assignment for PCI Express Links and may be present in Endpoint devices, Switch Ports (Upstream and Downstream), Root Ports and RCRBs. Some registers/fields in the PCI Express Virtual Channel Capability structure may have different interpretation for Endpoint devices, Switch Ports, Root Ports and RCRBs. Software must interpret the PCI Express device/Port Type field (Section 7.8.1) in the PCI Express Capability structure to determine the availability and meaning of these registers/fields.

The PCI Express ~~Virtual Channel~~VC Capability structure ~~is permitted~~ ~~can be present~~ in the Extended Configuration Space of all single-function devices or in RCRBs ~~with the restriction that it is only present in the Extended Configuration Space of Function 0 for multi-function devices at their Upstream Ports.~~

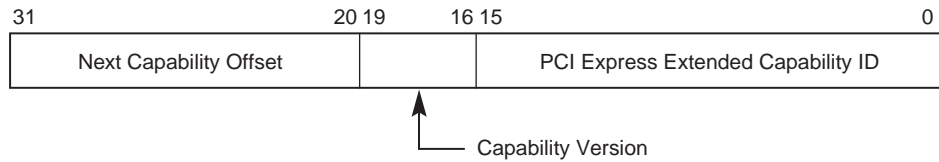
A multi-function device at an Upstream Port may optionally contain a Multi-Function Virtual Channel (MFVC) Capability structure (see Section 7.15). If a multi-function device contains an MFVC Capability structure, any or all of its functions are permitted to contain a VC Capability structure. Per-function VC Capability structures are also permitted for devices inside a Switch that contain only Switch Downstream Port functions, or for Root Complex Integrated Devices. Otherwise, only function 0 is permitted to contain a VC Capability structure.

To preserve software backwards 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.

Section 7.11.1, p. 379, change text as shown:

7.11.1. Virtual Channel Enhanced Capability Header

See Section 7.9.3 for a description of the PCI Express Enhanced Capability header. ~~The Extended Capability ID for the~~ Virtual Channel Capability ~~is~~ must use one of two Extended Capability IDs: 0002h or 0009h. See Section 7.11 for rules governing when each shall be used. Figure 7-39 details allocation of register fields in the Virtual Channel Enhanced Capability header; Table 7-34 provides the respective bit definitions.



OM14526

Figure 7-39: Virtual Channel Enhanced Capability Header

Table 7-34: Virtual Channel Enhanced Capability Header

Bit Location	Register Description	Attributes
15:0	<p>PCI Express Extended Capability ID – This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability.</p> <p>Extended Capability ID for the Virtual Channel Capability is <u>either</u> 0002h <u>or</u> 0009h.</p>	RO
19:16	<p>Capability Version – This field is a PCI-SIG defined version number that indicates the version of the capability structure present.</p> <p>Must be 1h for this version of the specification.</p>	RO
31:20	<p>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.</p> <p>For Extended Capabilities implemented in device 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.</p>	RO

Section 7.11.6, p. 383, change text as shown:

7.11.6. VC Resource Capability Register

The VC Resource Capability register describes the capabilities and configuration of a particular Virtual Channel resource. Figure 7-44 details allocation of register fields in the VC Resource Capability register; Table 7-39 provides the respective bit definitions.

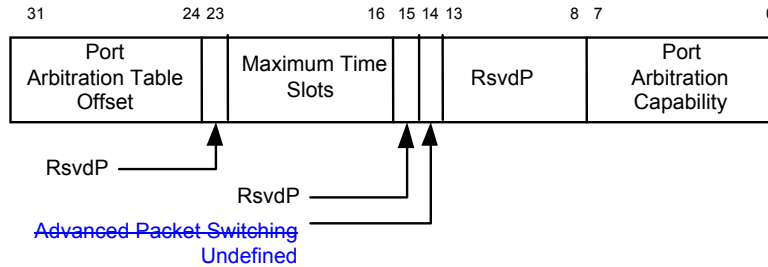


Figure 7-44: VC Resource Capability Register

Table 7-39: VC Resource Capability Register

Bit Location	Register Description	Attributes
14	<p>Advanced Packet Switching— Indicates the VC resource only supports transactions optimized for Advanced Packet Switching (AS). This field is valid for all PCI Express Ports and RCRB.</p> <p>When this field is set to 0, it indicates that the VC resource is capable of supporting all transactions defined by this specification (including AS transport packets).</p> <p>When this field is set to 1, it indicates that the VC resource only supports transactions optimized for Advanced Packet Switching, and, therefore, must not be used for non-AS packet traffic.</p> <p><u>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.</u></p>	RO

Add the following new section at the end of Chapter 7 after page 398 (Note that Section 7.14 is used by another errata):

7.15. Multi-Function Virtual Channel Capability

The Multi-Function Virtual Channel (MFVC) capability is an extended capability required for PCI Express multi-function devices that support functionality beyond the general-purpose IO traffic, i.e. the default Traffic Class (TC0) over the default Virtual Channel (VC0). 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 7-56 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.

