



PCI-SIG ENGINEERING CHANGE NOTICE

| | |
|---------------------------|---|
| TITLE: | Function Level Reset (FLR) |
| DATE: | July 12, 2005, Approved 27 June 2006 |
| AFFECTED DOCUMENT: | PCI Express Base Specification, Rev. 1.1 |
| SPONSOR: | Intel Corporation, Microsoft Corporation, Hewlett-Packard |

Note: Based on feedback received, this copy includes minor (non content) clarifications from the copy sent for membership review – these are highlighted

Part I

1. Summary of the Functional Changes

Adds an architected method to initiate a Function-level reset (FLR) on a PCIe Function, independent of any associated system or Function-specific driver.

2. Benefits as a Result of the Changes

5 The new FLR mechanism improves the robustness and security of several usage models including models where a Function can be reassigned from one virtual machine to another or from one physical host to another.

3. Assessment of the Impact

The FLR mechanism is a new optional capability. For Functions implementing the capability, the impact will depend highly on the nature of the device. An implementation note is included to address several likely implementation concerns.

10

4. Analysis of the Hardware Implications

Hardware supporting this optional capability must provide for Function-independent reset, including the ability to reset device-specific state associated with a single Function.

15

5. Analysis of the Software Implications

FLR has no impact on existing software. New software that supports FLR must follow the rules described below.

Part II Detailed Description of the change

20

Add to the Terms and Acronyms:

Conventional Reset A Hot, Warm or Cold reset; Distinct from Function Level Reset (FLR)

25

...

FLR, Function Level Reset A mechanism for resetting a specific Function of an Endpoint (see Section 6.6.2)

Fundamental Reset

A hardware mechanism for setting or returning all Port states to the initial conditions specified in this document (see Section 6.6).

30

Change Section 2.3.1 as follows:

2.3.1. Request Handling Rules

...

35 If the Request arrives between the time an FLR has been initiated and the completion of the FLR by the targeted Function, the Request is permitted to be silently discarded (following update of flow control credits) without logging or signaling it as an error. It is recommended that the Request be handled as an Unsupported Request (UR).

❑ Otherwise (supported Request Type, not a Message), process the Request

• ...

40

- For Configuration Requests only, following reset it is possible for a Function to terminate the request but indicate that it is temporarily unable to process the Request, but will be able to process the Request in the future – in this case, the Configuration Request Retry Status (CRS) Completion Status is used (see Section 6.6). Valid reset conditions after which a device is permitted to return CRS are:

45

- ◆ Cold, Warm and Hot Link Resets
- ◆ FLRs
- ◆ A reset initiated in response to a D3_{hot} to D0uninitialized device state transition.

50

A ~~Function~~ Function is explicitly not permitted to return CRS following a software-initiated reset (other than an FLR) of the device, e.g., by the device's software driver

writing to a device-specific reset bit. Additionally, a Function is not permitted to return CRS after having previously returned a Successful Completion without an intervening valid reset (i.e., FLR or Conventional Reset) condition.

Change Section 2.3.2 as follows:

2.3.2. Completion Handling Rules

55 ...

Completions with a Completion Status other than Successful Completion, or Configuration Request Retry Status (in response to Configuration Request only) must cause the Requester to:

60

- Free Completion buffer space and other resources associated with the Request.
- Handle the error via a Requester-specific mechanism (see Section 6.2.3.2.5).

65

If the Completion arrives between the time an FLR has been initiated and the completion of the FLR by the targeted Function, the Completion is permitted to be handled as an Unexpected Completion or to be silently discarded (following update of flow control credits) without logging or signaling it as an error. Once the FLR has completed, received Completions corresponding to Requests issued prior to the FLR must be handled as Unexpected Completions, unless the Function has been re-enabled to issue Requests.

Change Section 3.2.1 as follows:

3.2.1. Data Link Control and Management State Machine Rules

70

Rules per state:

DL_Inactive

- Initial state following PCI Express hot, warm, or cold reset. Note that DL states are unaffected by an FLR (see Section 6.6).

