Building Failover Systems with IDT's PES32NT24G2 PCIe® Switch

Application Note AN-716

Notes

By Alex Chang

Introduction

The IDT PES32NT24AG2 PCIe® switch supports a flexible failover mechanism that provides vital functions required for the construction of fault tolerant systems. The failover mechanism consists of the following features:

- Dynamic switch partitioning
- Runtime re-configurable upstream/NTB ports
- Failover Capability structures with automatic reconfiguration

The switch allows up to eight active switch partitions and each switch partition represents an independent PCIe hierarchy whose operation is independent of other switch partitions. The dynamic switch partitioning supports two forms of dynamic reconfiguration: Each switch port may be dynamically assigned to partitions and the operating mode of a port may be dynamically re-configured without affecting the unrelated switch partitions.

The switch supports up to eight NTB (Non-Transparent Bridge) endpoint functions that can serve as built-in communication channels for system interconnecting to exchange information among different roots via the inter-domain communication facilities. The NTB endpoint functions can be dynamically enabled and configured to associate with Ports 0, 2, 4, 6, 8, 12, 16, and 20 when these ports are configured to operate in one of the following modes:

- Upstream switch port with NTB function
- NTB function

The PES32NT24AG2 switch provides four Failover Capability structures that can be selected in conjunction with individual partition/port configuration related to failover operations, which features automatic and dynamic reconfiguration for the involved switch partitions and ports upon detection of a predefined trigger from the selected Failover Capability structure.

There are three failover usage models provided to demonstrate how the failover mechanism can be used to construct applications. Followed by more details in terms of configuring Failover Capability structures, different failover initiations, failover switch events and interrupts, etc. At the end of this document, an example of using the Failover Capability structure is provided with explicit procedures, which represents a complete reference as users adopt the failover mechanism in their applications.

Failover Usage Models

This section provides three failover usage models that users may use as references to build their own usage model for their application, taking advantage of the flexibility of the failover mechanism. With the supported switch event signaling scheme and inter-processor communication via NTB, the following failover events are identified as signals to initiate failover:

- A link failure between the upstream switch port and its corresponding root
- The monitoring heart beat had not been received for a designed period
- User-initiated failover

N+1 Protection

Figure 1 illustrates the topology of an N+1 protection model within a PES32NT24AG2 switch before the use of dynamic switch partitioning. In such a system, a single standby root complex protects N active root complexes. Each root connects to a switch partition. If an identified failover event happens to one of the

Notes

active root complexes, the standby root complex will take over as the root complex. In this example, N is three. However, this usage model can be extended to seven switch partitions connecting to active root complexes and one to a standby root complex.

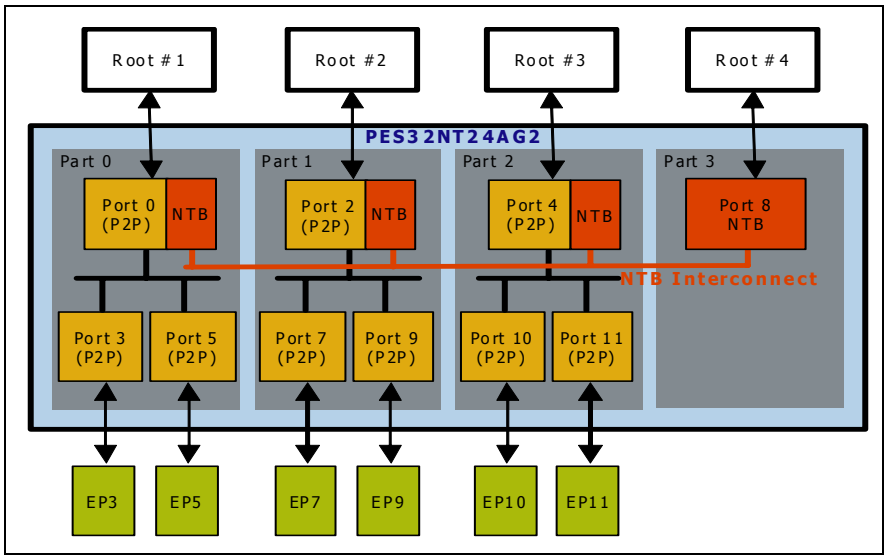

Figure 1 Dynamic Switch Partitioning (Before)

Assuming that the monitoring system had not received the heart beat message from Root #3 for a designated period, and Root #4 decides to take over the associated downstream switch ports (Ports 10 and 11) and the I/O endpoints (EPs 10 and 11) in the following steps:

- Dynamically configure Port 4 from an upstream switch port with NTB function to an NTB function
- Dynamically configure Ports 10 and 11 and move them to Partition 3
- Dynamically configure Port 8 from NTB function to an upstream switch port with NTB function
- The system software running in Root #4 re-scans and re-enumerates the PCI buses to discover the new topology, including EPs 10 and 11
- Root #3 becomes the standby root complex and Root #4 becomes one of the active root complexes as shown in Figure 2.

The system software running in Root #3 needs to re-scan and re-enumerate the PCI buses due to the changes of Port 4 and its removed downstream devices after a power cycle on Root #3.

