Initialization & Management of the PES32NT24G2 PCI Express® Switch

Application Note AN-723

Notes

By Cesar Talledo

Introduction

The PES32NT24G2 PCI Express (PCIe) switch is highly flexible in its configurability. The switch may be configured with multiple switch partitions, each of which can have a configurable number of ports. Ports can be configured in one of several modes, and may include PCI-to-PCI bridge, Non-Transparent Bridge (NTB), and Direct Memory Access (DMA) functions. The switch partitions and ports can be dynamically reconfigured during run-time, either by software or automatically by the hardware.

In some systems, it is expected that the PES32NT24G2 will require some initial configuration prior to normal system operation[1]. This initial configuration, referred to as “boot-time initialization”, may be done by hard-wiring switch pins on the board, by configuring the switch to read instructions from a serial EEPROM, or by a switch manager device issuing configuration commands to the switch.

In addition to boot-time initialization, in some systems it is expected that the switch will be actively managed during run-time by a switch manager device. Depending on the system’s complexity, the switch manager could be in charge of monitoring the switch’s status, communicating with PCIe roots attached to the switch, and reconfiguring the switch dynamically.

This application note describes key aspects related to boot-time initialization and active management of the PES32NT24G2 switch.

Background

Before proceeding with the main topic of this application note, a brief background on switch partitioning, switch management interfaces, and the switch’s global address space is required. These concepts are important in order to understand the manner in which the switch is initialized and managed.

Switch Partitioning

Switch partitioning allows the logical division of the PCIe switch into multiple partitions (up to 8), each of which is composed of a configurable number of ports, and each of which connects to a separate PCIe domain[2]. Each switch partition is logically isolated from the other partitions.

From the switch’s perspective, a switch partition represents a logical container that contains switch ports associated with a PCIe domain. Any switch port can be configured to belong to one partition. The assignment of ports to partitions is left to the system designer. A partition may be configured with zero, one, or many ports, although certain rules apply (see below) regarding valid partition configurations. Figure 1 shows a PES32NT24G2 configured with 3 partitions.

A partition can be placed in one of three modes: disabled, active, and reset. When a partition is disabled, all ports associated with the partition (if any) are disabled and can’t be used. When a partition is active, all ports associated with the partition are active. When a partition is reset, all ports associated with the partition are in PCIe fundamental reset.

[1] In this context, normal system operation refers to the ability of the PCIe roots to start enumeration of the switch.

[2] A PCIe domain is the collection of PCIe devices under a common processor/memory complex (i.e., root-complex), and sharing common PCIe memory, I/O, and configuration spaces.

Notes

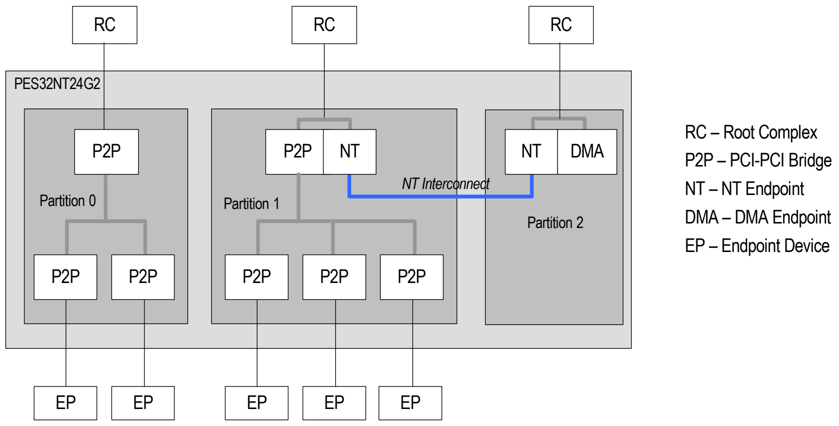


Figure 1 The PES32NT24G2 Configured with 3 Switch Partitions

As shown in Figure 1, each port can be placed in one of several modes. Depending on the mode, ports may contain between one and three PCIe functions. The PCI-to-PCI bridge function allows the port to forward packets across the same partition. By placing two or more ports with a PCI-to-PCI bridge function within a partition, a PCI Express switch is logically created within that partition. On the other hand, the NT endpoint function serves as a gateway to other PCIe domains via the non-transparent interconnect (shown in blue in Figure 1). Finally, the DMA endpoint function provides a DMA engine to off-load the partition's host during data transfers. Table 1 lists the port operating modes and the PCIe functions in each mode.

Port Mode	PCIe Functions in the Port		
	Function 0	Function 1	Function 2
Disabled	The port is not operational		
Upstream switch port	PCI-to-PCI bridge		
Upstream switch port with NT function	PCI-to-PCI bridge	NT Endpoint	
Upstream switch port with NT and DMA functions	PCI-to-PCI bridge	NT Endpoint	DMA Endpoint
NT function	NT Endpoint		
NT with DMA function	NT Endpoint		DMA Endpoint
Downstream switch port	PCI-to-PCI bridge		
Unattached	The port is not associated with any partition. The port accepts configuration requests that allow for switch management.		

Table 1 Port Operating Modes[1]

[1] Not all switch ports support all modes. Refer to the PES32NT24G2 User Manual for further details.

Not all possible partition configurations are valid. For example, it is not valid to configure a partition with more than one upstream port. The following are the common partition configurations supported by the PES32NT24G2:

- A switch partition with one upstream switch port (with or without NT or DMA functions) and one or more downstream switch ports.
- A switch partition with an NT endpoint port (with or without DMA function).

Partitions and ports can be configured at boot-time, and reconfigured dynamically during run-time by software or automatically by hardware as a result of a failover event. When a port's mode is re-configured, the change can be stateless (i.e., the port is reset during the change) or state-full (i.e., the port preserves its configuration during the change).

Notes

Switch Management Interfaces

A switch manager device can connect to the PES32NT24G2 switch using one of two methods:

- Through the switch's SMBus slave interface.
- Via a switch's PCI Express port.

SMBus Slave Interface

When a switch manager device connects to the switch via the switch's SMBus slave interface, the manager acts as an SMBus master. Through this interface, the switch manager has access to all switch configuration and status registers by issuing switch-specific read/write commands. Each register has a unique address in the switch's *global address space* (described in a later section).

By accessing switch configuration and status registers, the switch manager can configure all aspects of the switch. In addition, mechanisms exist in the switch that allow communication between the switch manager and the PCIe roots connected to the switch. These mechanisms are described in the section entitled *Active Switch Management*.

A limitation of switch management via the SMBus slave interface is that the PES32NT24G2 can't proactively notify the switch manager of events. For example, if the switch automatically reconfigures its partitions due to a failover event, no notification is sent to the switch manager via the SMBus. Therefore, a switch manager connected via the SMBus slave interface must poll switch status registers in order to obtain information about switch events.

PCI Express Interface

Switch management via PCI Express is not subject to the restriction described in the previous section. When the switch manager connects to the switch via one of the switch's PCIe ports, the switch manager can configure all aspects of the switch and be notified by the switch (via PCIe interrupts) when important events occur. In addition, the switch manager can communicate with other PCIe roots connected to the switch.

