sRIO Hot Swap Application Note - 552

Introduction

The objective of this application note is to provide some hot swap solutions for IDT's Pre-Processing Switch (PPS) and Central Packet Switch (CPS) switch families with different system connections.

The PPS is specially designed for wireless telecommunication applications such as line card and baseband card. The CPS is a pure serial RapidIO (sRIO) switch. It can be also used in wireless telecommunication, IPTV (Internet Protocol Television) infrastructure and other multi-processor applications. The hot swap application is often used in these multi-connection systems. This document is intended to provide the user with the understanding of how to handle the hot swap application requirement in different system connections with IDT sRIO switches.

sRIO Protocol Background

The sRIO protocol is targeted for high speed chip-to-chip communication and data transfer. Compared to Gigabit Ethernet and PCI-Express, the sRIO protocol has minimal latency, less header overhead, higher efficiency and guaranteed delivery advantage.

To achieve this guaranteed delivery advantage, sRIO uses AckID (Acknowledge Identification) counters to keep track of each packet and ensure each packet has been delivered to each link partner, and ultimately to the final destination endpoint. The AckID is located in the physical layer, not in the packet header. Each port keeps track of their own AckID counters and these values are used in the packet acknowledgment control symbol to update the packet transmit status between devices.

For example, when the device sends out a packet, the Outbound AckID will increase 1, but the Outstanding AckID will not increase until it receives a response from the link partner that has accepted the same AckID packet. On the receiving side, when the device receives a packet, the device will use its Inbound ackID to respond to the sender. If the packet accepts, then the Inbound AckID will increase 1 for the next expected packet. The AckID counter is a five bit counter and when it reaches 31, it will rollover to “0”.

Each port has its own Inbound AckID, Outbound AckID, and Outstanding AckID. **Inbound AckID** is the next expected AckID value for an input port. When the input port receives a packet, it will use that value as the packet ID, send a response back to the sender and indicate whether or not the packet with packet ID has been accepted. **Outbound AckID** is the output port’s next packet AckID value. **Outstanding AckID** is the next expected packet AckID value. When the sending port receives a packet-accept control symbol at its receiver, the sender will compare the response AckID with the Outstanding AckID. If it matches, it will increase the outstanding AckID by 1. Otherwise, it will report "unexpected AckID error". That is to say that the sender’s Outstanding AckID should be equal to the receiver’s Inbound AckID. Please refer to the RapidIO Interconnect Specification Part 6: 1x/4x LP-Serial Physical Layer Specification Rev. 1.3 for full detail.

Hot Swap Challenge

The main application challenge to support hot swap is to match the Inbound and Outbound AckID. As mentioned, the sRIO link protocol does provide an optional Software Assisted Error Recovery mechanism. This mechanism uses per-port Control Status Registers (CSRs) that can be accessed remotely over an sRIO link using maintenance packets or directly accessed by a local processor. These CSRs enable software to cause a port to initiate the transmission of link-request control symbols at the physical layer. They can recover the contents of link-response control symbols received by the port. They also enable software access to and control over the port's various AckID registers. This enables the AckIDs to match in link partners and traffic can resume. Serial RapidIO provides optional Software Assisted Error Recovery registers and mechanisms to support hot swap.

Notes

When one of the devices connected by a sRIO link is hot swapped, its power is either cycled or it is reset. It is highly unlikely that the device connected by the link will be able to exchange packets due to packet AckID mismatch. In accordance with sRIO specification, sRIO ports on the device that have been hot swapped where power has been cycled or reset, will begin packet transmission with AckID value of "0" and expect to receive packets from the connected port beginning with AckID value of "0". If the connected port was previously transmitting and receiving packets, it is unlikely that it will resume packet transmission with AckID value of zero or expect the next packet received to have an AckID value of zero.