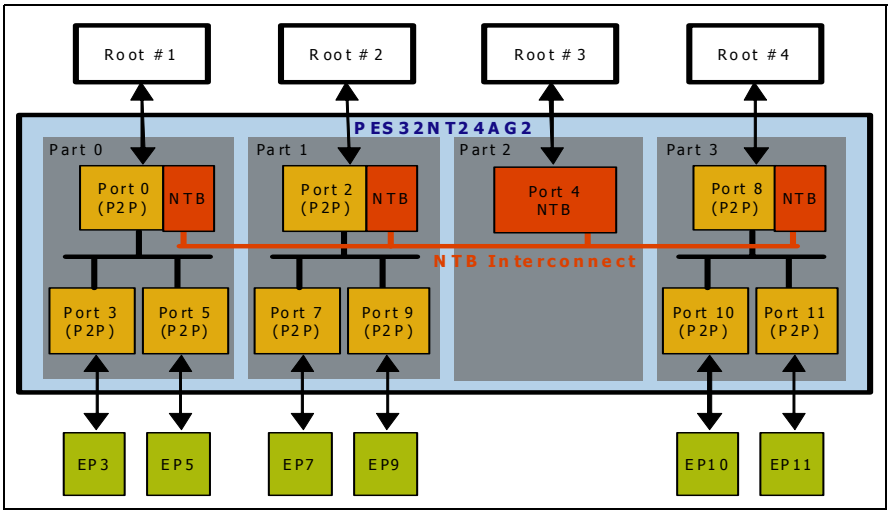

Figure 2 Dynamic Switch Partitioning (After)

Notes

Dynamic Partitioning with NTB

In this usage model, two PES32NT24AG2 switches are symmetrically connected together via an NTB Crosslink. Both switches are partitioned similarly and connected to separate roots as shown in Figure 3. With this model, not only the heart beat message can be exchanged through the NTB Crosslink and NTB interconnect, the configuration TLPs can be sent by Root #2 to Root #1 and vice versa.

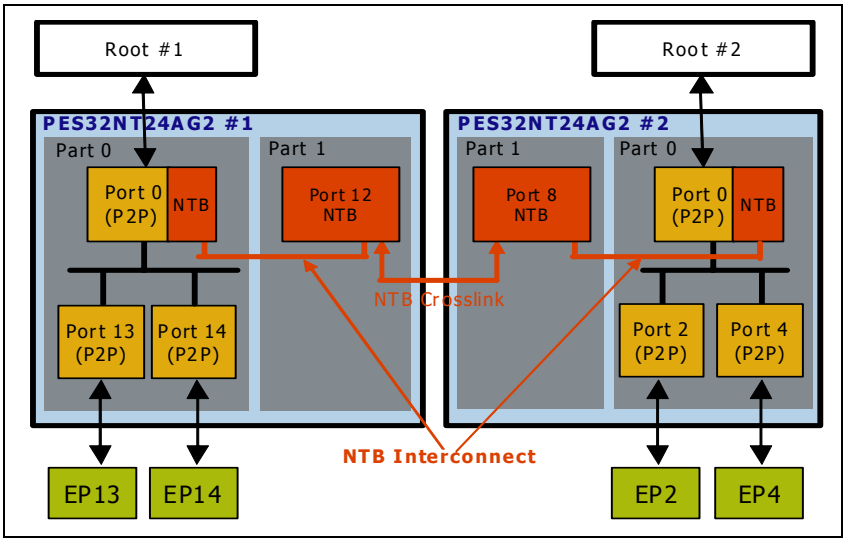

Figure 3 Dynamic Partitioning with NTB (Before Failover)

Assume that a user-initiated failover needs to take place so that I/O operations of EPs 13 and 14 can be managed via Root #2. Since the downstream switch ports associated with EPs 13 and 14 are physically in the PES32NT24AG2 #1 switch, it takes two phases to complete the transaction.

In the first phase, Root #2 engages the dynamic switch partitioning in the PES32NT24AG2 #1 switch by configuration writes via the Global Address Space Access Address (GASAADDR) and Global Address Space Access Data (GASADATA) registers of Port 12:

- Dynamically configure Port 0 from an upstream switch port with NTB function to an NTB function
- Dynamically configure Ports 13 and 14 and move them to Partition 1
- Dynamically configure Port 12 from NTB function to an upstream switch port with NTB function

The system software running in Root #1 re-scans and re-enumerates the PCI buses to realize the new topology illustrated in Figure 4.