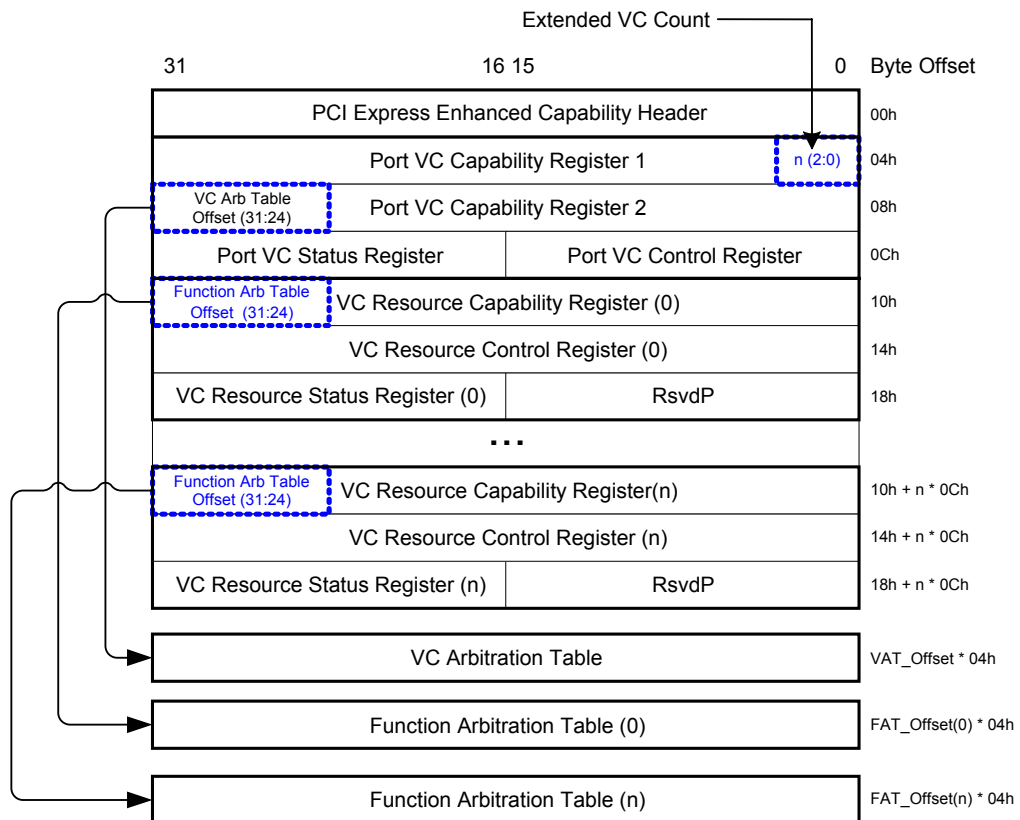
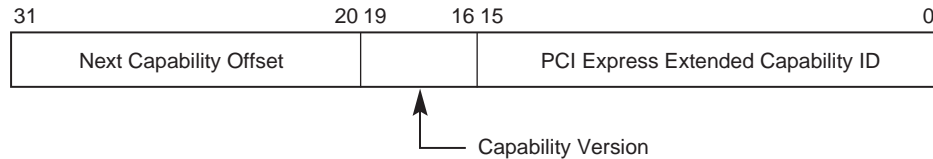


Figure 7-56: PCI Express MFVC Capability Structure

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

7.15.1. MFVC Enhanced Capability Header

See Section 7.9.3 for a description of the PCI Express Enhanced Capability header. The Extended Capability ID for the MFVC Capability is 0008h. Figure 7-57 details allocation of register fields in the MFVC Enhanced Capability header; Table 7-50 provides the respective bit definitions.



OM14526

Figure 7-57: MFVC Enhanced Capability Header

Table 7-50: MFVC Enhanced Capability Header

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>15:0</u>	<u>PCI Express Extended Capability ID</u> – This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability. <u>Extended Capability ID for the MFVC Capability is 0008h.</u>	<u>RO</u>
<u>19:16</u>	<u>Capability Version</u> – This field is a PCI-SIG defined version number that indicates the version of the capability structure present. <u>Must be 1h for this version of the specification.</u>	<u>RO</u>
<u>31:20</u>	<u>Next Capability Offset</u> – 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. <u>For Extended Capabilities implemented in device 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.</u>	<u>RO</u>

7.15.2. 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 7-58 details allocation of register fields in the Port VC Capability register 1; Table 7-51 provides the respective bit definitions.