Change Section 5.3.1.1. as follows:

75

5.3.1.1. D0 State

All PCI Express ~~F~~functions must support the D0 state. D0 is divided into two distinct sub-states, the “un-initialized” sub-state and the “active” sub-state. When a PCI Express component ~~initially has its power applied~~ comes out of Conventional Reset or FLR, it defaults to the D0_{uninitialized} state. Components that are in this state will be enumerated and configured by the PCI Express Hierarchy enumeration process. Following the completion of the enumeration and configuration process, the ~~F~~function enters the D0_{active} state, the fully operational state for a PCI Express ~~F~~function. A ~~F~~function enters the D0_{active} state whenever any single or combination of the ~~F~~function’s Memory Space Enable, I/O Space Enable, or Bus Master Enable bits have been enabled by system software

...

Change Section 6.6 as follows:

6.6. PCI Express Reset - Rules

This section specifies the ~~behavior of~~ PCI Express ~~Fundamental~~ Reset mechanisms. This section covers the relationship between the architectural mechanisms defined in this document and the reset mechanisms defined in this document. Any relationship between the PCI Express Conventional Reset and component or platform reset is component or platform specific (respectively, except as explicitly noted).

6.6.1 Conventional Reset

Conventional Reset includes all reset mechanisms other than Function Level Reset. There are two categories of Conventional Resets: Fundamental Reset and resets that are not Fundamental Reset. This section applies to all types of Conventional reset.

In all form factors and system hardware configurations, there must, at some level, be a hardware mechanism for setting or returning all Port states to the initial conditions specified in this document – this mechanism is called “Fundamental Reset”. This mechanism can take the form of an auxiliary signal provided by the system to a component or adapter card, in which case the signal must be called PERST#, and must conform to the rules specified in Section 4.2.4.5.1. When PERST# is provided to a component or adapter, this signal must be used by the component or adapter as Fundamental Reset. When PERST# is not provided to a component or adapter, Fundamental Reset is generated autonomously by the component or adapter, and the details of how this is done are outside the scope of this document. If a Fundamental Reset is generated autonomously by the component or adapter, and if power is supplied by the platform to the component/adapter, the component/adapter must generate a Fundamental Reset to itself if the supplied power goes outside of the limits specified for the form factor or system.

❑ There are three distinct types of Conventional Reset: cold, warm, and hot:

- A Fundamental Reset must occur following the application of power to the component. This is called a cold reset.
- In some cases, it may be possible for the Fundamental Reset mechanism to be triggered by hardware without the removal and re-application of power to the component. This is called a warm reset. This document does not specify a means for generating a warm reset.
- There is an in-band mechanism for propagating Conventional Reset across a Link. This is called a hot reset and is described in Section 4.2.4.5.

Note also that the Data Link Layer reporting DL_Down is in some ways identical to a hot reset – see Section 2.9.

- ❑ On exit from any type of Conventional Reset (cold, warm, or hot), all Port registers and state machines must be set to their initialization values as specified in this document, except for sticky registers (see Section 7.4 and Section 7.6).
 - Note that, from a device point of view, any type of Conventional Reset (cold, warm, hot, or DL_Down) has the same effect at the Transaction Layer and above as would RST# assertion and de-assertion in conventional PCI.

- ❑ On exit from a Fundamental Reset, the Physical Layer will attempt to bring up the Link (see Section 4.2.5). Once both components on a Link have entered the initial Link Training state, they will proceed through Link initialization for the Physical Layer and then through Flow Control initialization for VC0, making the Data Link and Transaction Layers ready to use the Link

- Following Flow Control initialization for VC0, it is possible for TLPs and DLLPs to be transferred across the Link

Following a Conventional Reset, some Functions may require additional time before they are able to respond to Requests they receive. Particularly for Configuration Requests it is necessary that components and Functions behave in a deterministic way, which the following rules address.

The first set of rules addresses requirements for components and Functions:

- ❑ A component must enter the LTSSM Detect state within 20 ms of the end of Fundamental Reset (Link Training is described in Section 4.2.4)
 - Note: In some systems, it is possible that the two components on a Link may exit Fundamental Reset at different times. Each component must observe the requirement to enter the initial active Link Training state within 20 ms of the end of Fundamental Reset from its own point of view.