The port to which the switch manager connects may be operating in one of many modes (e.g., upstream switch port, NT function, unattached, etc.) The operating mode of the port determines the tasks that the switch manager may perform through that port, as shown in Table 2.

Port Mode	Switch manager access to all switch registers	Capability to notify switch manager of switch events	Support for switch manager communication with other PCIe roots[1]
Upstream switch port	Yes	Yes	Yes
Downstream switch port	Switch management is not supported in this port mode.		
Upstream switch port with NT function	Yes	Yes	Yes
Upstream switch port with NT and DMA functions	Yes	Yes	Yes
NT function	Yes	Yes	Yes
NT with DMA function	Yes	Yes	Yes
Unattached	Yes	No	No
Disabled	Switch management is not supported in this port mode.		

Table 2 Switch Management Tasks for each Port Operating Mode

[1] Refer to section *Active Switch Management* later in this document.

Notes

Global Address Space

The PES32NT24G2 switch contains an internal global address space that is separate from the PCIe address spaces (i.e., PCIe configuration address space, I/O address space, and memory address space). Every configuration and status register in the switch has a unique address in the global address space. Therefore, access to the global address space allows for full management of the switch.

Figure 2 shows the layout of the PES32NT24G2 global address space. Configuration registers that are associated with a port's PCIe function (e.g., the PCI Command register in port 0's PCI-to-PCI bridge function) have an address in PCIe configuration space and a separate address in the switch's global address space.

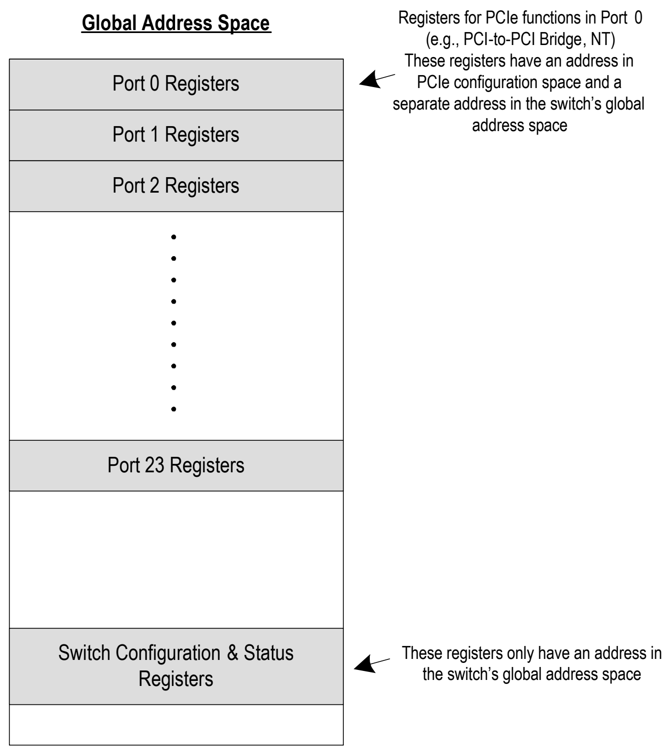

Global Address Space
Port 0 Registers
Port 1 Registers
Port 2 Registers
Port 23 Registers
Switch Configuration & Status Registers
Registers for PCIe functions in Port 0 (e.g., PCI-to-PCI Bridge, NT) These registers have an address in PCIe configuration space and a separate address in the switch's global address space
These registers only have an address in the switch's global address space

Figure 2 Layout of the Switch's Global Address Space

Some configuration registers are not associated with any port's PCIe function. These registers only have an address in the global address space. All such registers are located in a segment of the global address space known as the *Switch Configuration and Status Registers*. Registers in this segment control proprietary switch functionality and include the partition and port configuration registers that allow partitions to be configured as needed.

The global address space can be accessed through several methods. When the switch is initialized from a serial EEPROM (as described in the *Boot-Time Initialization* section later in this document), all commands in the EEPROM use global space addresses to access switch configuration registers. Similarly, when the switch is managed by a switch manager device connected to the switch's SMBus slave interface, the manager uses global space addresses to access switch configuration registers.

It is also possible to access the switch's global address space through PCIe. For example, a switch manager connected to a switch port can access the switch's global address space (and therefore any register within the switch) by accessing two special registers in the port's PCIe function(s). These special registers are known as "global address space indirection" registers, as they provide a gateway between PCIe domain and the switch's global address space[3].

Notes

Since access to the global address space allows full management of the switch, the PES32NT24G2 contains a global address space protection mechanism that allows the system designer to select which PCIe ports have access to the global address space. Typically, such access is only given to a port connected to a switch manager device.

Boot-Time Initialization

As described in the introduction, in some systems it is expected that the PES32NT24G2 switch will require some initial configuration prior to normal system operation. Boot-time initialization tasks include switch-partitioning, hot-plug configuration, GPIO configuration, SerDes transmitter/receiver adjustments, etc. This section describes boot-time initialization tasks and the methods by which this initialization is done.

Boot-time initialization occurs once during system operation, immediately after the switch's main reset signal (i.e., fundamental reset) is pulsed. Initialization may done via three non-exclusive methods: by hard-wiring the switch's boot-vector pins, by configuring the switch to load instructions from a serial EEPROM device, or by a switch manager device connected to the switch through one of the switch management interfaces (i.e., PCIe or SMBus).

Boot Vector Initialization

The PES32NT24G2 contains boot-vector pins that determine some aspects of the initial configuration of the switch. In particular, the boot-vector's "switch-mode" and "stack configuration" pins play a significant role in the initial configuration of the switch.

The stack configuration pins select the ports that the device will use during its operation. The PES32NT24G2 contains four "stacks", numbered 0 to 3. Stacks can be though-of as configurable port stations. Stacks 0 and 1 can each be configured with one x8 port, two x4 ports, four x2 ports, and combinations in between. Stacks 2 and 3 can be configured with one x8 port, two x4 ports, four x2 ports, eight x1 ports, and combinations in between. Each stack has dedicated stack configuration pins.

Figure 3 shows some of the configurations supported by each of the four stacks. Not all configurations are shown.

Stack 0			
Port 0	Port 1	Port 2	Port 3
x8	▨	▨	▨
x4	▨	x4	▨
x2	x2	x2	x2

Stack 1			
Port 4	Port 5	Port 6	Port 7
x8	▨	▨	▨
x4	▨	x4	▨
x2	x2	x2	x2

Stack 2							
Port 8	Port 9	Port 10	Port 11	Port 12	Port 13	Port 14	Port 15
x8	▨	▨	▨	▨	▨	▨	▨
x4	▨	▨	▨	x4	▨	▨	▨
x2	▨	x2	▨	x2	▨	x2	▨
x1	x1	x1	x1	x1	x1	x1	x1

Stack 3							
Port 16	Port 17	Port 18	Port 19	Port 20	Port 21	Port 22	Port 23
x8	▨	▨	▨	▨	▨	▨	▨
x4	▨	▨	▨	x4	▨	▨	▨
x2	▨	x2	▨	x2	▨	x2	▨
x1	x1	x1	x1	x1	x1	x1	x1

Figure 3 Stack Configurations

[3] Since the global address space indirection registers are located in the configuration space of each port PCIe function (e.g., PCI-to-PCI bridge, NT, or DMA), it is possible to access the global address space via PCIe configuration requests. Furthermore, the NT and DMA functions allow mapping of their configuration space into memory space using PCIe base address registers (BARs). Therefore, it is also possible to access the global address space via memory read/write accesses.