These mismatches will result in "unexpected AckID" errors which will cause transmitted packets to be rejected by the receiving port and the device will go into an output-error stopped state. When the output error remains, then it will run into a fatal error state. However, it does not allow automatic hardware resynchronization of the AckID. It does allows the software to resynchronize the AckIDs so that packet transmission can resume. This provides the user a strong level of control and flexibility in how and when to resolve a broken link. It also places a requirement for software to manage any hot-swap activity.

This document illustrates several hot swap scenarios using different IDT switch solutions to demonstrate how the AckID resynchronization can be achieved.

IDT Hot Swap Support Features

The PPS Gen 2 (80KSW0001), CPS family (80KSW0002 CPS-16, 80KSW0003 CPS-8, and 80KSW0004 CPS-12), and Serial Buffer (80KSBR200) all have "Software Assist Error Recovery" Control and Status Registers (CSRs) in accordance with the optional sRIO specifications. Note that the PPS (70K2000) does not have the Software Assist Error Recovery registers, but it does have proprietary implementations to support hot swap described below.

The Software Assist Error Recovery registers include a local AckID-Status read/write register, Link-Maintenance-Request register, and Link-Response-Status register. The **AckID-Status** register allows the user, via software, to manually set the AckID values or read back the AckID values. The **Link-Maintenance-Request** register allows user generate a link-request or reset-device control symbol. The link-request control symbol will cause a link response from the link partner. The reset-device control symbol will cause the link partner to reset the device or the port. The **Link-Response-Status** register keeps track the link response and reports this in the status field. The status fields indicate that the link response has been received and the status fields are valid. If the link request isn't followed by a link response, this bit indicates that the link request has been transmitted. This bit is clear on read. The AckID status field will contain the last response control symbol value.

In addition to the Software Assist Error Recovery registers, all of IDT's switch device support a proprietary "per-port reset" option. In accordance with sRIO specification, a device receiving a reset control symbol must perform a full device reset. For switches, this would mean clearing all route tables and resetting all ports. Because this might halt all traffic in a system, this may not be suitable for a run-time application. IDT provides an alternative control to allow this reset control symbol to instead reset JUST the local sRIO port that the control symbol was received on. This function is provided by the "PPS_CONTROL" register's "enable_port_reset" field for the PPS family of devices, or "CPS CONTROL" register's "PORT_RESET_BEHAVIOR" register for the CPS family of devices. Please refer to the device's user's manual for full details.

It should also be noted that, for IDT's sRIO devices, disabling any port also has the effect of resetting to zero the Inbound, Outbound, and Outstanding AckID registers for all devices. Thus, a simple disable of the port - a typical state transition during a hot-swap insertion or extraction - is sufficient to clear the AckIDs and start resynchronized when a new link partner is inserted and re-powered. Also note that when an output port is disabled, either through "Port Disable" (power down) or "Output Port Disable" (via sRIO CSR to allow only non-maintenance packets), or the port is in a fatal error state, packets destined to this output port will be dropped. This ensures that traffic does not congest at a port that was intentionally disabled, thereby preventing congestion back to the input ports. This ultimately prevents the entire system from congesting.

Notes

Solutions for Hot Swap

The usage scenario is that if a new board (B) is connected and powered up to an already powered-on board (A); board B will have AckIDs set to "0" and board A most likely will have a non-zero AckID. Therefore, the AckIDs of A and B boards will be mismatched. This will cause an error stopped state on the link. The switch and link partner can issue a link-request control symbol which determines the expected Outbound AckID value. This AckID value can be programmed into boards A and B. At which point, the link will be recovered. However, since packet transfer is not possible over a sRIO link with unsynchronized AckIDs, remote access to these CSRs is not available when they are on a port that has been hot swapped. Resetting the AckID to "0" for both sides is a simple way to resynchronize the AckID.

If the system has non-stop traffic, then the disconnected output port needs to be disabled before unplug. Otherwise, the traffic will back pressure to the sender and cause the input buffer to go full such that no other packet can go through. When the output port is disabled, all the packets sent to that disabled output port will be dropped. In this case, it will not back pressure to the input port(s) and ultimately congest the system.