- ❑ On the completion of Link Training (entering the DL_Active state, see Section 3.2), a component must be able to receive and process TLPs and DLLPs

The second set of rules addresses requirements placed on the system:

- 150 To allow components to perform internal initialization, system software must wait for at least 100 ms from the end of a Conventional Reset of one or more devices before it is permitted to issue Configuration Requests to those devices
- A system must guarantee that all components intended to be software visible at boot time are ready to receive Configuration Requests within 100 ms of the end of Conventional~~Fundamental~~ Reset at the Root Complex – how this is done is beyond the scope of this specification
- 155 The Root Complex and/or system software must allow at least 1.0 s ~~(+50%/–0%)~~ after a Conventional Reset of a device, before it may determine that a device which fails to return a Successful Completion status for a valid Configuration Request is a broken device
- 160 Note: This delay is analogous to the T_{rhfa} parameter specified for PCI/PCI-X, and is intended to allow an adequate amount of time for devices which require self initialization.
- ...

6.6.2 Function-Level Reset (FLR)

165 The FLR mechanism enables software to quiesce and reset Endpoint hardware with Function-level granularity. Three example usage models illustrate the benefits of this feature:

- 170 In some systems, it is possible that the software entity that controls a Function will cease to operate normally. To prevent data corruption, it is necessary to stop all PCI Express and external IO (not PCI Express) operations being performed by the Function. Other defined reset operations do not guarantee that external IO operations will be stopped.
- 175 In a partitioned environment where hardware is migrated from one partition to another, it is necessary to ensure that no residual “knowledge” of the prior partition be retained by hardware, for example, a user’s secret information entrusted to the first partition but not to the second. Further, due to the wide range of Functions, it is necessary that this be done in a Function-independent way.
- 180 When system software is taking down the software stack for a Function and then rebuilding that stack, it is sometimes necessary to return the state to an uninitialized state before rebuilding the Function’s software stack.

Implementation of FLR is optional (not required), but is strongly recommended.

180 FLR applies on a per Function basis. Only the targeted Function is affected by the FLR operation. The Link state must not be affected by an FLR.

FLR modifies the Function state described by this specification as follows:

- Function registers and Function-specific state machines must be set to their initialization values as specified in this document, except for the following:
 - sticky-type registers (ROS, RWS, RW1CS)

- 185 ○ Registers defined as type HwInit
- these other registers:
 - Captured Slot Power Limit Value in the Device Capabilities Register
 - Captured Slot Power Limit Scale in the Device Capabilities Register
 - Max Payload Size in the Device Control register
 - 190 ▪ Active State Power Management (ASPM) Control in the Link Control Register
 - Read Completion Boundary (RCB) in the Link Control Register
 - Common Clock Configuration in the Link Control Register
 - Extended Synch in the Link Control Register
 - 195 ▪ Enable Clock Power Management in the Link Control Register
 - All registers in the Virtual Channel capability structure
 - All registers in the Multi-Function Virtual Channel capability structure

200 Note that the controls that enable the Function to initiate requests on PCI Express are cleared, including Bus Master Enable, MSI interrupt enable, and the like, effectively causing the Function to become quiescent on the link.

Note that port state machines associated with Link functionality including those in the Physical and Data Link Layers are not reset by FLR, and VC0 remains initialized following an FLR.

- 205 Any outstanding INTx interrupt asserted by the function must be de-asserted by sending the corresponding Deassert INTx Message prior to starting the FLR.

Note that when the FLR is initiated to a Function of a Multi-Function device, if another Function continues to assert a matching INTx, no Deassert INTx Message will be transmitted.