Notes

For example, stack 0 may be configured with four x2 ports (i.e., ports 0, 1, 2, and 3). Stack 1 may be configured with two x4 ports (i.e., ports 4 and 6). In this case ports 5 and 7 are not used since the stack is configured with only two ports. Stack 2 may be configured with eight x1 ports (i.e., ports 8 through 15). And stack 3 may be configured with one x8 port (i.e., port 16). In this case, ports 17 through 23 are not used since the stack is configured with one x8 port.

The stack configuration selected in a particular system depends on the number of ports desired, the lane-width of the ports, and the routing of PCIe signals on the system board. The switch's User Manual contains further information on possible stack configurations.

In addition to the stack configuration pins, the boot vector contains switch-mode pins that select the initial partition configuration, the initial operating mode of the ports (e.g., upstream, downstream, disabled, etc.), and whether the switch reads instructions from a serial EEPROM.

Using the switch mode pins, the switch can start in one of several modes. The most important of these are listed in Table 3. Refer to the PES32NT24G2 User Manual for a complete list of switch modes.

Switch Mode	Name
0	Single Partition Mode
1	Single Partition Mode with Serial EEPROM Initialization
10	Multi-Partition Mode with Unattached Ports
12	Multi-Partition Mode with Unattached Ports and Serial EERPOM Initialization

Table 3 Basic Switch Modes for the PES32NT24G2

In Single Partition Mode, the switch starts as a basic PCI Express switch. Only one partition is active (i.e., partition 0) and all other partitions are disabled. Port 0 is the upstream port of the active partition, and all other ports are downstream ports. Port 0 is configured as a basic switch upstream port (i.e., PCI-to-PCI bridge function) without NTB and DMA functions. In this mode, no other configuration of the switch partitions is required in order for PCIe software to enumerate the switch[4].

In Multi-Partition Mode with Unattached ports, the switch starts with all its partitions in the disabled state and all ports in the unattached mode. In this mode, the switch's partitions must be configured before the PCIe root(s) can start the enumeration process. The configuration can be done via serial EEPROM or by an active switch manager device, as described later in this document.

Except for the serial EEPROM initialization procedure, switch modes with serial EEPROM initialization are identical to the corresponding modes without it. For example, switch modes 0 and 1 are identical, except that switch mode 1 causes the switch to load from a serial EEPROM (see next section). Note that regardless of the initial switch configuration chosen by the switch-mode pins, the switch can be dynamically re-configured during system operation.

Initialization from Serial EEPROM

The PES32NT24G2 switch supports initialization by reading from a pre-loaded serial EEPROM device. Serial EEPROM initialization is expected to be widely used, as it provides a simple mechanism to pre-configure switch functionality prior to PCIe root enumeration. For example, the serial EEPROM can assign ports to partitions, configure port modes, configure port clocking modes, configure I/O expander addresses for hot-plug, configure GPIO pin directions, etc. The Boot-time Initialization Tasks section in this document provides more examples.

Serial EEPROM instructions use global addresses and therefore the serial EEPROM can configure all aspects of the switch.

The serial EEPROM device connects to the PES32NT24G2 via the switch's SMBus master interface. Immediately after the fundamental reset signal is de-asserted, the switch's SMBus master logic reads and executes instructions pre-loaded in the serial EEPROM. While this process takes place, all switch ports enter a mode in which they reject any received PCIe configuration requests by responding with a PCIe

[4]. This sentence specifically refers to switch partition configuration. Other switch configuration (e.g., I/O expander address initialization) may still be required in order for the switch to be ready for system operation.

Notes

configuration-request-retry completion. In this way, the PCIe enumeration process does not interfere with the EEPROM loading process. Once the serial EEPROM loading completes[5], the switch ports enter a normal operation mode in which they accept and process PCIe configuration requests.

The PES32NT24G2 supports reading from serial EEPROM devices that store as much as 64 KB of instructions. Instructions in the serial EEPROM are IDT-proprietary commands that modify configuration registers internal to the switch. The serial EEPROM can only write to registers in the switch; it can't read register values from the switch. Table 4 lists the serial EEPROM command types supported by the switch. For further details on this refer to the PES32NT24G2 User Manual.

Instruction Type	Description
Single Double-Word Initialization Sequence	Writes to a single double word register at a specified address in the switch.
Sequential Double-Word Initialization Sequence	Sequentially writes to double word registers starting from a specified address in the switch.
Jump	Jumps to another instruction in the serial EEPROM.
Wait	Waits for the switch to complete an action (e.g., configure a partition) before executing the next instruction. This may be used as a synchronization barrier.
Configuration Done	Indicates completion of the instruction sequence.

Table 4 Serial EEPROM Instructions supported by the PES32NT24G2

Initialization by a Switch Manager

The PES32NT24G2 may also be initialized by a switch manager device connected to the switch via one of the switch management interfaces (i.e., SMBus slave interface or PCIe). Unlike serial EEPROM initialization, a switch manager can read or write any register in the switch; therefore, this method is the most flexible for boot-time initialization.

Boot-time initialization by a switch manager occurs prior to the switch entering normal system operation (i.e., before the PCIe root(s) enumerate the switch). This type of initialization is only supported when the PES32NT24G2 boots in multi-partition mode as selected by the boot-vector (e.g., switch modes 10 to 15). In multi-partition mode, the switch ports enter a mode in which they temporarily reject PCIe root accesses while the switch manager configures the switch. In this way, the switch manager can configure the switch before the PCIe roots are given access to enumerate the switch. The *Switch Partition Configuration* section later in this document describes this process in detail.

In scenarios where the switch is initialized by both a serial EEPROM and a switch manager, the switch manager should wait for serial EEPROM loading to complete in order to avoid conflicts. The switch manager can make this determination by reading status registers within the switch.

Boot-time Initialization Tasks

This section describes common boot-time initialization tasks. Below is the list of tasks, and each of these is briefly described in the following sections. The list is not exhaustive, and tasks other than the ones described here are possible. The tasks are:

- Switch Partition configuration
- Port Clocking Mode selection
- Hot-Plug initialization
- GPIO initialization
- NTB initialization
- DMA initialization
- Failover initialization

[5] In order to meet the PCIe reset requirements, the serial EEPROM loading must complete in less than 1 second. This amount of time is enough to configure all aspects of the switch.

Notes

- Internal Error Reporting initialization
- SerDes Transmitter/Receiver Tuning
- Low Swing Mode enabling

Switch Partition Configuration

Switch partition configuration consists of configuring the switch partitions by placing each partition in the desired mode (i.e., disabled, active, reset), allocating ports to partitions, and configuring the operating mode of the ports (e.g., upstream switch port, NT function, etc.)

Switch partitioning is required when the PES32NT24G2 boots in multi-partition mode with unattached ports (i.e., switch modes 10 and 12). Switch partitioning is not required when the PES32NT24G2 boots in the single partition modes (i.e., switch modes 0 or 1), because in these modes the switch starts with a valid partition configuration (see section *Boot Vector Initialization*).