Notes

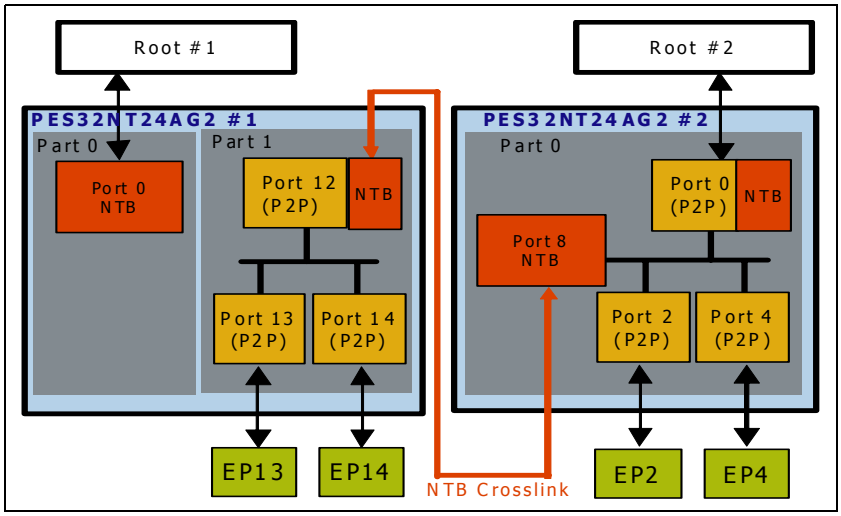

Figure 4 Dynamic Partitioning with NTB (After First Phase)

In the second phase, the following dynamic switch partitioning proceeds in the PES32NT24AG2 #2 switch:

- Dynamically configure Port 8 from an NTB function to a downstream switch port, and move it to Partition 0
- The link between Port 8 and Port 12 is no longer an NTB Crosslink. Instead, it's a link between a downstream port and an upstream switch port with NTB function.

The system software running in Root #2 re-scans and re-enumerates the PCI buses to realize the new topology illustrated in Figure 5.

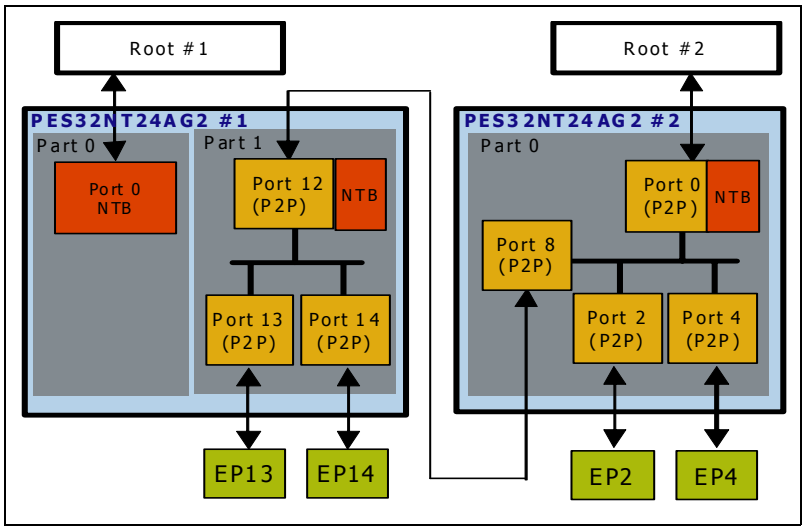

Figure 5 Dynamic Partitioning with NTB (After Failover)

Primary and Secondary Failover

The third usage model, shown in Figure 6, demonstrates a primary/secondary failover operation which can be achieved via a single trigger with the Failover Capability structures and its automatic reconfiguration designed in the PES32NT24AG2 switch. There are two partitions configured when the system is powered up. The partitions connect to separate root complexes, primary and secondary. The primary root complex manages EPs 11 and 14 which connect to two downstream switch ports, Ports 11 and 14, respectively.

Notes

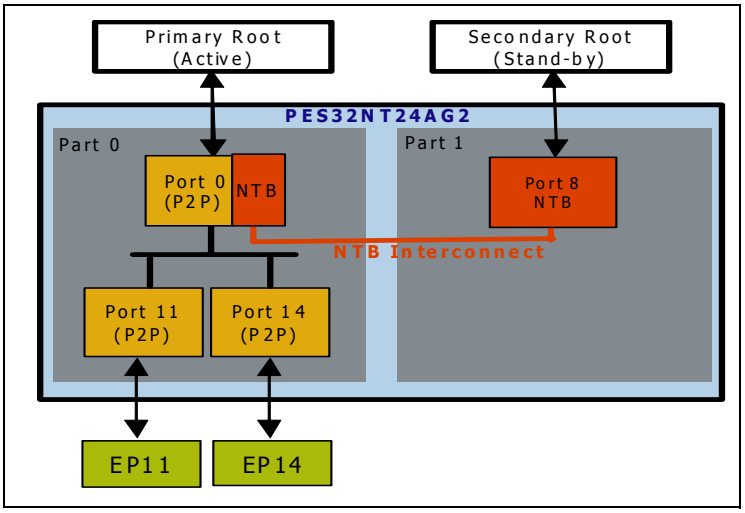

Figure 6 Primary and Secondary Failover (Before)

Assume that a user-initiated failover is engaged to move the downstream switch ports and the endpoints to the secondary root complex. After the necessary configurations in the selected Failover Capability and control registers are in place, a software-initiated failover and the automatic reconfiguration can take place via a configuration write (as a 1) to the Failover Software Trigger field of the Failover Capability Control register. The resulting topology is shown in Figure 7.