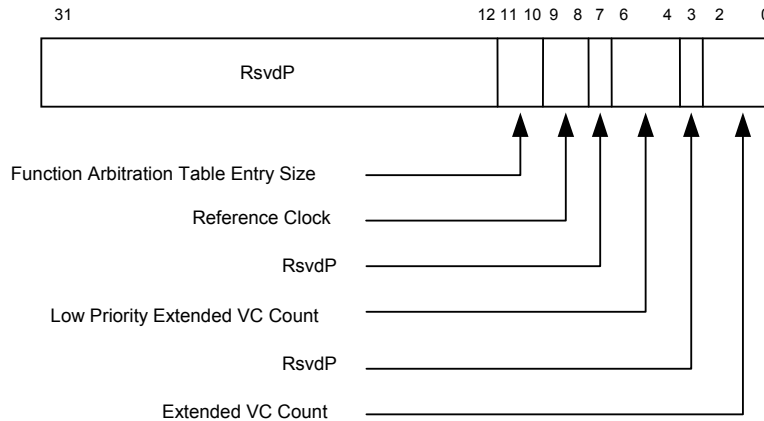


Figure 7-58: Port VC Capability Register 1

Table 7-51: Port VC Capability Register 1

Bit Location	Register Description	Attributes
<u>2:0</u>	<p>Extended VC Count – Indicates the number of (extended) Virtual Channels in addition to the default VC supported by the device.</p> <p>The minimum value of this field is 0 (for devices that only support the default VC). The maximum value is 7.</p>	<u>RO</u>
<u>6:4</u>	<p>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.</p> <p>The minimum value of this field is 0 and the maximum value is Extended VC Count.</p>	<u>RO</u>
<u>9:8</u>	<p>Reference Clock – Indicates the reference clock for Virtual Channels that support time-based WRR Function Arbitration.</p> <p>Defined encodings are:</p> <p><u>00b</u> <u>100 ns reference clock</u></p> <p><u>01b – 11b</u> <u>Reserved</u></p>	<u>RO</u>

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>11:10</u>	<p>Function Arbitration Table Entry Size – Indicates the size (in bits) of Function Arbitration table entry in the device.</p> <p>Defined encodings are:</p> <p><u>00b</u> Size of Function Arbitration table entry is 1 bit</p> <p><u>01b</u> Size of Function Arbitration table entry is 2 bits</p> <p><u>10b</u> Size of Function Arbitration table entry is 4 bits</p> <p><u>11b</u> Size of Function Arbitration table entry is 8 bits</p>	<u>RO</u>

7.15.3. 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 7-59](#) details allocation of register fields in the Port VC Capability register 2; [Table 7-52](#) provides the respective bit definitions.



Figure 7-59: Port VC Capability Register 2

Table 7-52: Port VC Capability Register 2

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>7:0</u>	<p>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.</p> <p>Each bit location within this field corresponds to a VC Arbitration capability defined below. When more than one bit in this field is set, it indicates that the device can be configured to provide different VC arbitration services.</p> <p>Defined bit positions are:</p> <p><u>Bit 0</u> Hardware fixed arbitration scheme, e.g., Round Robin</p> <p><u>Bit 1</u> Weighted Round Robin (WRR) arbitration with 32 phases</p> <p><u>Bit 2</u> WRR arbitration with 64 phases</p> <p><u>Bit 3</u> WRR arbitration with 128 phases</p> <p><u>Bits 4-7</u> Reserved</p>	<u>RO</u>

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
31:24	<p>VC Arbitration Table Offset – Indicates the location of the VC Arbitration Table.</p> <p>This field contains the zero-based offset of the table in <u>DQWORDS (16 bytes) from the base address of the MFVC Capability structure. A value of 0 indicates that the table is not present.</u></p>	<u>RO</u>

7.15.4. Port VC Control Register

Figure 7-60 details allocation of register fields in the Port VC Control register; Table 7-53 provides the respective bit definitions.