210 After an FLR has been initiated by writing a 1b to the Initiate Function Level Reset bit, the Function must complete the FLR within 100ms. If software initiates an FLR when the Transactions Pending bit is 1b, then software must not initialize the Function until allowing adequate time for any associated Completions to arrive, or to achieve reasonable certainty that any remaining Completions will never arrive. For this purpose, it is recommended that software allow as much time as provided by the pre-FLR value for Completion Timeout on the device. If Completion Timeouts were disabled on the Function when FLR was issued,

215 then the delay is system dependent but must be no less than 100ms.

Note that upon receipt of an FLR, a device may either clear all transaction status including Transactions Pending or set the Completion Timeout to its default value so that all pending transactions will time out during FLR execution. Regardless, the Transactions Pending bit must be clear upon completion of the FLR.

220

Because FLR modifies Function state not described by this specification (in addition to state that is described by this specification), it is necessary to specify the behavior of FLR using a

225 set of criteria that, when applied to the Function, show that the Function has satisfied the requirements of FLR. The following criteria must be applied using Function-specific knowledge to evaluate the Function's behavior in response to an FLR:

- 230 The Function must not give the appearance of an initialized adapter with an active host on any external interfaces controlled by that Function. The steps needed to terminate activity on external interfaces are outside of the scope of this specification.
 - 235 For example, a network adapter must not respond to queries that would require adapter initialization by the host system or interaction with an active host system, but is permitted to perform actions that it is designed to perform without requiring host initialization or interaction. If the network adapter includes multiple Functions that operate on the same external network interface, this rule affects only those aspects associated with the particular function reset by FLR.
- 240 The Function must not retain within itself software readable state that potentially includes secret information associated with any preceding use of the Function. Main host memory assigned to the Function must not be modified by the Function.
 - 245 For example, a Function with internal memory readable directly or indirectly by host software must clear or randomize that memory.
- 250 The Function must return to a state such that normal configuration of the Function's PCI Express interface will cause it to be useable by drivers normally associated with the Function

When an FLR is initiated, the targeted Function must behave as follows:

- 245 The Function must return the Completion for the configuration write that initiated the FLR operation and then initiate the FLR.
- 250 While an FLR is in progress:
 - 255 If a Request arrives, the Request is permitted to be silently discarded (following update of flow control credits) without logging or signaling it as an error.
 - 255 If a Completion arrives, the Completion is permitted to be handled as an Unexpected Completion or to be silently discarded (following update of flow control credits) without logging or signaling it as an error.
 - 255 While a Function is required to complete the FLR operation within the time limit described above, the subsequent Function-specific initialization sequence may require additional time. If additional time is required, the Function must return a Configuration Request Retry Status (CRS) Completion Status when a Configuration Request is received after the time limit above. After the Function responds to a Configuration Request with a Completion status other than CRS, it is not permitted to return CRS until it is reset again.



IMPLEMENTATION NOTE

Avoiding Data Corruption From Stale Completions

260 An FLR causes a Function to lose track of any outstanding nonposted Requests. Any
corresponding Completions that later arrive are referred to as being "stale". If software
issues an FLR while there are outstanding Requests, and then re-enables the Function for
operation without waiting for potential stale Completions, any stale Completions that arrive
265 afterwards may cause data corruption by being mistaken by the Function as belonging to
Requests issued since the FLR.

Software can avoid data corruption from stale Completions in a variety of ways. Here's a
possible algorithm:

-
- 270 1. Software that's performing the FLR synchronizes with other software that might
potentially access the Function directly, and ensures such accesses don't occur during
this algorithm.
 2. Software clears the entire Command register, disabling the Function from issuing any
new Requests.
 - 275 3. Software polls the Transactions Pending bit in the Device Status register either until
it's clear or until it's been long enough that software is reasonably certain that
Completions associated with any remaining outstanding Transactions will never
arrive. On many platforms, the Transactions Pending bit will usually clear within a
few milliseconds, so software might choose to poll during this initial period using a
tight software loop. On rare cases when the Transactions Pending bit doesn't clear
by this time, software will need to poll for a much longer platform-specific period
280 (potentially seconds), so software might choose to conduct this polling using a timer-
based interrupt polling mechanism.
 4. Software initiates the FLR.
 5. Software waits 100 msec.
 6. Software reconfigures the Function and enables it for normal operation.