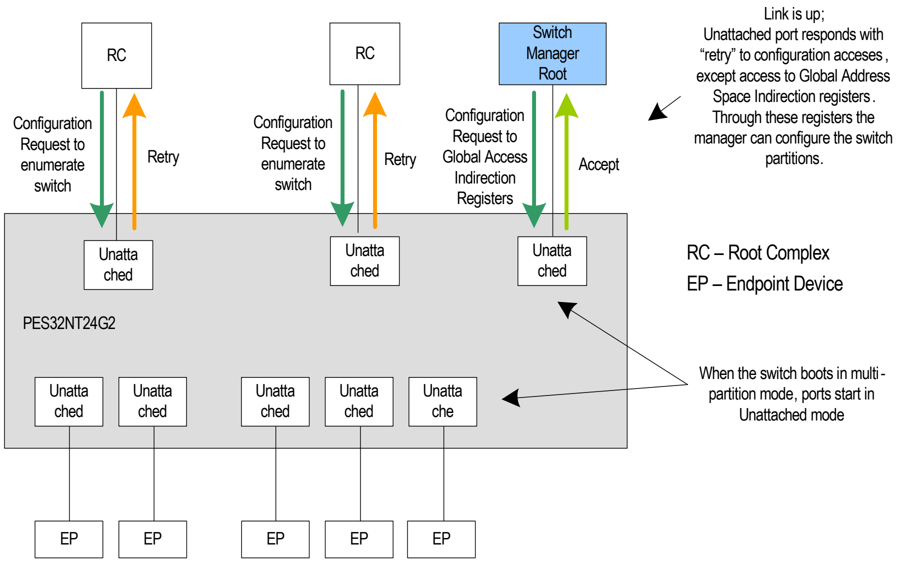
When the PES32NT24G2 boots in multi-partition mode, none of the switch partitions are configured. Therefore, these modes require that partitions be configured by the serial EEPROM or by a switch manager prior to normal system operation. Depending on the system's configuration, a switch port may be connected to a PCIe root or an endpoint on the system board. It is expected that ports that connect to PCIe roots will be configured as upstream ports in a switch partition (e.g., the port's mode is set to upstream switch port, NT function, etc.) Similarly, it is expected that switch ports that connect to endpoints will be configured as downstream ports in a partition. If a port's connection is not known during boot-time initialization, the port mode can be selected once the configuration is known since the switch supports dynamic partition reconfiguration. Finally, switch ports that are not connected on the system board may be disabled.

When the switch boots in multi-partition mode with unattached ports, all ports start in unattached mode. In this mode, each port trains the PCIe link with its link-partner[6]. Once the link is up, the port responds to received PCIe configuration requests with a 'retry' completion, except for configuration requests that target the global address indirection registers in the port (refer to section *Global Address Space*). The idea is that switch ports that connect to regular PCIe roots (i.e., roots that are not the switch manager root) respond with a 'retry' completion when the roots start PCIe enumeration, thereby preventing those roots from enumerating the switch while the switch partitions are configured. The switch partitions can be configured via the serial EEPROM or by a switch manager. If a switch manager is connected to the switch via a PCIe port, the switch manager can configure the switch by accessing the port's global address indirection registers. It is expected that the switch manager will have a customized BIOS to perform this switch configuration, and that the switch configuration will be completed in less than 1 second from the time the switch starts booting in order to meet PCIe specification reset requirements.

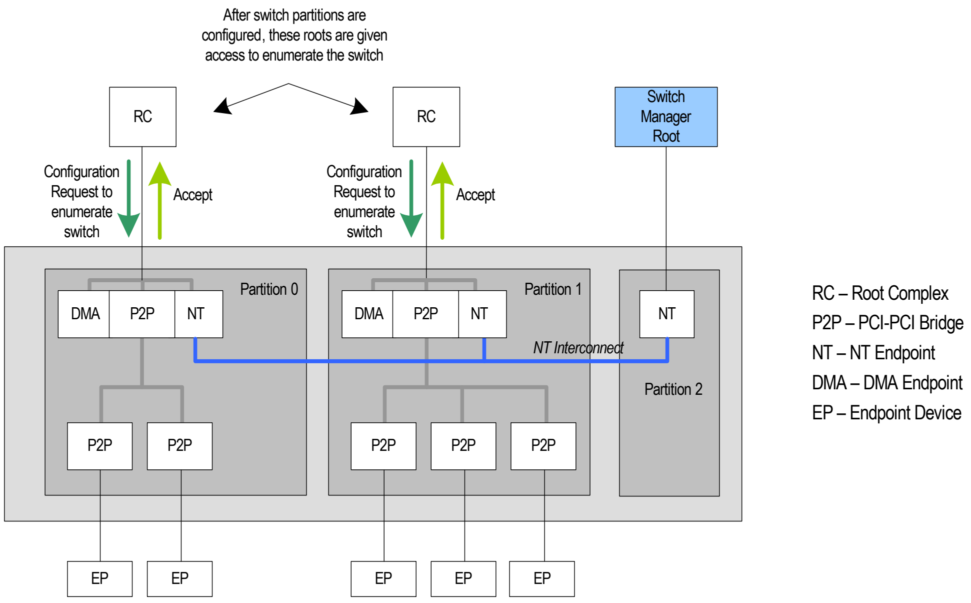
Figure 4 shows an example of a PES32NT24G2 which is configured with multiple partitions by a switch manager device connected to one of the switch's ports. Once a partition is configured, the ports in the partition start processing requests normally and the PCIe root associated with that partition's upstream port gets access to the switch.

[6]. Unattached ports train naturally with PCIe root ports or another switch's downstream ports. Otherwise, the unattached port tries to form a PCIe crosslink with its link partner. Crosslinking only succeeds when both devices in the link support it.

Notes


RC
Configuration Request to enumerate switch
Retry
RC
Configuration Request to enumerate switch
Retry
Switch Manager Root
Configuration Request to Global Access Indirection Registers
Accept
Link is up; Unattached port responds with "retry" to configuration acceses, except access to Global Address Space Indirection registers. Through these registers the manager can configure the switch partitions.
Unattached
Unattached
Unattached
PES32NT24G2
RC – Root Complex
EP – Endpoint Device
Unattached
Unattached
Unattached
Unattached
Unattache
When the switch boots in multi-partition mode, ports start in Unattached mode
EP
EP
EP
EP
EP