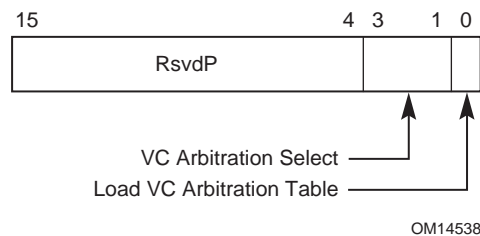


Figure 7-60: Port VC Control Register

Table 7-53: Port VC Control Register

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>0</u>	<p>Load VC Arbitration Table – Used for software to update the VC Arbitration Table. This field is valid when the selected VC Arbitration uses the VC Arbitration Table.</p> <p>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.</p> <p>This bit always returns 0 when read.</p>	<u>RW</u>
<u>3:1</u>	<p>VC Arbitration Select – Used for 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.</p> <p>The value of this field is the number corresponding to one of the asserted bits in the VC Arbitration Capability field.</p> <p>This field cannot be modified when more than one VC in the LPVC group is enabled.</p>	<u>RW</u>

7.15.5. 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 7-61 details allocation of register fields in the Port VC Status register; Table 7-54 provides the respective bit definitions.

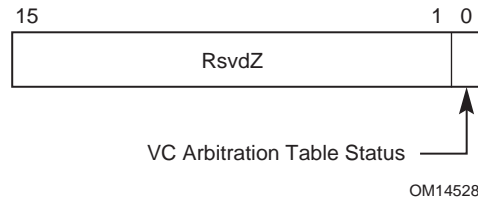


Figure 7-61: Port VC Status Register

Table 7-54: Port VC Status Register

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>0</u>	<p>VC Arbitration Table Status – Indicates the coherency status of the VC Arbitration Table. This field is valid when the selected VC uses the VC Arbitration Table.</p> <p>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.</p> <p>Default value of this field is 0.</p>	<u>RO</u>

7.15.6. VC Resource Capability Register

The VC Resource Capability register describes the capabilities and configuration of a particular Virtual Channel resource. Figure 7-62 details allocation of register fields in the VC Resource Capability register; Table 7-55 provides the respective bit definitions.

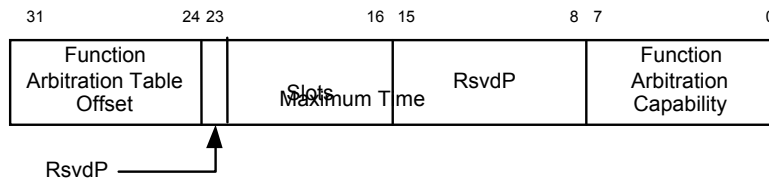


Figure 7-62: VC Resource Capability Register

Table 7-55: VC Resource Capability Register

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>7:0</u>	<p><u>Function Arbitration Capability</u> – Indicates types of Function Arbitration supported by the VC resource.</p> <p><u>Each bit location within this field corresponds to a Function Arbitration capability defined below. When more than one bit in this field is set, it indicates that the VC resource can be configured to provide different arbitration services.</u></p> <p><u>Software selects among these capabilities by writing to the Function Arbitration Select field (see below).</u></p> <p><u>Defined bit positions are:</u></p> <p><u>Bit 0 Non-configurable hardware-fixed arbitration scheme, e.g., Round Robin (RR)</u></p> <p><u>Bit 1 Weighted Round Robin (WRR) arbitration with 32 phases</u></p> <p><u>Bit 2 WRR arbitration with 64 phases</u></p> <p><u>Bit 3 WRR arbitration with 128 phases</u></p> <p><u>Bit 4 Time-based WRR with 128 phases</u></p> <p><u>Bit 5 WRR arbitration with 256 phases</u></p> <p><u>Bits 6-7 Reserved</u></p>	<u>RO</u>
<u>22:16</u>	<p><u>Maximum Time Slots</u> – 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 Function Arbitration. <u>For example, a value 0 in this field indicates the supported maximum number of time slots is 1 and a value of 127 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.</u></p>	<u>HwInit</u>
<u>31:24</u>	<p><u>Function Arbitration Table Offset</u> – Indicates the location of the Function Arbitration Table associated with the VC resource.</p> <p><u>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 0 indicates that the table is not present.</u></p>	<u>RO</u>

7.15.7. VC Resource Control Register

Figure 7-63 details allocation of register fields in the VC Resource Control register; Table 7-56 provides the respective bit definitions.