285

Change Section 7.4 as follows:

7.4. Configuration Register Types

Configuration register fields are assigned one of the attributes described in Table 7-2. All
PCI Express components, with exception of the Root Complex and system-integrated
290 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 enabled for re-initialization by a cold reset (see Section 6.6.1) or a platform specific mechanism. <u>HwInit register bits are not modified by an FLR.</u> |
| RO | Read-only register: Register bits are read-only and cannot be altered by software. Register bits may be initialized by hardware mechanisms such as pin strapping or serial EEPROM. |
| RW | Read-Write register: Register bits are read-write and may be either set or cleared by software to the desired state. |
| RW1C | Read-only status, Write-1-to-clear status register: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect. |
| ROS | Sticky - Read-only register: Registers are read-only and cannot be altered by software. <u>Registers-Bits are not-neither</u> initialized <u>nor</u> modified by hot reset <u>or FLR</u> . Where noted, devices that consume AUX power must preserve sticky register values when AUX power consumption (either via AUX power or PME Enable) is enabled. In these cases, registers are <u>not-neither</u> initialized <u>nor</u> modified by hot, warm, or cold reset (see Section 6.6). |

| Register Attribute | Description |
|--------------------|--|
| RWS | <p>Sticky - Read-Write register: Registers are read-write and may be either set or cleared by software to the desired state. Bits are not<u>neither</u> initialized <u>nor</u> modified by hot reset <u>or FLR</u>.</p> <p>Where noted, devices that consume AUX power must preserve sticky register values when AUX power consumption (either via AUX power or PME Enable) is enabled. In these cases, registers are not<u>neither</u> initialized <u>nor</u> modified by hot, warm, or cold reset (see Section 6.6).</p> |
| RW1CS | <p>Sticky - Read-only status, Write-1-to-clear status register: Registers indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1CS bits has no effect. Bits are not<u>neither</u> initialized <u>nor</u> modified by hot reset <u>or FLR</u>.</p> <p>Where noted, devices that consume AUX power must preserve sticky register values when AUX power consumption (either via AUX power or PME Enable) is enabled. In these cases, registers are not<u>neither</u> initialized <u>nor</u> modified by hot, warm, or cold reset (see Section 6.6).</p> |
| RsvdP | Reserved and Preserved: Reserved for future RW implementations. Registers are read-only and must return 0 when read. Software must preserve the value read for writes to bits. |
| RsvdZ | Reserved and Zero: Reserved for future RWIC implementations. Registers are read-only and must return 0 when read. Software must use 0 for writes to bits. |

In Section 7.6, Table 7-8, change as shown:

| Bit Location | Register Description | Attributes |
|--------------|--|------------|
| 8 | <p>PME Enable – No added requirements</p> <p>Note: Devices that consume AUX power must preserve the value of this sticky register when AUX power is available. In such devices, this register value is not modified by <u>Conventional Reset or FLR</u>hot, warm, or cold reset.</p> | Unchanged |

| Bit Location | Register Description | Attributes |
|--------------|--|------------|
| 15 | PME Status – No added requirements Note: Devices that consume AUX power must preserve the value of this sticky register when AUX power is available. In such devices, this register value is not modified by <u>Conventional Reset or FLRhot, warm, or cold reset.</u> | Unchanged |
| ... | | |

295 ...

Change Section 7.8.3 Device Capabilities Register as follows:

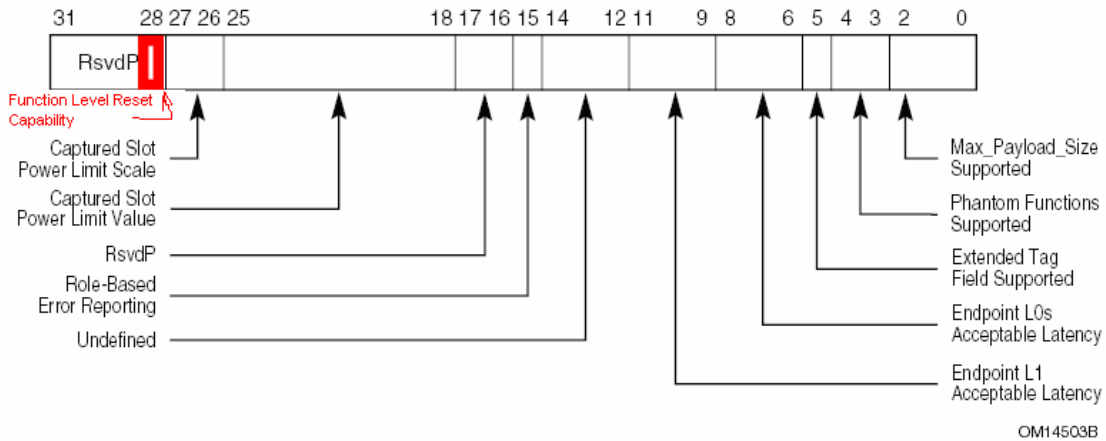


Figure 7-12: Device Capabilities Register

...

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

Change Section 7.8.4 Device Control Register as follows:

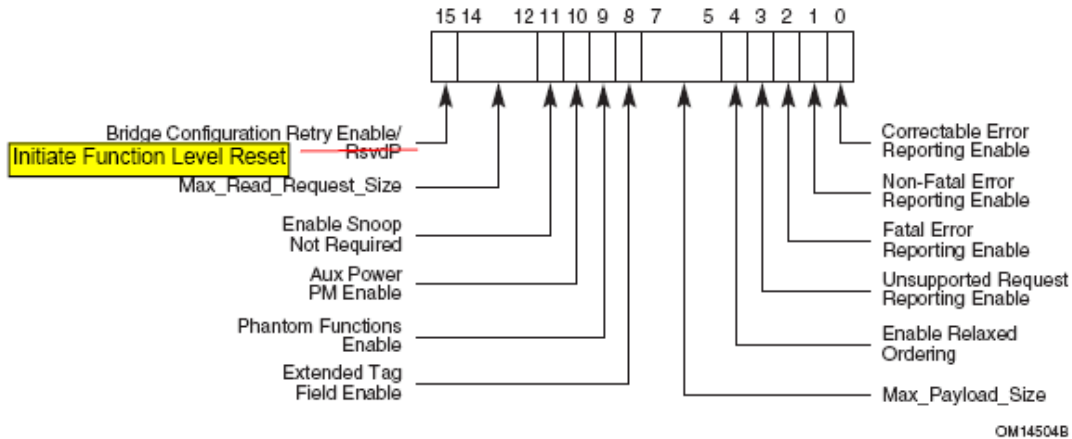


Figure 7-13: Device Control Register

300

...

| Bit Location | Register Description | Attributes |
|--------------|---|---|
| ... | | |
| 15 | <p><i>PCI Express to PCI/PCI-X Bridges:</i> 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 <i>PCI Express to PCI/PCI-X Bridge Specification, Rev. 1.0</i> for further details. Default value of this field is 0b.</p> <p><u>Endpoints with Function Level Reset Capability set to 1b:</u> 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.</p> <p>All others: Reserved – Must hardwire the field to 0b.</p> | <p>PCI Express to PCI/PCI-X Bridges: RW</p> <p><u>FLR Capable Endpoints:</u> RW</p> <p>All others: RsvdP</p> |

In Section 7.8.5:

Table 7-13: Device Status Register

| Bit Location | Register Description | Attributes |
|--------------|----------------------|------------|
|--------------|----------------------|------------|

| Bit Location | Register Description | Attributes |
|--------------|--|------------|
| 5 | <p>Transactions Pending –</p> <p><i>Endpoints:</i> This bit when set indicates that the device has issued Non-Posted Requests which have not been completed. A device reports this bit cleared only when all outstanding Non-Posted Requests have completed or have been terminated by the Completion Timeout mechanism. <u>This bit must also be cleared upon the completion of an FLR.</u></p> <p><i>Root and Switch Ports:</i> This bit when set 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.</p> | RO |

305