Before Switch Partitions are Configured

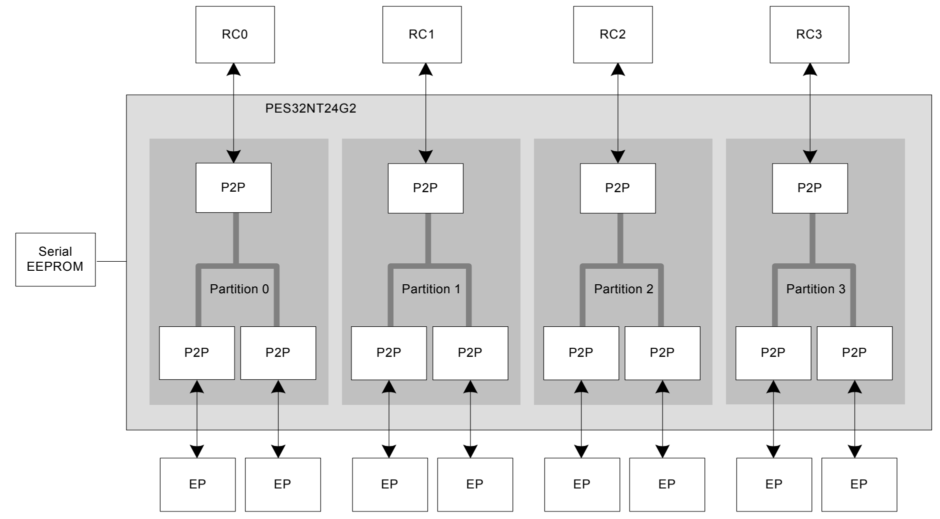


After Switch Partitions are Configured

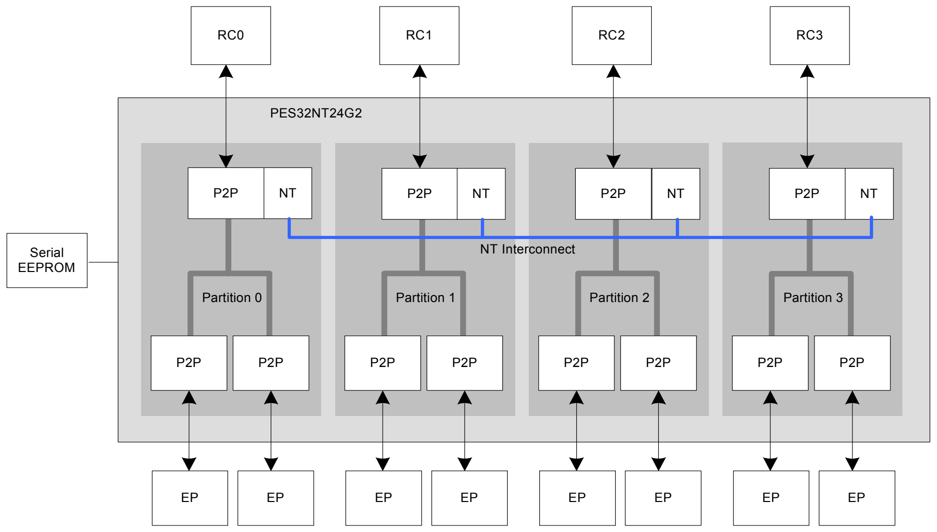
Figure 4 Example of the Switch Configuration process in Multi-Partition Mode

Switch partitions can be configured in multiple ways. Figure 5 shows examples of common partition configurations. In the figure, the partitions are assumed to be configured by a serial EEPROM device, but they could be configured by a switch manager device as described before. Note that partition configurations other than the ones shown are possible.

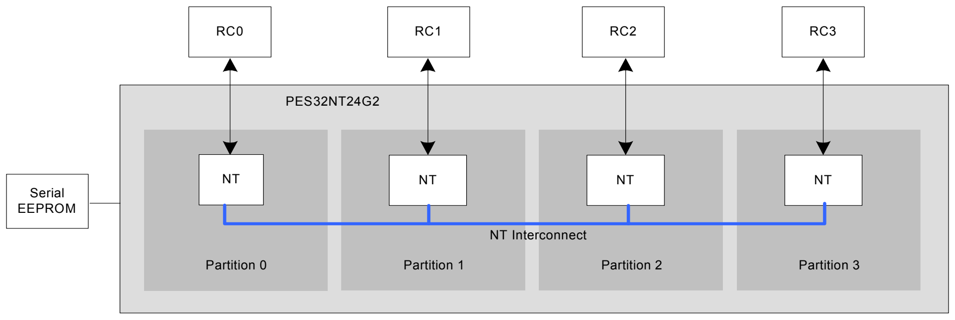
Notes



Basic Multi-Partitioned Switch Configuration



Multi-Partition Switch Configuration with Non-Transparent (NT) Interconnection



PES32NT24G2 as a Multi-Processor System Interconnect

Figure 5 Examples of Switch Partition Configurations in the PES32NT24G2 Switch

Notes

Port Clocking Mode Selection

Ports in the PES32NT24G2 support two clocking modes: global clocked or local port clocked. These modes are described in detail in the User Manual and their usage is discussed in Application Note AN-715.

The clocking mode for a port is typically selected during boot-time initialization and not changed during system operation. By default, all ports start in global clocked mode. In order to modify a port's clocking mode, the serial EEPROM or switch manager can write to the PCLKMODE register in the switch's global address space. Refer to the User Manual for a list of requirements that must be met when modifying a port's clocking mode.

Hot-Plug Initialization

All ports in the PES32NT24G2 contain a hot-plug controller, but this controller is only active on switch ports configured as downstream ports. Hot-plug signaling is done via the use of external I/O expanders that connect to the switch's SMBus master interface.

Hot-plug initialization is required when a switch downstream port is connected to a slot. Hot-plug initialization typically consists of the following tasks. These are accomplished by writing to proprietary registers in the switch.

- Programming in the switch the SMBus addresses of the I/O expanders.
- Selecting the polarity of the hot-plug signals.
- Selecting the manner in which the hot-plug controller detects the presence of a card in the slot.
- Configuring the hot-plug reset output control.

The tasks listed above are done by the serial EEPROM or a switch manager during boot-time initialization, before the system's PCIe roots enumerate the switch and configure the downstream port hot-plug mechanism via the PCI Express standard registers. Refer to the switch's User Manual for further details.

GPIO Initialization

The PES32NT24G2 contains 9 general-purpose IO (GPIO) pins. Each pin may be configured as an input or output pin. A register in the switch allows software to write a value to be driven on the pin (i.e., GPIO output pin) or to read a value sampled from the pin (i.e., GPIO input pin). In addition, all GPIO pins support "alternate functionality", in which the pin is associated with a specific switch function (e.g., the GPIO[8] pin acts as the I/O expander interrupt input).

In systems in which the switch GPIO pins are used, GPIO initialization is required. From the switch's perspective, GPIO initialization is very simple and consists of just programming proprietary registers in the switch to select the mode of each GPIO pin (i.e., input, output, alternate). Refer to the switch's User Manual for further details on GPIO pins and their alternate functionality. Note that when hot-plug is enabled, the GPIO[8] pin must be configured in alternate functionality in order for the switch to detect hot-plug I/O expander interrupts.

Non-Transparent Bridge (NTB) Initialization

The PES32NT24G2 supports NTB to allow inter-domain communication across PCIe domains (i.e., switch partitions). In order for any two partitions to communicate across the NTB, the upstream port in each partition must be configured with an NT function (see section *Switch Partitioning* on page 1).

Prior to communicating across the NTB, several aspects of the NTB mechanism must be configured. Some of this configuration is expected to be done during boot-time initialization, while the rest is expected to be done by software (i.e., the NT function's driver).

Notes

Table 5 shows a list of NTB initialization tasks expected to be done during boot-time initialization.