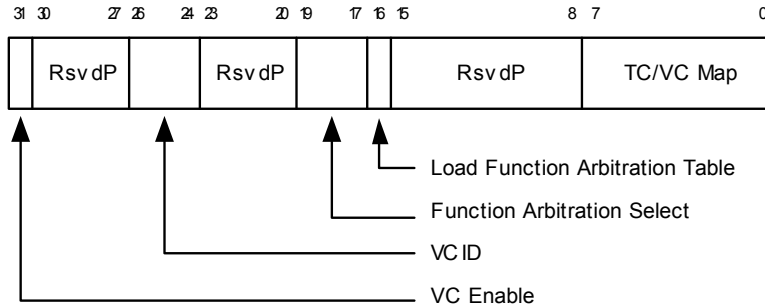


Figure 7-63: VC Resource Control Register

Table 7-56: VC Resource Control Register

Bit Location	Register Description	Attributes
7:0	<p>TC/VC Map – This field indicates the TCs that are mapped to the VC resource.</p> <p>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 one bit in this field is set, it indicates that multiple TCs are mapped to the VC resource.</p> <p>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.</p> <p>Default value of this field is FFh for the first VC resource and is 00h for other VC resources.</p> <p><u>Note:</u></p> <p>Bit 0 of this field is read-only. It must be set to 1 for the default VC 0 and set to 0 for all other enabled VCs.</p>	<p>RW</p> <p>(see the note for exceptions)</p>

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
<u>16</u>	<p><u>Load Function Arbitration Table</u> – This bit, when set, 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).</p> <p>Software sets this bit to signal hardware to update Function Arbitration logic with new values stored in 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.</p> <p><u>This bit always returns 0 when read.</u></p> <p><u>Default value of this field is 0.</u></p>	<u>RW</u>
<u>19:17</u>	<p><u>Function Arbitration Select</u> – This field configures the VC resource to provide a particular Function Arbitration service.</p> <p><u>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.</u></p>	<u>RW</u>
<u>26:24</u>	<p><u>VC ID</u> – This field assigns a VC ID to the VC resource (see note for exceptions).</p> <p><u>This field cannot be modified when the VC is already enabled.</u></p> <p><u>Note:</u></p> <p><u>For the first VC resource (default VC), this field is a read-only field that must be set to 0 ('hard-wired').</u></p>	<u>RW</u>

<u>Bit Location</u>	<u>Register Description</u>	<u>Attributes</u>
31	<p>VC Enable – This field, when set, enables a Virtual Channel (see note 1 for exceptions). The Virtual Channel is disabled when this field is cleared.</p> <p>Software must use the VC Negotiation Pending bit to check whether the VC negotiation is complete. When VC Negotiation Pending bit is cleared, a 1 read from this VC Enable bit indicates that the VC is enabled (Flow Control initialization is completed for the PCI Express Port); a 0 read from this bit indicates that the Virtual Channel is currently disabled.</p> <p>Default value of this field is 1 for the first VC resource and is 0 for other VC resource(s).</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. This bit is hardwired to 1 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

7.15.8. VC Resource Status Register

Figure 7-64 details allocation of register fields in the VC Resource Status register; Table 7-57 provides the respective bit definitions.

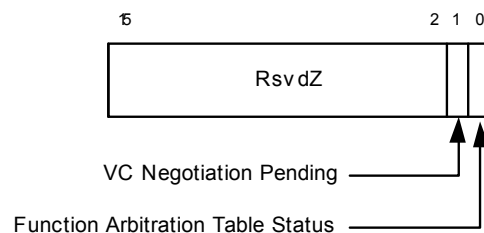


Figure 7-64: VC Resource Status Register

Table 7-57: VC Resource Status Register

Bit Location	Register Description	Attributes
<u>0</u>	<p>Function Arbitration Table Status – This bit indicates the coherency status of the Function Arbitration Table associated with the VC resource. This field is valid only when the Function Arbitration Table is used by the selected Function Arbitration for the VC resource.</p> <p>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 field.</p> <p>Default value of this field is 0.</p>	<u>RO</u>
<u>1</u>	<p>VC Negotiation Pending – This bit indicates whether the Virtual Channel negotiation (initialization or disabling) is in pending state.</p> <p>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.</p> <p>Before using a Virtual Channel, software must check whether the VC Negotiation Pending fields for that Virtual Channel are cleared in both components on a Link.</p>	<u>RO</u>

7.15.9. 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 for details.

7.15.10. 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 corresponds to a phase within a Function Arbitration period.

The table entry size 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.

A Function Number written to a table entry indicates that the phase within the Function Arbitration period is assigned to the selected Function (the Function 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 indicated by the Function 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 without delay if the Function 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 indicated by the Function 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 that does not exist, or
 - the Function 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 determines the table entry size. The length of the table is determined by the Function Arbitration Select field as shown in Table 7-58.

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 active Functions 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 7-58: Length of Function Arbitration Table

<u>Function Arbitration Select</u>	<u>Function Arbitration Table Length (in number of Entries)</u>
<u>001b</u>	<u>32</u>
<u>010b</u>	<u>64</u>
<u>011b</u>	<u>128</u>
<u>100b</u>	<u>128</u>
<u>101b</u>	<u>256</u>