There are different ways to perform the AckID re-synchronization. There are many variables including: which device it is connected to, which side of the host device it is at, and which side is powered down. In general, there are two solutions to re-synchronize the AckID:

- *Solution 1: Hardware reset Inbound/Outbound AckID to "0" can reset the port in five different ways. It is the most simple and direct solution, but the disadvantage is that all packets in the retransmit buffer will be lost.*
- *Solution 2: The Software Assisted Error Recovery Mechanism can be used to change the Inbound/ Outbound AckID enabling a match to the other side. The Software Assisted Error Recovery Mechanism needs a host device to execute this process. There are specific procedures that need to be executed to ensure AckIDs are matched during hot swap.*

Solution 1: Hardware reset the Inbound/Outbound AckID to "0". There are five different ways to reset the AckID on the switch. Methods a, b, and c below require the link partner to change its own AckID to "0" after unplug and before replug. Methods d and e will reset link partner's AckID automatically. All methods below will also reset the retransmit buffer, which means all packets in the retransmit buffer will be lost.

a) Disable and enable the port. It will reset Inbound and Outbound AckID to "0". (See the PORT_n_CTRL_CSR RIO Register for more detail.)

b) Hard reset by pulling the reset pin low. It requires reprogramming the switch after reset.

c) Soft reset by write to the SOFT_RESET register. It also requires reprogramming the switch after reset.

d) Reset the port by sending a reset control symbol. The PPS Gen 2 and CPS family are able to generate a reset via the Software Assist Error Recovery registers.

e) Reset the port or reset the chip only by receiving the sRIO reset control symbol. The port reset option will reset the particular port only when the port receives a reset control symbol. Refer to the device user's manual page, CPS/PPS_CONTROL register, "RESET_BEHAVIOR" field for details. This control register allows the user to have an option to reset either the port only or the entire switch. Note that the default, in accordance with sRIO specification, is to reset the full chip. If this behavior is not desired, it is recommended to always set this control to allow per-port reset. All IDT sRIO switches have this per-port reset option.

Solution 2: Program the AckID register to match the other side. Note, the PPS (70K2000) doesn't apply in this case, since it doesn't have the Software Assist Error Recovery registers. There are three different ways to get the same the AckID on both sides. Method a and b each require matching the AckID before link up or stop all traffic after link up. Method c allows matching the AckID after link up. Method c will stop all transitions on both sides.

a) Have a common value which will be used on both sides when the link down, such as "0" or any other fixed value.

b) Have a second bus connected to the link partner and identify the AckID.

c) Send a link-request control symbol to the link partner and get its Inbound AckID value. Pro-

Notes

gram that value into the switch, where the Outbound AckID will match the link partner Inbound AckID. The link partner also needs to do the same procedure, by sending a link-request to the switch and by using the switch Inbound AckID value for its Outbound AckID

Note: In the current sRIO Rev1.3 implementation, the Inbound AckID fields and the Outbound AckID fields are implemented as readable/writable fields in the same register. This allows software to update both fields with a single write to that register-address which works well for hot swap scenarios. However, if the software is to update the Outbound AckID field only (as in a "only TX link is locked up" scenario), then it is recommended to stop traffic on the RX link (from the link partner to our device) also. Stopping traffic on the RX link is needed to allow the Inbound AckID field to reach a constant value which can then be read and written back in and along with the new value of the Outbound AckID field.

Fault Management: Link Error Detection

Fault management features support hot swap requirements. IDT's switches and endpoints employ error handling and management functions in support of detecting, logging, and reporting all types of error conditions from all physical-layer, link-layer, ports, and other functional blocks. This allows, among others, detection and reporting when a link is disconnected and in error. The function thus allows, with appropriate software programming, for the devices to automatically detect and respond to error conditions and for a host processor to take appropriate response to those conditions.

The link error detection plays a very important role in the hot swap application. There are many ways to do the link error detection. Both switch and link partner can do the link error detection. It can also be done in the hardware by having an unplug switch or sensor built into the board. When the device unplugs, it can signal to the host device.