Task	Description	Requirement
Base Address Register (BAR) Setup	Select the size and type of the memory range requested by each BAR in the NT function.[1]	Required
NT Doorbell Routing	Mapping doorbell bits for switch partition notification.	Optional
NT Message Routing	Mapping of message outbound to inbound registers across switch partitions.	Optional
NT Mapping Table Virtualization	The NT mapping table is a table shared across all switch partitions. To prevent conflicts when roots access this table, the switch supports virtualization of the table. This allows the division of the NT mapping table into segments and the assignment of each segment to a switch partition.	Optional
NT Mapping Table Protection	Protects against NT mapping table mis-configuration.	Optional

Table 5 NTB Initialization Tasks

[1] The actual allocation of memory space to the NT function is done when the BIOS and/or Operating System executing in the root-complex programs the NT function's BARs.

DMA Initialization

The PES32NT24G2 contains two DMA engines located in ports 0 and 8. A DMA engine is logically visible to PCI Express only when the port is configured in a mode with a DMA function (see section *Switch Partitioning* on page 1).

Prior to using the DMA, a several aspects of the DMA need to be configured. One of these is expected to be done during boot-time initialization, while the rest is expected to be done by software (i.e., the DMA driver). Specifically, it is expected that the DMA's BAR setup will be done during boot-time initialization. DMA BAR setup is done in order to map the DMA function configuration registers into the system's memory space associated with DMA BAR 0. DMA BAR setup is done by writing to a proprietary switch register. Refer to the switch's User Manual for further details.

Failover Initialization

The PES32NT24G2 supports automatic hardware-based failover. This feature allows the switch to automatically reconfigure its partitions upon detection of a trigger. The selection of the trigger, as well as the way in which the switch reconfigures itself is programmable. Application Note AN-716 describes failover usage models.

Failover may be programmed during boot-time initialization such that the failover mechanism is armed early during system operation. In systems where failover is required but the PES32NT24G2 is not connected to a switch manager, it is expected that the serial EEPROM will initialize the failover mechanism. In systems where a switch manager is present, this manager can initialize the failover mechanism as well as be notified by the switch when a failover occurs. Such notification is done using the Event Signaling mechanism described later in this document.

Initializing the failover mechanism consists of the following actions:

- Selecting the failover trigger (i.e., signal pin, watchdog timer, or software trigger).
- Selecting the action taken by the partitions and ports when a failover is triggered.
- Enabling the failover mechanism

The failover initialization actions are done by programming proprietary registers in the switch. Refer to the User Manual for details on this.

The failover mechanism supports "primary" and "secondary" failover, which allows alternating actions to be taken depending on the number of times the failover triggers. Typically, the primary action undoes changes caused by a secondary action, and vice-versa. Therefore, once initialized, the failover mechanism does not require active management by a switch manager device.

Notes

Internal Error Reporting Initialization

The PES32NT24G2 supports reporting of internal errors via the Advanced Error Reporting (AER) mechanism defined by the PCIe 2.1 specification. The PES32NT24G2 is capable of detecting several types of internal errors, such as internal buffer memory errors, parity errors across the packet's internal data-path, and packet time-outs due to external link congestion for very long periods of time. Registers in each switch port allow control of which internal error types are reported by the port and their severity rating (e.g., correctable or uncorrectable). In an effort to inter-operate with software which does not support internal error reporting in AER, it is also possible to disable the internal error reporting mechanism on a per-port basis.

By default, internal error reporting is disabled. For systems in which internal error reporting by the switch is desired, the mechanism needs to be initialized before the operating system in each PCIe root configures the port AER registers. Initializing this mechanism consists of programming proprietary registers in the switch to enable internal error reporting, selecting the errors types to be reported, and assigning them a severity. The switch offers per-error granularity in this control. Once the internal error reporting mechanism is initialized, the operating system in each PCIe root needs to unmask the reporting of internal errors in the AER registers. Refer to the switch's User Manual for further details on internal error reporting.

SerDes Transmitter/Receiver Tuning

The PES32NT24G2 contains proprietary registers that allow control of the switch's SerDes transmitter and receiver circuitry on a per-port and per-lane basis. Such controls include transmit driver level control, fine de-emphasis adjustment, and receiver equalization controls.

The control values selected for a particular SerDes depend heavily on the channel's characteristics on the system board. Systems with long channels may require setting of the SerDes controls in order to improve link reliability. This setting is typically done very early in the boot process, before any packet traffic is transmitted across the PCIe links (e.g., by instructions in the serial EEPROM that set the SerDes controls according to values determined a-priori during system characterization.) Since the link may become unstable when SerDes controls are modified, it is recommended that the port's link be retrained after its SerDes controls are modified. Link retraining is accomplished by writing to the link retrain bit in the port's link control register. Refer to the switch's User Manual for further details on SerDes controls.

Low Swing Mode Enabling

The PES32NT24G2 supports the low-swing mode specified in the PCIe 2.1 specification. In this mode, the differential voltage swing is reduced to half of its default value, thereby reducing power consumption in the switch. The switch has proprietary registers that enable low-swing mode on a per-port basis.

Low swing mode is typically enabled during boot-time initialization (e.g., by instructions in the serial EEPROM). Since the link may become unstable when transitioning from full-swing to low-swing mode, it is recommended that the port's link be retrained after this mode is enabled. Refer to the switch's User Manual for further details on low-swing mode.

Notes

Active Switch Management

The previous sections described boot-time initialization tasks. In addition to being initialized, the PES32NT24G2 switch can be actively managed by a switch manager device during system operation. In this case, the switch manager monitors the switch's status and applies changes to the switch configuration, sometimes coordinating these changes with PCIe roots attached to the switch's partitions.

The need for active switch management in a system depends on the complexity of the switch capabilities being used. Switch management tasks include but are not limited to switch partition reconfiguration (e.g., migration of ports among partitions) and event signaling management. This section briefly describes these tasks.

As described in the section *Switch Management Interfaces*, a switch manager device connects to the PES32NT24G2 switch through the switch's SMBus slave interface or via a PCIe port.

Communication between a Switch Manager and System PCIe Roots

Before describing common switch management tasks, the topic of communication between a switch manager and regular (i.e., non-manager) PCIe roots is introduced. For further details on this, refer to the switch's User Manual and Application Note AN-722.

The PES32NT24G2 contains mechanisms that allow a switch manager device to communicate with regular PCIe roots. In this way, the switch manager can notify the PCIe roots when a significant event has occurred (e.g., failover) as well as coordinate switch reconfiguration actions with the PCIe roots (e.g., migration of ports between partitions.)

Communication between the system's PCIe roots and the switch manager is best done when the switch manager is itself a PCIe root connected to a switch's port. In this way, the PES32NT24G2 can notify the switch manager of a received message by issuing interrupts[7]. Furthermore, if the PCIe port to which the switch manager connects has an NT function, the switch manager can use the capabilities of the non-transparent bridge (e.g., NT data transfers, doorbells, message registers) to communicate with other PCIe roots. This option provides the best communication capabilities.

Since the number of NTB ports is limited to 8, the switch manager may not be connected to such a port. For these cases, communication between the PCIe roots attached to the switch and the switch manager is still possible via the proprietary Global and Switch Signals mechanisms.