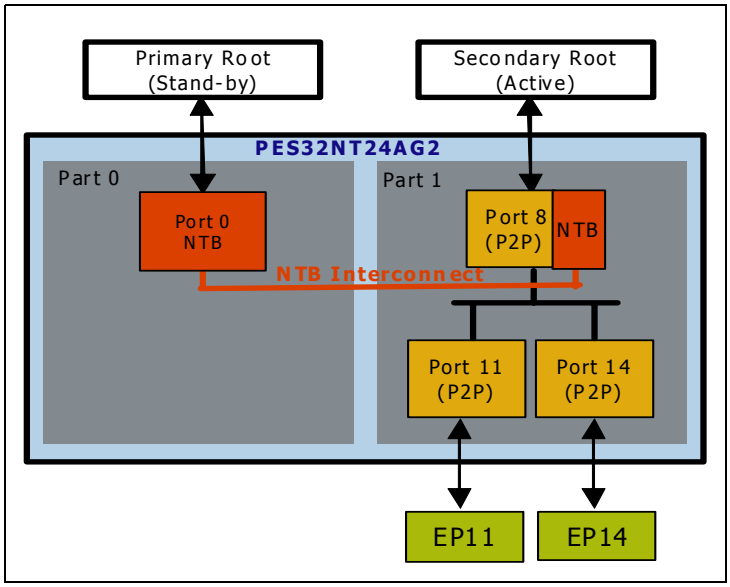

Figure 7 Primary and Secondary Failover (After)

Failover Configurations

As illustrated in the Primary and Secondary Failover usage model, the PES32NT24AG2 switch supports failover that is initiated with a single trigger, from primary to secondary or vice versa. This section describes how to configure the failover in greater details. Most of the settings can be done via a serial EEPROM while a fundamental reset is being applied to the system, but settings can also be modified dynamically.

Notes

Failover Capability

The PES32NT24AG2 switch supports four Failover Capability structures to increase the flexibility of the failover mechanism. Each structure includes three identical registers:

- Failover Capability Control register (FCAPCTL)
- Failover Capability Status register (FCAPSTS)
- Failover Capability Watchdog Timer (FCAPTIMER)

To configure failover that is initiated by a signal, the Failover Signal Trigger Enable (FSIGEN) field in FCAPxCTL needs to be set to 1. The polarity of the input signal can be configured via the Failover Signal Polarity (FSIGPOL) field in FCAPCTL, shown in Table 1.

	Active high (set as 0)	Active low (set as 1)
Primary failover	Signal transition from high to low	Signal transition from low to high
Secondary failover	Signal transition from low to high	Signal transition from high to low

Table 1 Failover Signal Polarity Settings

To configure failover initiated by a timer, the Watchdog Timer Count (COUNT) field in FCAPxTIMER register is used to set up the length of the period in microseconds before the timer expires. When the value transitions from 1 to zero and the Failover Timer Trigger Enable (FTIMEN) field in FCAPxCTL register is set to 1, then a failover is triggered.

There is no need to alter the capability registers to configure failover initiated by software.

Primary/Secondary Failover State

For a switch partition to function correctly, the Primary Failover Switch Partition State (PFSTATE) and Secondary Failover Switch Partition State (SFSTATE) fields in the associated Switch Partition x Failover Control (SWPARTxFCTL) register need to be specified with proper states for the switch partition. After a failover operation completes, one of the specified states is copied to the Switch Partition State (STATE) field in the associated Switch Partition x Control (SWPARTxCTL) register depending on the current failover mode indicated in the Failover Mode (FMODE) field of the Failover Capability x Status (FCAPxSTS) register associated with the failover operation.

For a switch port to function correctly, the fields in Switch Port x Failover Control (SWPORTxFCTL) register are transferred to the corresponding fields in its associated Switch Port x Failover Control (SWPORTxCTL) register after a failover operation completes. There are two sets of fields described in Table 2 for primary and secondary failover modes. It depends on the value of Failover Mode (FMODE) field in the Failover Capability x Status (FCAPxSTS) register associated with the failover operation.

Fields in SWPORTFCTL	Destined Fields in SWPORTCTL
Failover Port Mode (PFMODE or SFMODE)	Port Mode (MODE)
Failover Switch Partition (PFSWPART or SFSWPART)	Switch Partition (SWPART)
Failover Device Number (PFDEVNUM or SFDEVNUM)	Device Number (DEVNUM)

Table 2 Transferred Failover Control Fields For a Switch Port

Notes

Enabling Failover

In order to associate switch partitions and ports to the selected Failover Capability, the following configurations are required:

- Set Failover Enable (FEN) in the corresponding Switch Partition x Control (SWPARTxCTL) register to 1 to enable initiation of the failover reconfiguration.
- Set Failover Enable (FEN) in the corresponding Switch Port x Control (SWPORTxCTL) register to 1 to enable initiation of the failover reconfiguration.
- Set Failover Capability Select (FCAPSEL) in the corresponding Switch Partition x Control (SWPARTxCTL) register to select the associated Failover Capability.
- Set Failover Capability Select (FCAPSEL) in the corresponding Switch Port x Control (SWPORTxCTL) register to select the associated Failover Capability.
- Set Operating Mode Change Action (OMA) in the corresponding Switch Port x Control (SWPORTxCTL) register to 1 to ensure the proper port reset behavior associated with fundamental reset due to failover operation.

Failover Switch Events

The PES32NT24AG2 switch defines two switch events associated with failover operations:

- Failover Mode Change Initiated (FMCI) event is triggered by a Failover Capability
- Failover Mode Change Completed (FMCC) event occurs when a switch reconfiguration, resulting from a failover event, completes.

The FMCI event can be disabled by setting the Failover Capability x Failover Mode Change Initiated Mask (FCAPxFNCI) field in the Switch Event Failover Mask (SEFOVRMSK) register to 1 in order to mask the Failover Mode Change Initiated (FMCI) bit in the Failover Capability x Status (FCAPxSTS) register and to prevent generating a switch event.

The FMCC event can be disabled by setting the Failover Capability x Failover Mode Change Completed Mask (FCAPxFNCC) field in the Switch Event Failover Mask (SEFOVRMSK) register to 1 in order to mask the Failover Mode Change Initiated (FMCC) bit in the Failover Capability x Status (FCAPxSTS) register and to prevent generating a switch event.

Failover events can be configured independently regardless the given Failover Capability is being selected by any partitions or ports to initiate failover operation. When a failover switch event is not masked and is signaled to a partition, the event may be used to generate an MSI or INTx interrupt within the partition. Refer to section Failover Interrupts on page 8 for more details.

Failover Initiation

The PES32NT24AG2 switch supports four Failover Capability structures associated with each switch partition. Each structure has three registers associated with it and the registers can be used to set the policy for determining when a failover is triggered. There are three possible policies to initiate failover:

- Software initiated failover with a configuration register write,
- Watchdog timer initiated failover when watchdog timer expires, or
- Signal initiated failover when a device GPIO pin state transition happens.

In order to successfully complete the failover operation, both partitions and their associated ports must select the same Failover Capability structure. In other words, the same value must be programmed into the Failover Capability Select (FCAPSEL) field in the associated SWPARTxCTL and SWPORTxCTL registers. It's not prohibited to enable multiple policies in a single Failover Capability. However, an undefined behavior may occur if a second failover is triggered while an earlier failover is still in progress.

Software Initiated Failover

A failover event can be initiated by writing a 1 to the Failover Software Trigger (FSWTRIG) field in the Failover Capability Control (FCAPxCTL) register associated with the selected Failover Capability via Failover Capability Select (FCAPSEL) in the SWPARTxCTL and SWPORTxCTL registers. The current

Notes

failover mode reported in the Failover Mode (FMODE) field in the corresponding Failover Capability Status (FCAPxSTS) register will depend on the mode that existed before the triggering. If the previous mode was the secondary failover mode, the current mode becomes the primary failover mode, and vice versa.

Watchdog Timer Initiated Failover

A failover may be triggered as the result of an expiration of a Watchdog timer. Such a failover is initiated when the Failover Timer Trigger Enable (FTIMEN) field in the associated Failover Capability x Control (FCAPxCTL) register is set to 1 and the Count (COUNT) field in the Failover Capability Watchdog Timer (FCAPxTIMER) register transitions from 1 to zero.

The COUNT field provides a maximum watchdog timer interval of over one hour and may be written by software at any time to rearm the timer with desired interval before expiration.

Signal Initiated Failover

The alternate function of certain GPIO pins of the PES32NT24AG2 switch may be configured as an input to initiate the failover operation, referred as FAILOVERx signal, where x can be 0, 1, 2, or 3. The mapping of Failover Capabilities and GPIO pins are shown in Table 3 below.

GPIO Pin	Alternate Function 0	Alternate Function 1
4	Failover Capability 0	
6	Failover Capability 1	Failover Capability 3
7	Failover Capability 2	

Table 3 Mapping of Failover Capabilities and GPIO Pins

For example, when GPIO pin #4 is configured as Alternate Function 0, i.e., FAILOVER0, it means Failover Capability 0 is used to trigger failover. In addition to the failover configuration related to signal triggering mentioned previously, the following steps are required:

- Set the GPIO Function (GPIOFUNC) field in General Purpose I/O Function (GPIOFUNC) register as 0x10 to specify pin #4 is in Alternate Function mode.
- Set the GPIO Pin 4 Alternate Function Select (AFSEL4) field in General Purpose I/O Alternate Function Select (GPIOAFSEL) register as 0 to select Alternate Function 0.

Now, the state of FAILOVER0 can be retrieved from bit 4 of the GPIO Data (GPIOD) field in the General Purpose I/O Data (GPIOD) register. Whether it's a primary or secondary failover will depend on the polarity setting of the FAILOVER0 signal in the Failover Signal Polarity (FSIGPOL) field of the FCAPCTL register (refer to Table 1).

Note that the state of the FAILOVERx signal should not be changed at the same time that the signal's polarity is being changed. Otherwise, the failover may not be triggered. To avoid this situation, IDT recommends that polarity be configured prior to enabling the GPIO alternate function associated with the FAILOVERx signal. In addition, the state of the FAILOVERx signal should not be modified more frequently than once per second. Otherwise, undefined results may occur.

Failover Interrupts

The PES32NT24AG2 switch defines two switch events related to the failover operation: Failover mode change initiated and change completed. These may be used to generate interrupts within selected partitions. The interrupts can be utilized by system software as vital references in monitoring failover-related events, in collaborating with the system software to update current PCI device topologies and resources, and in managing (together with the software associated with the connected endpoint devices) the changes resulting from the failover operations.

Notes

Failover Event Signaling

Corresponding to the failover events is a status bit, FOVER, in the Switch Event Status (SESTS) register and a mask bit, FOVER, in the Switch Event Mask (SEMSK) register. When a failover event is detected, the event is signaled to all partitions not masked by the Switch Event Partition Mask (SEPMSK) register.

Figure 8 illustrates a simplified representation of the failover event detection and signaling mechanism. If a failover event is not masked in the Switch Event Failover Mask (SEFOVRMSK) register, when the event is generated the SESTS register logs the occurrence of the event by setting the FOVER bit to 1, the non-zero value of FOVER in SEMSK register signals the event to the partitions, and the bits of the Partition Mask (PMSK) field in the SEPMSK register controls which partitions are notified of the event occurrence.

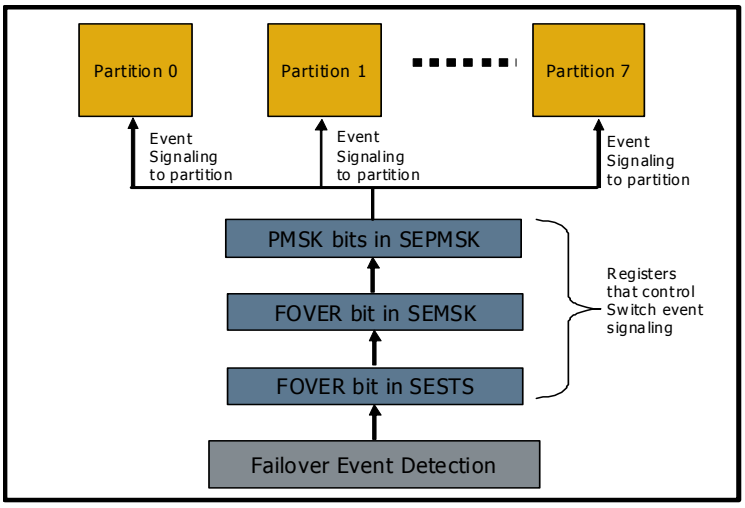

Figure 8 Failover Event Detection and Signaling Mechanism

Generating Failover Interrupts

When a failover switch event is signaled to a partition, the event, by default, will not generate an MSI or INTx interrupt within the partition. In order to generate Failover Mode Change Initiated (FMCI) or Completed (FMCC) interrupt due to the corresponding failover switch event, the corresponding mask bit needs to be unmasked. There are three possible scenarios that can occur based on the port mode setting of the associated upstream port in the partition.

- For transparent partitions (i.e., those without an NTB function), failover interrupts may be reported by the upstream switch port (i.e., the PCI-to-PCI bridge function) if the Failover Mode Change Initiated (FMCI) and/or Failover Mode Change Completed (FMCC) bits in the PCI-to-PCI Bridge Interrupt Mask (P2PINTMSK) register are cleared. In addition, when an FMCI or FMCC interrupt is triggered, the corresponding bit, FMCI or FMCC, in the PCI-to-PCI Bridge Interrupt Status (P2PINTSTS) register is set.
- For the upstream ports of partitions consisting only of an NTB function or an NTB function with DMA function, failover interrupts may be reported by the NTB function if the Failover Mode Change Initiated (FMCI) and/or Failover Mode Change Completed (FMCC) bits in the NT Endpoint Interrupt Mask (NTINTMSK) register are cleared. In addition, when an FMCI or FMCC interrupt is triggered, the corresponding bit, FMCI or FMCC, in the NT Endpoint Interrupt Status (NTINTSTS) register is set.
- For the partitions consisting of both PCI-to-PCI bridge and an NTB function, FMCI and FMCC interrupts may by reported by either one or both of the functions if the corresponding mask bit is cleared.

Notes

Example

In this section, an example of a step-by-step Failover Capability configuration and operation via signal initiation is provided, based on the topology illustrated in Figure 6. In this example, the Failover Capability 0 is selected for all the associated partitions and ports. The Alternate Function of GPIO pin #4 is used for the signal to initiate failover. Table 4 details the failover related configurations in the serial EEPROM.

Registers	Offsets	Values	Descriptions
SWPART0CTL	0x3E100	0x00080001	STATE = 1 (Active) FEN = 1 (Failover Enabled) FCAPSEL = 0 (Failover Capability 0 selected)
SWPART0FCTL	0x3E108	0x00000401	PFSTATE = 1 (Active Primary Failover state) SFSTATE = 1 (Active Secondary Failover state)
SWPART1CTL	0x3E120	0x00080001	STATE = 1 (Active) FEN = 1 (Failover Enabled) FCAPSEL = 0 (Failover Capability 0 selected)
SWPART1FCTL	0x3E128	0x00000401	PFSTATE = 1 (Active Primary Failover state) SFSTATE = 1 (Active Secondary Failover state)
SWPORT0CTL	0x3E200	0x00090004	MODE = 4 (Upstream switch port with NTB function) SWPART = 0 (Switch partition 0) DEVNUM = 0 (Device number 0) OMA = 1 (Operation Mode Change Action - reset) FEN = 1 (Failover Enabled) FCAPSEL = 0 (Failover Capability 0 selected)
SWPORT0FCTL	0x3E208	0x00130004	PFMODE = 4 (Upstream switch port with NTB function) PFSWPART = 0 (Primary switch partition 0) PFDEVNUM = 0 (Primary device number 0) SFMODE = 3 (NTB function) SFSWPART = 0 (Secondary switch partition 0) SFDEVNUM = 0 (Secondary device number 0)
SWPORT8CTL	0x3E300	0x00092013	MODE = 3 (NTB function) SWPART = 1 (Switch partition 1) DEVNUM = 8 (Device number 8) OMA = 1 (Operation Mode Change Action - reset) FEN = 1 (Failover Enabled) FCAPSEL = 0 (Failover Capability 0 selected)
SWPORT8FCTL	0x3E308	0x20142013	PFMODE = 3 (NTB function) PFSWPART = 1 (Primary switch partition 1) PFDEVNUM = 8 (Primary device number 8) SFMODE = 4 (Upstream switch port with NTB function) SFSWPART = 1 (Secondary switch partition 1) SFDEVNUM = 8 (Secondary device number 8)
SWPORT11CTL	0x3E360	0x00092C01	MODE = 1 (Downstream switch port) SWPART = 0 (Switch partition 0) DEVNUM = 11 (Device number 11) OMA = 1 (Operation Mode Change Action - reset) FEN = 1 (Failover Enabled) FCAPSEL = 0 (Failover Capability 0 selected)

Table 4 Serial EEPROM Image (Page 1 of 3)

Notes

Registers	Offsets	Values	Descriptions
SWPORT11FCTL	0x3E368	0x2C112C01	PFMODE = 1 (Downstream switch port) PFSWPART = 0 (Primary switch partition 0) PFDEVNUM = 11 (Primary device number 11) SFMODE = 1 (Downstream switch port) SFSWPART = 1 (Secondary switch partition 1) SFDEVNUM = 11 (Secondary device number 11)
SWPORT14CTL	0x3E3C0	0x00093801	MODE = 1 (Downstream switch port) SWPART = 0 (Switch partition 0) DEVNUM = 14 (Device number 14) OMA = 1 (Operation Mode Change Action - reset) FEN = 1 (Failover Enabled) FCAPSEL = 0 (Failover Capability 0 selected)
SWPORT14FCTL	0x3E3C8	0x38113801	PFMODE = 1 (Downstream switch port) PFSWPART = 0 (Primary switch partition 0) PFDEVNUM = 14 (Primary device number 14) SFMODE = 1 (Downstream switch port) SFSWPART = 1 (Secondary switch partition 1) SFDEVNUM = 14 (Secondary device number 14)
FCAP0CTL	0x3E500	0x00000002	FSIGEN = 1 (Failover Signal Trigger Enabled) FSIGPOL = 0 (Failover Signal Polarity as active high)
GPIOFUNC	0x3F16C	0x00000010	Set GPIO pin #4 as alternate function
SEMSK	0x3EC04	0xFFFFFF00	Enable generating switch events for Link Up, Link Dn, Fundamental Reset, Hot Reset, Failover and Global Signal.
SEPMSK	0x3EC08	0x000000FC	Enable receiving switch events in partition 0 and 1.
SEFOVRMSK	0x3EC2C	0x000E000E	FCAP0FNCI = 0 (Unmask mode change initiated event for Failover Capability 0) FCAP0FNCC = 0 (Unmask mode change completed event for Failover Capability 0)
SEGSIGMSK	0x3EC34	0x000000FC	Enable global switch event signaling to switch partition 0 and 1.
P0P2PINTMSK (Optional)	0x408	0x000000C0	SSIGNAL = 0 (Enable interrupt generating for switch signaling) SEVENT = 0 (Enable interrupt generating for switch events) FMCI = 0 (Enable interrupt generating for Failover Mode Change Initiated event) FMCI = 0 (Enable interrupt generating for Failover Mode Change Completed event)
P8P2PINTMSK (Optional)	0x10408	0x000000C0	SSIGNAL = 0 (Enable interrupt generating for switch signaling) SEVENT = 0 (Enable interrupt generating for switch events) FMCI = 0 (Enable interrupt generating for Failover Mode Change Initiated event) FMCI = 0 (Enable interrupt generating for Failover Mode Change Completed event)

Table 4 Serial EEPROM Image (Page 2 of 3)

Notes

Registers	Offsets	Values	Descriptions
P0NTINTMSK (Optional)	0x1408	0x000000C3	SSIGNAL = 0 (Enable interrupt generating for switch signaling) SEVENT = 0 (Enable interrupt generating for switch events) FMCI = 0 (Enable interrupt generating for Failover Mode Change Initiated event) FMCI = 0 (Enable interrupt generating for Failover Mode Change Completed event)
P8NTINTMSK (Optional)	0x11408	0x000000C3	SSIGNAL = 0 (Enable interrupt generating for switch signaling) SEVENT = 0 (Enable interrupt generating for switch events) FMCI = 0 (Enable interrupt generating for Failover Mode Change Initiated event) FMCI = 0 (Enable interrupt generating for Failover Mode Change Completed event)

Table 4 Serial EEPROM Image (Page 3 of 3)

1. System boots up:
   - Fundamental reset is applied to the system and all registers contain their default values
   - The PES32NT24AG2 switch powers up in switch mode "Multi-partition with Disabled ports and Serial EEPROM initialization" by setting SWMODE signals
2. After EEPROM loading completes, the roots proceed to configure the devices:
   - The primary root complex connects to a PCIe switch (Partition 0) with one upstream port and two downstream ports
   - The two downstream ports have device numbers 11 and 14
   - There is one endpoint device connecting to both downstream ports
   - The secondary root complex connects to port 8 of the PES32NT24AG2 switch. There is an endpoint (NTB function) enabled in Port 8
3. Now, if the platform asserts the failover signal trigger
   - A failover switch event is signaled to both partitions
   - Ports 11 and 14 are moved from partition 0 to partition 1 with same device numbers
   - Partition 0 becomes active and then port 0 becomes active as an endpoint function
   - Partition 1 becomes active and therefore port 8 becomes active as an upstream switch port of the partition
   - The new PCIe topology shown in Figure 7

If the platform de-asserts the failover signal and triggers another failover operation, the PES32NT24AG2 switch automatically re-configures itself back to the initial PCIe topology shown in Figure 6.

This example briefly discussed the Failover Capability structures and its automatic reconfiguration. Referring back to section Dynamic Partitioning with NTB on page 3, instead of following the dynamic switch partitioning methods in the first phase, Root #2 may initiate a software failover by writing a 1 in the Failover Software Trigger (FSWTRIG) field in the Failover Capability 0 Control (FCAP0CTL) register of Port 12, which achieves the same result in a signal trigger as long as the selected Failover Capability 0 structure and the corresponding failover control registers are pre-configured as below:

- Set up Partition 0, 1 and their associated ports to share the Failover Capability 0
- Program Partitions 0 and 1 to Active state for both primary and secondary failover states
- Program Port 0 mode to failover from an upstream switch port with NTB function to an NTB function with same device number
- Program Port 12 mode to failover from an NTB function to an upstream switch port with NTB function with same device number
- Program Ports 13 and 14 to failover from downstream ports of Partition 0 to Partition 1 with the same device number.

Notes

Hot Reset

Failover operation causes the initiation of a hot reset due to the data link layer of the upstream port reporting a DL_Down status. Each downstream switch port associated with the partition (if any) whose link is up propagates a hot reset to its associated link partner. However, the hot reset on the upstream port and propagation to its downstream ports caused by a DL_Down condition can be disabled by writing a 1 to the Disable Link Down Hot Reset (DLDHRST) field in the Switch Partition x Control (SWPARTxCTL) register.

System Software Requirements

The system software is required to work with the failover mechanism in responding to the PCI device topology changes after a failover operation. In order to ensure that all devices can function correctly after a failover operation completes, the system software is responsible for the following:

- Maintain an image of the PCI topology before and after failover operations
- Re-scan and re-enumerate a partial or the entire PCI topology whenever it is changed due to a failover operation
- De-allocate and re-allocate system resources, such as bus numbers, interrupts, and memory buffers, for the affected functions.

Summary

Dynamic switch partitioning supported by the PES32NT24AG2 switch serves as the basic function of the failover mechanism. Not only can the partitions be re-configured at run time, all 24 ports within the switch can be dynamically moved to the desired partition to better suit system requirements. Also, the re-configured upstream/NTB ports allow the switch to be designed into system interconnect applications that require efficient inter-processor communication to exchange information among hosts during failover operations.

In addition to dynamic switch partitioning and runtime re-configurable upstream/NTB ports, the single trigger failover with automatic reconfiguration via the Failover Capability structures provides a unique and efficient method for building high-performance, dual-host active/standby systems while ensuring system reliability and availability in case of a host failure. This programmable feature ensures that swift failover transactions will meet the requirements of high-performance systems.

The PES32NT24AG2 switch provides a wide variety of innovative features to support its flexible and advanced failover mechanism. With its innovative features, the failover mechanism can be used to build sophisticated storage and server systems having redundancy and high reliability. Finally, the switch gives users the ability to build their own fault-tolerant models without compromising their architectural strength because of limitations in existing system interconnect or inter-processor communications.