IDT's switches provide customers another option for link-off error detection. All IDT switches provide the ability to detect, record, and report error and status information to a host device. Error reporting capabilitis are configured and enabled through the Special Error Filter registers. There are eight Special Error filter registers which allow the user to configure special behaviors when a specific error type is received by the Error Management logic. The user can configure each filter to track a specific error type (defined by Number, Group, and Source). Once configured, the device can then count the error, flag the error (generate an active-low external interrupt signal (/IRQ)), initiate the generation and transmission of a maintenance packet, or stop the Error Management function altogether. Each of the eight registers also allow the user to mask the error Source, Group, and Number values. There are only two steps to program the error detection in the switch:

Step 1: Enable the error report management function block in IDT's switch. Refer to the device user's manual's "Error Handling" chapter for full detail. This step allows the switch to detect any error conditions anywhere in the device.

Step 2: Program the Special Error Filter registers. As it applies to hot swap, the device can detect and also report if a link is broken (physical or link layer errors), and the host device can take appropriate action (see step 3) to recover. There are different ways to report errors:

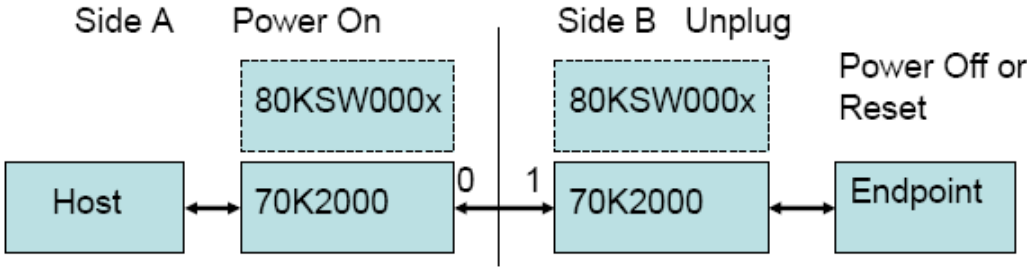
1. Send out an sRIO maintenance packet to the host device.
2. Generate an interrupt signal.
3. The CPS family can also report errors through I^2C bus in master mode only.

Notes

Hot Swap Case Examples:

Case 1: Hot Swap between IDT switches.

Condition: Only one Host and it is on IDT switch side. Side B will power off or reset during unplug.



Solution:

Step1: Disable the port 0

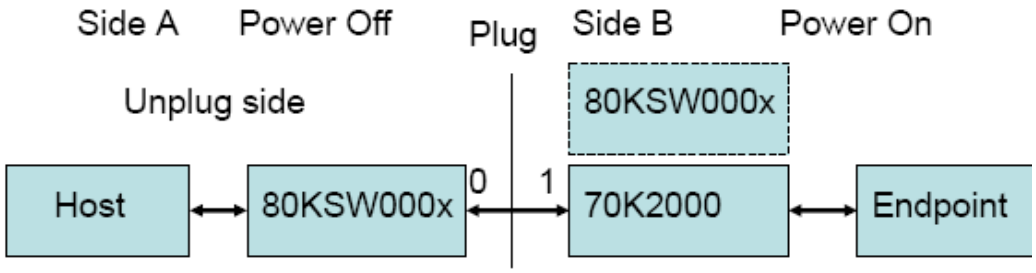
Step2: Enable the port 0.

Step3: Plug in the link or power up side B.

Note: This solution is assumes that side B AckID is always reset to "0" before link up.

Case 2: Hot Swap between IDT switches.

Condition: Only one Host and it is on powered off side. The power will turn off or reset on side A. Side B is power on all the time.



Solution:

Step1: Enable port reset option at side B before unplug (Register address 0xF2000C, bit0 for detail).

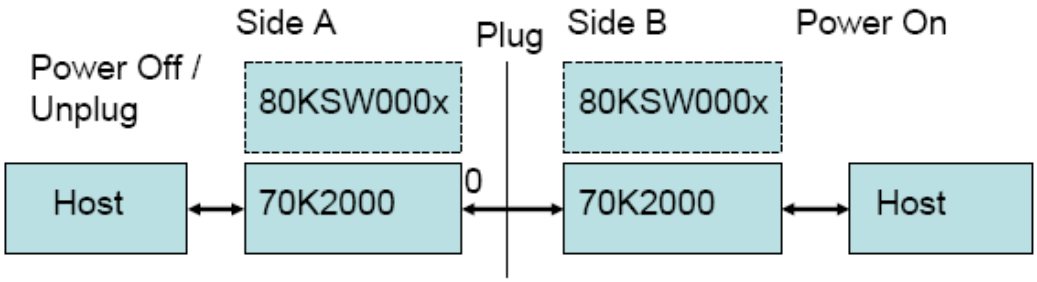
Step 2: After power up, send reset control symbol to side B (See register 0x000140). It will reset both side AckID to "0".

Notes

Case 3: Hot Swap between IDT switches.

Condition: Both sides has host device

.

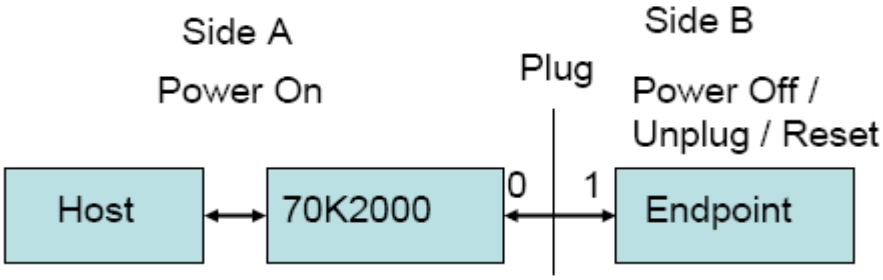


Solution: See solutions for Case 1 or Case 2 to solve this issue.

Note: Case 2 is for PPS Gen 2 and CPS serial in side A only. 70K2000 will not apply in this case. See Case 2 connection.

Case 4: Hot Swap between an IDT switch and any other endpoint device.

Condition: Only one Host and it is on side A (IDT switch side). The power will turn off or reset on side B.



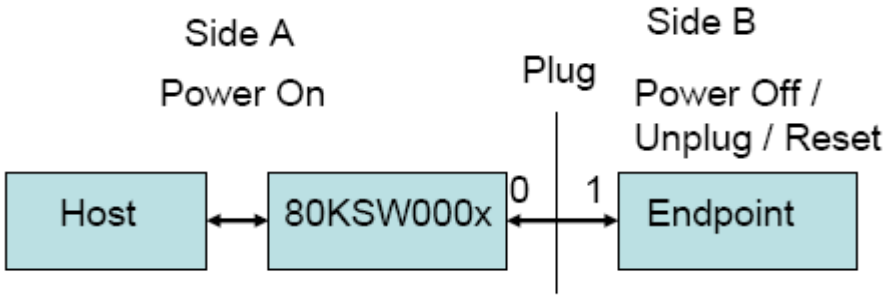
Solution 1: See solution Case 1.

Solution 2: If endpoint at side B supports Software Assist Error Recovery, then 70K2000 can be programmed into pre-port reset mode before unplug or power down. After power on side B, the end point can issue a reset control symbol to 70K2000 and reset 70K2000's AckID.

Notes

Case 5: Hot Swap between an IDT switch and any other endpoint device.

Condition: Only one Host and it is on side A (continuously powered). The power will turn off or reset on side B.



Solution 1: If side B endpoint is power off, then see solution Case 1 and 4.

Solution 2: If side B endpoint has either just been reset or reprogrammed, then the endpoint will reset Outbound AckID and/or Inbound AckID. Solution case 1 will fix this issue. If the endpoint only resets the Outbound AckID, then it will need to do more steps to solve this issue by using the Software Assist Error Recovery Mechanism.

Step 1: Stop traffic on both sides, if that applies.

Step 2: Send a link-request control symbol to endpoint. (See register 0xh140.)

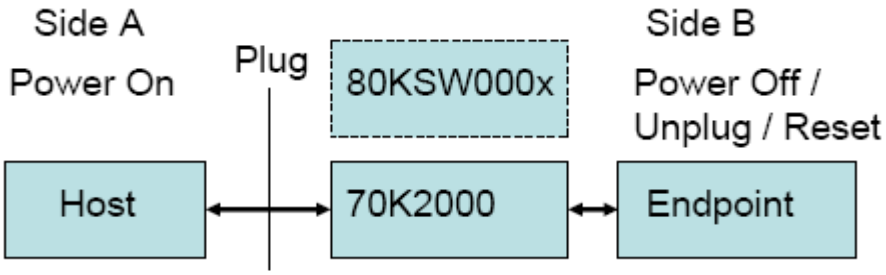
Step 3: Read back the link-request respond AckID value. (See register 0xh144.)

Step 4: Use endpoint's Inbound AckID as Outbound AckID.

Step 5: The end point needs to repeat same procedure from steps 2 to 4.

Case 6: Hot Swap between IDT switch and any other endpoint device.

Condition: Host device is on side A, IDT switch is on side B. The power will turn off or reset on side B.



Solution 1: If the switch side is power down and power up, then host endpoint needs to reset AckID to "0" before sending traffic over.

Notes

Hot Swap support registers reference

The following is a short general listing of some of the registers discussed in this application note. These registers are generally applicable to IDT devices. However, please always refer to the specific device's user's manual for the exact register implementation details.

Port Maintenance Request Command and Status Register

The Port Maintenance Request Command and Status Register is also called "sRIO Software Assisted Error Recovery Register". It has three registers, Port Link Maintenance Request Register, Port Link Maintenance Response Register and Local Port AckID Register. The Port Link Maintenance Request Register can generate a reset or link-request control symbol to the link partner. The Port Link Maintenance Respond Register is used to store the link-request respond information, which is content link partner Inbound AckID information. The Local Port AckID Register is able to read or change the local port AckID value.

The port specific register PORT_0_CTRL_CSR shown below is replicated, so there exists one of each per port. The following table shows the addresses for these registers.

Base Address (Hex)	Associated Registers
0x000140 - 0x000148	Port 0
0x000160 - 0x000168	Port 1
0x000180 - 0x000188	Port 2
0x0001A0 - 0x0001A8	Port 3
So on	So on

Table 1 Base Address for the local port control CSR Register

The Port Link Maintenance Request Register is accessible both by a local processor and an external device. It can generate a link-request or reset control symbol on the corresponding RapidIO port interface by writing to this register with value of 0b011 or 0b100.

Bit	Field Name	Type	Reset Value	Comment
2 - 0	Command	R/W	0b000	0b011 = Reset device 0b100 = Input status 0b000 - 0b010 = Reserved 0b101 - 0b111 = Reserved See RIO part 6 table 3-6 (rev 1.3)
31 - 3				Reserved

Table 2 PORT_0_LINK_MAINT_REQ_CSR 0x000140

The Port Link Maintenance Response Registers are accessible both by a local processor and an external device. A read to this register returns the status received in a link-response control symbol. The AckID_status and port_status fields are defined in sRIO 1.3 specification part 6 Table 3-3 and Table 3-5. This register is read-only.

Bit	Field Name	Type	Reset Value	Comment
4 - 0	LINK_STATUS	RO		Link status field from the link-response control symbol
9 - 5	ACKID_STATUS	RO		AckID status field from the link-response control symbol

Table 3 PORT_0_LINK_MAINT_RESP_CSR 0x000144

Bit	Field Name	Type	Reset Value	Comment
30 - 10				Reserved
31	Response_valid	RO		If the previous link-request causes a link-response, then this bit indicates that the link-response has been received and the status fields are valid. If the link-request didn't cause a link-response, this bit indicates that the link-request has been transmitted. This bit clear on read.

Table 3 PORT_0_LINK_MAINT_RESP_CSR 0x000144

The Local Port Link AckID status register is accessible both by a local processor and an external device. A read to this register returns the local AckID status for both the out and input ports of the device. A write to this register will update a new Outbound and Inbound AckID into the device.

Bit	Field Name	Type	Reset Value	Comment
4 - 0	Outbound_ackID			The next transmitted AckID value for the port. Writing this value can force retransmission of outstanding unacknowledged packets in order to manually implement error recovery
7 - 5				Reserved
12 - 8	Outstanding_ackID			The output port unacknowledged AckID status. The next acknowledge control symbol AckID field that indicates the AckID value expected in the next received acknowledge control symbol.
23 - 13				Reserved
28 - 24	Inbound_ackID			Input port next expected AckID
30 - 29				Reserved
31	CLR_Outstanding_ackIDs			Writing a value of 1 to this bit causes all outstanding unacknowledged packets to be discarded. This bit should only be written when trying to recover a failed link. This bit will always return a value of zero when read.

Table 4 PORT_0_LOCAL_ACKID_CSR 0x000148

Clr_Outstanding_AckIDs

This is a write-only field. When this bit is written to a value of 1, it treats all previously transmitted packets for which AckIDs have not been received as having been properly received by the link partner. Acknowledgment processing for these packets will no longer be required. If the Outbound AckID is going to be changed at the same time, it must be set to 1. This will drop packets in the retransmit buffer.

Inbound_AckID

A read of this register will return the expected AckID of the next packet that comes into the device.

A write of this register will set the expected AckID of the next incoming packet to be as specified. If the receiver is in any stopped state, it will be return to the normal operating state. If any packet is being received, it will be dropped without a response. If the user intended to transmit the dropped packet, the AckID will time-out per protocol and the packet will be resent.

Outstanding_AckID

A read of this register will return the AckID that is expected on the next received packet accept.

A write of this register can have a few effects based on the Outstanding AckIDs:

1. If no AckIDs are outstanding, the write will have no effect on the port.

2. If AckIDs are outstanding and the specified AckID is one of them, any AckIDs lower than the one specified will be automatically accepted. There is a similar effect if a link-response has come in with the specified AckID. However, this field value will have no effect on the AckID of the next packet to be transmitted.

3. If AckIDs are outstanding and the specified AckID is not one of them, an error will be flagged and the device will take no further action.

Outbound_AckID

A read of this register will return the AckID of the next packet that will be transmitted.

A write of this register will cause all outstanding packets to be retransmitted starting with the specified AckID. This occurs after the write of the Outstanding_AckID field has been fully processed. If no packets are outstanding, the next packet available will be transmitted with the specified AckID. The read value of the outstanding AckID will be adjusted accordingly.

sRIO Port Control Register

The port specific register PORT_0_CTRL_CSR shown below is replicated, so there exists one of each per port. The following table shows the addresses for these registers.

Base Address (Hex)	Associated Registers
0x00015C	Port_0_CTRL_CSR
0x00017C	Port_1_CTRL_CSR
0x00019C	Port_2_CTRL_CSR
0x0001BC	Port_3_CTRL_CSR
So On	So On

Table 5 Base Addresses for RIO Port Control Register

Bit	Field Name	Type	Reset Value	Comment
0	Port_Type	FR	0b1	0 = Parallel
3 - 1				Reserved
11 - 4	Retransmit_Suppression_Mask	R/W	0b0	Mask for retransmission suppression
16 - 12				Reserved
17	Enumberation_Boundary	R/W	0b0	The enumeration boundary
18				Reserved
19	Multicast_Event_Participation	R/W	0b0	Multicast event participation
20	Error_CHK_Disable	R/W	0b0	Disable error checking
21	Input_Port_Enable	R/W	0b0	Enable the Input Port
22	Output_Port_Enable	R/W	0b0	Enable the Output Port
23	Port_Disable	R/W	0b0	Port Disable
26 - 24	Port_Width_Override	R/W	0b000	000 = no override 010 = single lane port lane 0 011 = single lane port lane 2 Others reserved

Table 6 PORT_0_CTRL_CSR 0x00015C

Notes

Bit	Field Name	Type	Reset Value	Comment
29 -27	Init_Port_Width	RO	0b000	Initialized Port Width 000 = single lane port lane 0 001 = single lane port lane 2 010 = 4 Lane Port Others reserved
31 - 30	Port_Width	RO	0b00	00 = Single Lane Port 01 = 4 Lane Port Others Reserved

Table 6 PORT_0_CTRL_CSR 0x00015C

Port Error Report Enable Registers

The following table provides the base addresses of those registers associated with the switch port registers. The definitions of the switch port registers follow. The register definition includes the associated offset from the base address of a given port.

Base Address (Hex)	Associated Registers
0xF40008	Port 0 Error Report Enable Register
0xF40108	Port 1 Error Report Enable Register
0xF40208	Port 2 Error Report Enable Register
0xF40308	Port 3 Error Report Enable Register
So on	So on

Table 7 Base Address for Port Error Report Enable Register

Bit	Field Name	Type	Reset Value	Comment
0	Error_Report_Enable	R/W	0b0	0=Disable Error Reporting from this port 1=Enable Error Reporting from this port
1	Switch_Port_Error_Report_Enable	R/W	0b0	0=Disable error reporting from switch buffers 1=Enable error reporting from switch buffers
2	Retry_Error_Report_Enable	R/W	0b0	0=Disable Retry Symbol Received Reporting from Port 1=Enable Retry Symbol Received Reporting from Port
31 - 3				Reserved

Table 8 Port 0 Port_ERR_Report_Enable 0xF40008

Special Error Register

There are eight Special Filter Error Registers in the PPS and CPS switch family. The SPECIAL_ERR_0 is shown below. It is replicated to show each of the eight Special Filter Error Registers.

Base Address (Hex)	Associated Registers
0xFD0008	SPECIAL_ERR_REG_0
0xFD000C	SPECIAL_ERR_REG_1
0xFD0010	SPECIAL_ERR_REG_2
0xFD0014	SPECIAL_ERR_REG_3
0xFD0018	SPECIAL_ERR_REG_4

Table 9 Base Address for Special Error Filter Registers

Notes

Base Address (Hex)	Associated Registers
0xFD0020	SPECIAL_ERR_REG_5
0xFD0024	SPECIAL_ERR_REG_6
0xFD0028	SPECIAL_ERR_REG_7

Table 9 Base Address for Special Error Filter Registers

Bit	Field Name	Type	Reset Value	Comment
3 - 0	ERROR_NUMBER	R/W	0b0000	Error Number
7 - 4	ERROR_GROUP	R/W	0b0000	Error Group
13 - 8	ERROR_SOURCE	R/W	0x00	Error Source (6 bits)
15 - 14				Reserved
16	STOP_ENABLE	R/W	0b0	Enable stopping the error management function and generation of a maintenance packet
17	Maintenance_Packet_Enable	R/W	0b0	Enable the generation of a maintenance packet in event of an error
18	FLAG_ENABLE	R/W	0b0	Enable flagging the error
19	COUNT_ENABLE	R/W	0b0	Enable counting the error
20	ERROR_NUMBER_MASK	R/W	0b0	0 = compare the error number 1 = do not compare the error number
21	ERROR_GROUP_MASK	R/W	0b0	0 = compare the error group 1 = do not compare the error group
22	ERROR_SOURCE_MASK	R/W	0b0	0 = compare the error source 1 = do not compare the error source
31 - 23				Reserved

Table 10 Special Error Register 0: 0xFD0008

* All registers can be programmed through sRIO, I^2C, or JTAG bus.

02/22/08: Initial application note

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

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/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.