'Global signals' allow PCIe roots to send an interrupt-driven notification to the switch manager root[8]. Conversely, 'switch signals' allow the switch manager to send an interrupt-driven notification to one or more PCIe roots. In both cases, the sender can write to a proprietary register in the switch to store a 32-bit message to be picked-up by the receiver. This register is accessible by any PCIe root or by the switch manager via the switch's global address space; the PES32NT24G2 places no constraints on the data transferred via this register.

Switch Management Tasks

This section describes common switch management tasks for the PES32NT24G2 switch. These tasks include:

- Dynamic Partition reconfiguration
- Event Signaling management

[7]. When the switch manager connects to the PES32NT24G2 switch via the SMBus slave interface, the switch is not capable of proactively notifying the switch manager of events. Instead, the switch manager must poll switch status information.
[8]. The 'global signals' mechanism requires prior configuration of the event signaling mechanism described later in this document.

Notes

Dynamic Partition Reconfiguration

The PES32NT24G2 supports dynamic reconfiguration of switch partitions (i.e., after boot-time initialization). In this way, the resources assigned to the switch partitions can be modified at run-time, allowing for flexible usage models such as dynamic I/O resource allocation across PCIe domains and support for high-availability configurations. The following are common partition reconfigurations:

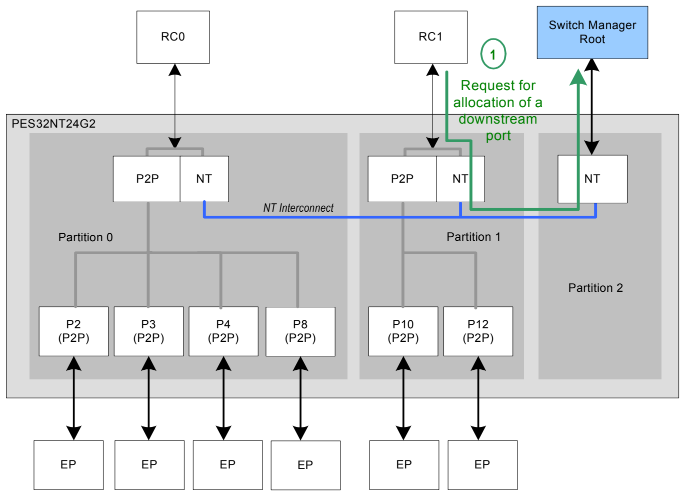
- A downstream port is added or removed from a partition (e.g., port migration between partitions)
- The operating mode of an upstream port is modified (e.g., upstream switch port becomes an NT port)

Switch partition reconfiguration can be done automatically by the hardware as a result of a failover event, or by software writing to registers in the switch. For the latter case, it is expected that a switch manager device connected to the switch's SMBus slave interface or to a PCIe port would perform the partition reconfiguration.

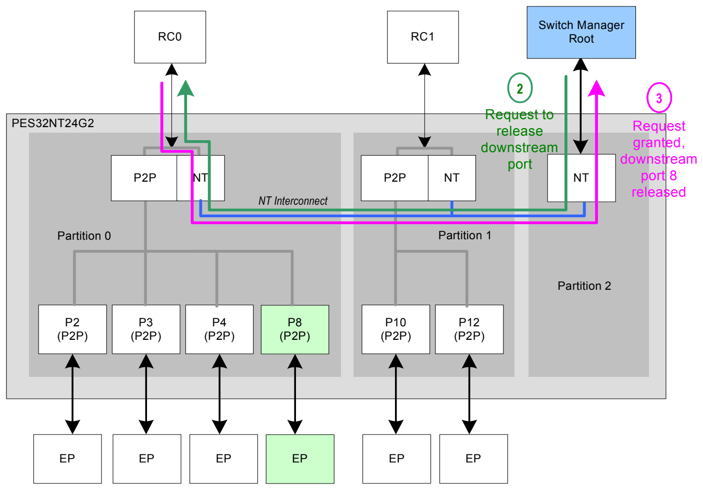
Reconfiguring the switch partitions may require coordination between the switch manager device and the PCIe roots attached to the switch partitions. For example, a PCIe root connected to a switch partition with one upstream port and two downstream ports may send a request to the switch manager asking for another downstream port (e.g., a downstream port connected to an I/O device such as a network interface card or storage controller). The switch manager, assumed to have knowledge of the switch partition configuration, could send a message[9] to other PCIe roots asking for a downstream port. Assume that one of these PCIe roots has low utilization on an I/O device and decides to grant the request. This root would quiesce any traffic to that I/O device, reset the device, and remove it from its table of I/O resources. This root would then send a message to the switch manager indicating that the request is granted and indicating which downstream port has been quiesced. The switch manager would proceed to reconfigure the switch partitions such that the downstream port is migrated to the switch partition associated with the PCIe root that requested the resource. Once the change is completed, the switch manager would send a message to the PCIe root that requested the resource. This root would then proceed to re-enumerate the PCIe hierarchy, allocate system resources to the new downstream port and the I/O device to which it connects, and proceed to run traffic through this I/O device. Figure 6 shows this process.

[9]. The message is sent using the communication mechanisms described before.

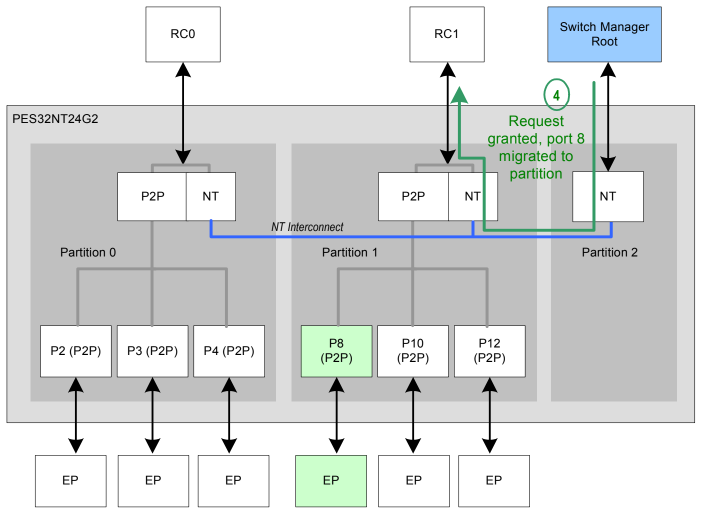
Notes



Downstream Port Migration (1)



Downstream Port Migration (2)



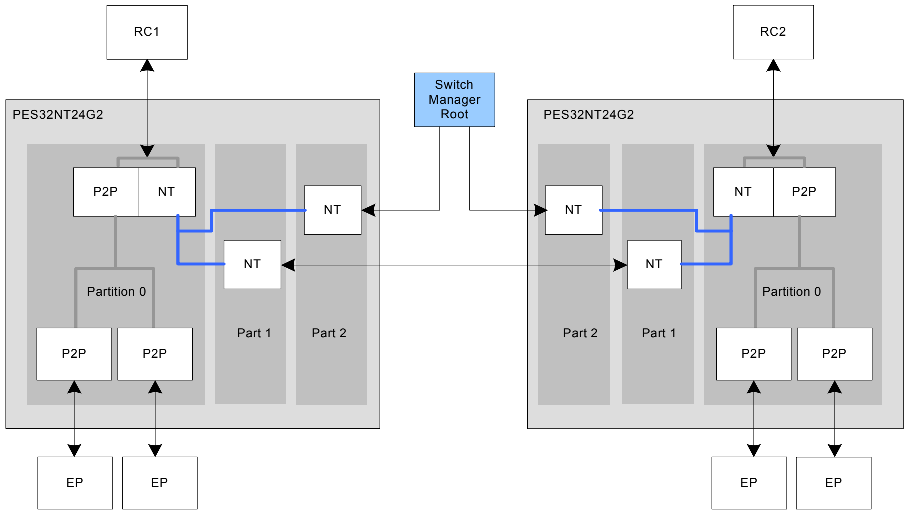
Downstream Port Migration (3)

Figure 6 Example of Dynamic Partition Reconfiguration (Downstream Port Migration)

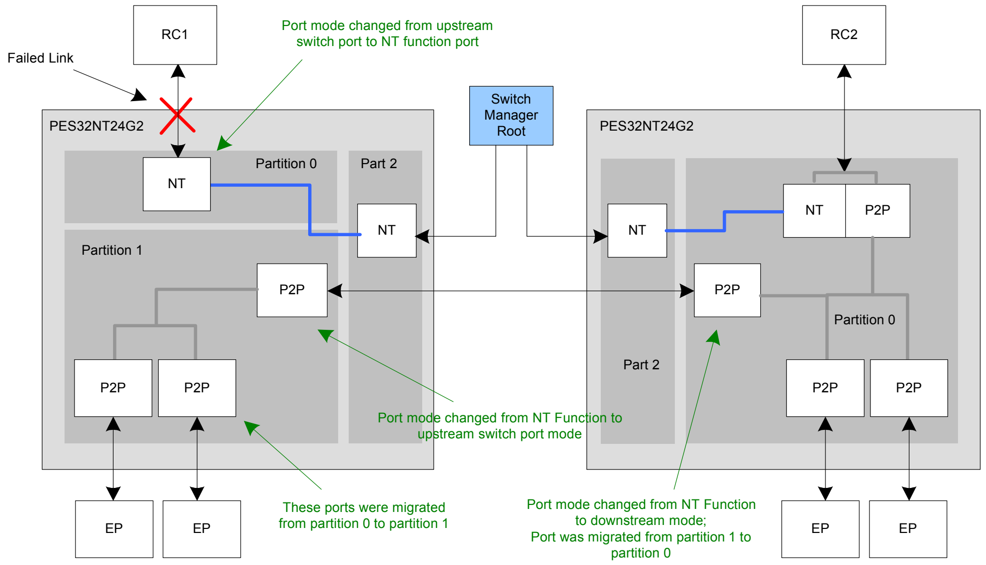
Notes

Another form of partition reconfiguration is when a switch manager modifies the operating mode in a partition's upstream port. This is done in switch configurations that target highly-available systems. Figure 7 shows a scenario where two PES32NT24G2 switches are used in a system to provide high-availability in case of switch failure or PCIe root failure. The figure shows the system before and after a failure is detected on one of the switch upstream port links. The switch manager coordinates the transition by communicating with the PCIe root that remains active and executing the partition reconfiguration. As shown in the figure, the switches have been reconfigured such that the root-complex that has not failed (i.e., RC2) can now access the PCIe endpoints that were previously under the other root's PCIe domain.

Refer to the PES32NT24G2 User Manual for further details on switch partition reconfiguration.



System Configuration before Failure


RC1
Failed Link
Port mode changed from upstream switch port to NT function port
RC2
Switch Manager Root
PES32NT24G2
PES32NT24G2
NT
Partition 0
Part 2
NT
Partition 1
P2P
P2P
P2P
NT
NT
P2P
P2P
Partition 0
Part 2
P2P
P2P
Port mode changed from NT Function to upstream switch port mode
EP
EP
These ports were migrated from partition 0 to partition 1
Port mode changed from NT Function to downstream mode; Port was migrated from partition 1 to partition 0
EP
EP

System Configuration after Switch Manager Re-configures Partitions

Figure 7 Highly Available System Configuration using two PES32NT24G2 Switches

Notes

Event Signaling Management

The PES32NT24G2 supports an "event signaling" mechanism that allows the switch to monitor hardware events and notify the PCIe root(s) connected to it. Typically, this notification is sent to a switch manager device connected to the switch via a PCIe port, such that the manager can take appropriate action. The event signaling mechanism is described in detail in the User Manual and in Application Note AN-722. This section will not focus on the details of event signaling, but rather on the management of this mechanism.

The use of switch event signaling in a system is optional, but this mechanism is valuable when the switch is configured with multiple partitions and a switch manager device wants to monitor anomalous behavior in one or more of the switch partitions. For example, if a switch partition is reset, the event signaling mechanism can automatically send an interrupt-driven notification to the switch manager[10]. The switch manager can then decide the appropriate action, which could include migrating downstream ports from the partition under reset to the active partitions, and/or modifying the operating mode of the upstream port in the partition under reset (e.g., from an upstream switch port to an NT function port).

The PES32NT24G2 supports monitoring and signaling of several events. These include partition resets, port links going down or up, failover events, and port error detection. The switch's User Manual contains a detailed list.

The event signaling mechanism requires active management by a switch manager device. Management of the event signaling mechanism involves the following tasks:

- Selecting the events to be monitored
- Selecting the switch partition to be notified of the event
  - This is the partition that contains the PCIe port to which the switch manager connects
- Enabling interrupt generation due to event signaling

When an event occurs and the switch generates an interrupt to notify the switch manager, the switch manager can access registers in the switch to determine the type of event that occurred and the port or partition on which it occurred. The switch manager can then take appropriate action (e.g., partition reconfiguration) and proceed to re-arm the event signaling mechanism.

Conclusion

The rich set of features in the PES32NT24G2 switch (e.g., partitioning, NTB, DMA, failover, etc.) and its highly flexible configurability, allow the use of the switch in a wide variety of system configurations, from basic PCI Express switching to multi-processor system interconnection and highly-available system configurations. In some systems, it is expected that the switch will have to be initialized at boot-time to enable some of its features before normal system operation. In other systems, it is expected that the switch will be actively managed by a dedicated device during system operation.

This application note introduced the mechanisms by which the PES32NT24G2 may be initialized during boot-time and actively managed thereafter. This application note also identified several common initialization and management tasks, and provided a brief description of each. For further details on the features in the PES32NT24G2 and the manner in which these are configured, refer to the switch's User Manual.

References

[1] IDT 89PES32NT24G2 PCI Express® Switch User Manual.

[2] PCI Express® Base Specification, Revision 2.1, March 4, 2009, PCI-SIG.

[3] Application Note AN-715, Clocking Architecture in IDT's PCIe® Gen2 System Interconnect Switch Family, IDT, August 2009.

[4] Application Note AN-716, Building Failover Systems with IDT's PES32NT24G2 PCIe® Switch, IDT, August 2009.

10. Assuming the switch manager is connected to a PCIe port. Interrupt-driven notification is not supported for a switch manager connected to the switch's SMBus slave interface.

Notes

[5] Application Note AN-722, Inter-Domain Communication in the PES32NT24G2 PCIe® Switch, IDT, September 2009.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks