Device Errata Summary

The following table summarizes the revision applicability and status of the SPS-1616 device errata. Each device contains a topmark date code, XXYY, which indicates the device's production date. Errata fixes are tracked according to a device's date code, where XX is the year and YY is the work week. The revision identifier refers to Device Information CAR[MINOR_REV].

Table 1. Device Errata Summary

Errata	Applicability			Status
	Revision A (1024–1119)	Revision B (1120–XXYY)	Revision C (80HSPS1616C)	
Switch is active when RST_N is asserted	●			Fixed
No PRBS checking	●	●	●	Viable work around[a]
Loopback mode restrictions	●			Fixed
JTAG 1149.1 chaining does not support register access	●	●	●	Viable work around
Packet generator/checker restrictions	●	●	●	Viable work around
Incorrect MCAST_MASK field	●	●	●	Viable work around
Extra field in Lane n Status 0 CSR	●	●	●	Viable work around
Ports fail to initialize	●			Fixed
Port cannot send packets	●			Fixed
Dynamic control of link partner transmit emphasis not working correctly	●	●	●	Viable work around
Maintenance packet buffer management	●	●	●	Viable work around
Maintenance packets congest input buffer	●			Fixed
Lane swapping may prevent port from achieving PORT_OK	●	●	●	Viable work around
Protocol error stops packet transmission	●	●		Fixed
IDLE2 defects	●	●		Fixed
False idle character in packet detection	●	●	●	Viable work around
Improper auto-negotiation of IDLE2 to IDLE1	●	●	●	Viable work around
Unable to downgrade in 2x mode when primary lane is disabled	●	●	●	Viable work around
Unable to downgrade when primary lane is inoperable	●	●	●	Viable work around
Packet with unexpected ackID is incompletely captured	●	●	●	No plans to fix
Low throughput with maximum sized packets	●	●	●	Viable work around
False unexpected ackID error detection	●	●	●	Viable work around
False loss of descrambler sync error detection	●	●	●	Viable work around
PORT_OK may indicate incorrect status	●	●	●	Viable work around
Receiving port's ackID not properly reset upon receipt of link-request/reset-device control symbol	●	●	●	Viable work around

Table 1. Device Errata Summary (Cont.)

	Applicability			
Errata	Revision A (1024–1119)	Revision B (1120–XXYY)	Revision C (80HSPS1616C)	Status
Clock frequency tolerance tighter than specification for 6.25 Gbaud	●	●	●	Viable work around
Subset of JTAG commands power down SerDes PLL	●	●	●	Viable work around
False BAD_PORT error detection	●	●	●	Viable work around
Writing Routing Table Entries Causes Dropped Packets	●	●	●	Viable work around

[a] "Viable work around" indicates a work around that allows a customer's products to reach production status.

Switch is active when RST_N is asserted

Description

The SPS-1616 is not held in reset when RST_N is asserted. Registers can be read/written, and ports will complete their initialization sequence when connected to a link partner.

The SPS-1616 is reset when RST_N is deasserted. At this time the ports reinitialize.

Impact

The SPS-1616 may attempt initialization before power supplies or reference clocks are stable, resulting in undefined operation.

Re-initialization of the link after successfully connecting with the link partner may lead to error conditions on the link that must be cleared before packets can be exchanged.

Workaround

Assert and deassert RST_N periodically to ensure that the SPS-1616 remains in reset. A low-to-high transition of RST_N must occur at least every 4 usec, and can occur as often as every 50 nsec, to keep the SPS-1616 in reset.

If it is not possible to implement the work around, then error conditions on the link may occur. The SPS-1616 link error states are cleared automatically. To clear link error states in a timely fashion, set the Port Link Timeout Control CSR to a value consistent with link partner's capabilities. Typically, this is in the range of 5 to 10 usec.

Status

Fixed in all parts with a date code of 1120 or later.

No PRBS checking

Description

The SPS-1616 cannot check Pseudo-Random Bit Sequence (PRBS) patterns.

Impact

The SPS-1616’s receive lane signal integrity verification cannot use PRBS bit patterns.

Workaround

RapidIO compliant data can be used to perform signal integrity analysis.

The IDLE2 sequence, which contains scrambled D0.0 characters and all possible RapidIO control characters, can be used to perform signal integrity analysis.

Packets containing random data must be exchanged on links that support IDLE1 to ensure that all valid 10B codes are transmitted/received.

RapidIO Error Management Extensions registers, and/or the Lane Status registers, can be used to determine the bit error rate on the links.

To use the Error Management Extensions to determine the total bit error rate for all receive lanes associated with a port, set the following registers as indicated:

- Port n Error Rate Enable CSR = 0x004E8015
- Port n Error Rate CSR = 0x00030000
- Port n Error Rate Threshold CSR = 0x00000000

The number of errors for a time period is located in the Port n Error Rate CSR.ERR_RATE_CNTR field. The number of errors that result in detectable corruption of 10-bit code groups on the link is located in the Lane n Status 0 CSR.ERR_8B10B field.

Status

There is a viable work around for this issue.

Loopback mode restrictions

Description

The following SPS-1616 loopback modes function only with RapidIO compliant data:

- 10-bit loopback
- 8-bit loopback

Impact

The implication of the restriction on 10-bit and 8-bit loopback is that PRBS data cannot be looped back to the link partner.

Workaround

To characterize bit error rates without PRBS data, see the work around for “No PRBS checking.”

Status

10-bit loopback – Fixed in all parts with a date code of 1120 or later.

8-bit loopback – This loopback mode works with RapidIO compliant data.

JTAG 1149.1 chaining does not support register access

Description

The SPS-1616's JTAG functionality does not support configuration register access (CRA) when it is part of a JTAG chain. This issue does not impact JTAG boundary-scan testing (1149.1 or 1149.6).

Impact

The SPS-1616 must be the only device on the JTAG chain when its JTAG CRA operations are performed.

Workaround

Either one of the following work arounds can be implemented:

1. Design your hardware to allow independent JTAG access to SPS-1616's registers during board bring-up.
2. Access the device's registers from the RapidIO or I2C interfaces.

Status

There are viable work arounds for this issue.

Packet generator/checker restrictions

Description

The SPS-1616 packet generator will send four additional zero characters at the end of any generated packet. The EOP and SOP bits in the Port n Packet Generation and Capture Mode Configuration Register are always zero.

Impact

Endpoints targeted by packet generator request packets that are longer than expected may detect a logical layer error on the packet. It is not possible to determine where a captured packet ends.

Workaround

Send only one packet from the packet generator. If the packet to be captured is the same as the packet sent, there is no need to check for EOP. If the packet captured is different from the packet sent, the format of this packet must have a predictable end in order to determine if the correct packet has been captured.

Status

There is a viable work around for this issue.

Incorrect MCAST_MASK field

Description

The MCAST_MASK field value in the Switch Multicast Information CAR is consistent with the total number of multicast masks found in all ports (40 * N ports), instead of the number of multicast masks that can be accessed through the RapidIO standard registers (40).

Impact

Software may attempt to program non-existent multicast masks resulting in a configuration error.

Workaround

Software should use the number of global masks, 40, instead of the value in this register field.

Status

There is a viable work around for this issue.

Extra field in Lane n Status 0 CSR

Description

The Lane n Status 0 CSR has a bit, LP_RX_TRAINED, which does not exist in the *RapidIO Specification (Rev. 2.1)* Part 6 definition of this register. As a result, the ERR_8B10B, RX_SYNC_CHG, and RX_TRAINED_CHG fields are shifted by one bit compared to the definition in the specification.

Impact

Software may misinterpret the contents of the register.

Workaround

Software should use the actual SPS-1616 field locations instead of those defined in the *RapidIO Specification (Rev. 2.1)*.

Status

There is a viable work around for this issue.

Ports fail to initialize

Description

An S-RIO port may be unable to initialize properly after a reset; however, there is a much lower likelihood that a 1x port will fail to initialize. The affected port may operate at less than maximum width or fail to achieve PORT_OK status.

An S-RIO port will fail to train on a SPS-1616 approximately every 10 resets; however, the likelihood varies from device to device, and from port to port on the same device.

Impact

The link operates at less than expected capacity or packet exchange is not possible.

Workaround

To ensure consistent port initialization, program the default value of the following registers in the order given (the order of lanes does not matter).

Register Base Address	Offset	New Default Value[a]	Description
0xF20010	0x0	0xFABADC05	This provides access to a test register.
0xF18008 + Lane * 0x100	0x8	0x00000240	This changes the default value of the test register for all lanes (0x0–0xF).
0xF20300	0x0	0x8000FFFF	This issues a port reset for the change to take effect: it will reset all ports.

[a] These changes can be made transparent by using the Serial EEPROM to load the registers upon power-up. For the binary file and EEPROM programming instructions, please contact IDT Sales.

A port may not achieve PORT_OK status even if this work around is implemented; however, the failure rate is expected to be a very low percentage for 1x ports.

If a PORT_OK status is not achieved, the failing port must be reset. To reset the failing port, write 0x8000xxxx: (x = bbbb) to offset 0xF20300 (Device Reset and Control Register), where “b” is the bit corresponding to the failing port. If a PORT_OK status is not achieved, the process must be repeated.

Status

Fixed in all parts with a date code of 1120 or later.

Port cannot send packets

Description

Any time during operation, S-RIO ports that are in 4x or 2x mode may prevent packet exchange even though they achieved PORT_OK status.

Impact

The link may be inoperable or downgraded to 1x mode.

Workaround

Either one of the following work arounds can be implemented:

1. Operate the affected link in 1x mode.
2. Reset the SPS-1616. Additional resets may be required if the port fails to exchange packets again.

Status

Fixed in all parts with a date code of 1120 or later.

Dynamic control of link partner transmit emphasis not working correctly

Description

The “Dynamic Control of Link Partner Transmit Emphasis” feature does not operate correctly. Enabling this feature will result in a bit error rate higher than 10^{-12} and possible link failures.

Impact

Another method must be used to set the link partner transmit emphasis values.

Workaround

Either one of the following work arounds can be implemented:

1. Statically configure the transmitter characteristics.
2. Change link partner’s transmitter characteristics using the software control of link partner transmit emphasis.

Status

There are viable work arounds for this issue.

Maintenance packet buffer management

Description

A system that has one of the following traffic patterns may experience deadlocks due to maintenance buffer management defects:

- A system consisting of two endpoints, X and Y, connected to switch A, has a traffic pattern as follows:
  - Endpoint Y sends at least 37 non-maintenance requests with priority 1 to endpoint X, which results in responses with a priority of at least 2
  - Endpoint X issues 14 or more outstanding maintenance reads/writes of priority 0 for switch A's registers
  - Endpoint X issues 20 or more outstanding maintenance reads/writes of priority 2 for switch A's registers after the priority 0 requests have been issued
- A system consisting of two endpoints, X and Y, and one switch A, has a traffic pattern as follows:
  - Endpoint X connected to switch A has at least 25 maintenance transactions outstanding to endpoint Y
  - Other traffic destined for endpoint Y has the same priority as the maintenance requests
- A system consisting of two endpoints X and Y, and two switches A and B, where endpoint X is connected to switch A and endpoint Y is connected to switch B, has a traffic pattern as follows:
  - Endpoints connected to switch A have at least 22 maintenance transactions outstanding to endpoint Y
  - Endpoints connected to switch B have at least 22 maintenance transactions outstanding to endpoint X

Impact

No traffic in the dependency loops described can make forward progress. This results in a cascade congestion failure of the system.

Workaround

Either one of the following work arounds can be implemented:

1. All of these situations are avoided if the following system design rules are used:
   a. Only one endpoint issues maintenance requests into a system at any given time
   b. The maintenance requests are always sent using a single priority
   c. No more than 24 maintenance transactions are outstanding at a time
2. If it is not possible to follow the above system design rules, the Time-to-Live timer can be used to discard packets in the final buffer and break the deadlock. Note that if endpoints continue to send maintenance packets, then the deadlock may reappear immediately.
3. If the use of Time-to-Live is not an option, the RETRY_LIM field in the Port n Congestion Retry Counter Register can be used to trigger packet discard after receiving too many retries. The value of RETRY_LIM is system specific.

   Note: Packets sent at a higher priority than the maintenance packets involved in the deadlock may still be transferred, which will reset the retry counter. As a result, RETRY_LIM is not a deterministic work around for all systems.

Status

There are viable work arounds to this issue.

Maintenance packets congest input buffer

Description

Maintenance packets of priority 2 or less can completely occupy the SPS-1616's input buffer, which is a violation of RapidIO deadlock avoidance rules. Any traffic patterns that require transmission of responses may deadlock when the input buffer is completely occupied by maintenance requests.

Impact

Traffic cannot make forward progress. This results in a cascade congestion failure of the system.

Workaround

Either one of the following work arounds can be implemented:

1. This erratum can be avoided by adhering to the following system design rules:
   a. Ensure that only one endpoint can issue maintenance requests into a system at one time.
   b. Always issue maintenance requests with priority 2.
   c. Ensure that no more than 11 maintenance transactions are outstanding at a time.
2. If it is not possible to follow the above system design rules, then the Time-to-Live timer can be used to discard packets in the SPS-1616's final buffer and break the deadlock. Note that if endpoints continue to send maintenance packets then the deadlock may reappear immediately.
3. If the use of the Time-to-Live timer is not an option, the RETRY_LIM field in the Port n Congestion Retry Counter Register can be used to trigger packet discard after receiving too many retries. The value of RETRY_LIM is system specific.

   Note: Packets sent at a higher priority than the maintenance packets involved in the deadlock may still be transferred, which will reset the retry counter. As a result, RETRY_LIM is not a deterministic work around for all systems.

Status

Fixed in all parts with a date code of 1120 or later.

Lane swapping may prevent port from achieving PORT_OK

Description

When lane reversal is used, the SPS-1616 does not support automatic downgrade or port width override of a 4x or 2x port (using Port n Control 1 CSR.PWIDTH_OVRD).

Impact

A 4x or 2x S-RIO port will not achieve PORT_OK when it encounters bit errors.

Workaround

Do not use lane reversal during layout of the board design.

Status

There is a viable work around for this issue.

Protocol error stops packet transmission

Description

If the SPS-1616 experiences congestion when using receiver-based flow control, the device may issue a retry control symbol when it has not received a packet after its link partner issues restart from retry. This is a protocol error whose impact is limited to error recovery. Also, if the link partner issues a packet retry when a packet is not outstanding (a protocol error by the link partner), the SPS-1616's transmitter may stop sending all packets. This event occurs only on this protocol error from the link partner concurrent with a small window of time for the SPS-1616.

Impact

Packets cannot make forward progress; however, no error is detected. The system may also congest and fail.

Workaround

Either one of the following work arounds can be implemented:

1. Use transmitter-based flow control for all switch-to-switch links.
2. Engineer traffic patterns to avoid congestion-generated retries.

Status

Fixed in the Revision C parts.

IDLE2 defects

Description

When IDLE2 is enabled, the IDLE2 sequence and packet transmission may be corrupted.

Impact

RapidIO standard error recovery is invoked and packet transmission resumes. Performance is degraded and links may fail to train at the maximum width.

Workaround

Disable IDLE2. As a result of this work around, operation at 6.25 Gbaud lane rates is not possible.

Status

Fixed in the Revision C parts.

False idle character in packet detection

Description

When the SPS-1616 issues a retry, the link partner may respond by sending a Restart-From Retry control symbol immediately after transmitting a Start-of-Packet control symbol. When IDLE1 characters follow the Restart-From-Retry, the SPS-1616 falsely detects a packet corruption, which is captured in Port n Implementation Specific Error Detect Register[IDLE_IN_PKT].

Impact

RapidIO standard error recovery is invoked, and packet transmission resumes.

Workaround

To work around this issue, please choose one of the following approaches:

1. If the performance impact does not affect correct system operation, ignore the issue.
2. Use transmitter-based flow control for switch-to-switch links. Note that the use of transmitter-based flow control may impact performance to a greater or lesser degree than this issue.
3. Engineer traffic patterns to avoid congestion-generated retries.

Status

There are viable work arounds for this issue.

Improper auto-negotiation of IDLE2 to IDLE1

Description

A port may not properly auto-negotiate the use of an IDLE2 to IDLE1 control symbol when experiencing a bit error rate on the link of greater than 10^{-12}.

Impact

The link partner may experience a high number of Packet-Retry Control Symbols.

Workarounds

Implement the work around applicable to the connection:

1. Enable IDLE2 for all switch-to-switch connections.
2. Disable IDLE2 for all IDLE1-only endpoint connections.

Status

There are viable work arounds for this issue.

Unable to downgrade in 2x mode when primary lane is disabled

Description

When lanes 2, 6, 10, and 14 are mapped to a 2x port, and a fault occurs on the primary lane of a port, the port will not downgrade correctly. For example, if lane 2 (the primary lane) experiences a fault then the port will not downgrade to 1x on lane 3 (the redundant lane).

Impact

A 2x port may not train if it is downgraded when a primary lane is disabled.

Workarounds

Either one of the following work arounds can be implemented:

1. Use the primary lanes for all connections when a port is configured in 2x mode, and leave the redundant lanes unconnected.
2. Do not disable the primary lanes when a port is configured in 2x mode.

Status

There are viable work arounds for this issue.

Unable to downgrade when primary lane is inoperable

Description

A port may not downgrade properly from 2x or 4x mode to its redundant 1x lane.

Impact

The affected port may not train if downgraded while the primary lane is inoperable.

Workaround

Configure the port to operate in 1x mode.

Status

There are viable work arounds for this issue.

Packet with unexpected ackID is incompletely captured

Description

According to the *RapidIO Specification (Rev. 2.1)*, the first 16 bytes of a packet with an unexpected ackID are captured in the Port n Capture 0–3 CSRs. The SPS-1616 captures the first 4 bytes of a packet with an unexpected ackID in the Port n Capture 0 CSR; however, Port n Capture 1–3 CSRs contain undefined information.

Impact

Software cannot use the contents of the Port n Capture 1–3 CSRs to diagnose packets with unexpected ackIDs.

Workaround

None.

Status

No plans to fix.

Low throughput with maximum sized packets

Description

The maximum size of a RapidIO packet is an NWRITE transaction with 256-byte data payload, 16-bit device IDs, and 66-bit extended addressing. When this type of packet is sent back to back, the SPS-1616 falsely detects a “packet too large” error and initiates error recovery.

Impact

Low throughput.

Workaround

Do not use NWRITE transactions with 256 byte data payload, 16-bit device IDs, and 66-bit extended addressing.

Status

There are viable work arounds for this issue.

False unexpected ackID error detection

Description

When the Port n Local ackID CSR is used to recover a failed link, a false error indication bit named UNEXP_ACKID may be set in the Port n Error Detect CSR and the Port n Implementation Specific Error Detect Register.

Impact

Software cannot use the contents of Port n Local ackID CSR, Port n Error Detect CSR, or Port n Implementation Specific Error Detect Register, to diagnose acknowledge control symbols with unexpected ackIDs.

Workaround

Configure software to ignore the UNEXP_ACKID error detect bit.

Status

There is a viable work around for this issue.

False loss of descrambler sync error detection

Description

When a port is configured in 4x mode using an IDLE2 control symbol, and it is connected to a link partner in 1x mode, the unconnected lanes can falsely detect DESCRAM_SYNC errors.

Impact

It can prematurely invoke system software error recovery if error detection is enabled. Software cannot use the contents of the Lane n Error Detect Register to diagnose descrambler synchronization issues.

Workaround

Either one of the following work arounds can be implemented:

1. Disable the unconnected lanes of the 4x port.
2. Configure the 4x port in 1x mode.

Status

There are viable work arounds for this issue.

PORT_OK may indicate incorrect status

Description

When two connected S-RIO ports reach PORT_OK status and while one of the ports is subjected to the following behavior, the PORT_OK bit may remain set if software performs one or more of the following:

- Disables the port using PORT_DIS in the Port n Control 1 CSR
- Sets the SerDes Transmit drive strength to 0 using TX_AMP_CTL in the Lane n Control Register
- Powers down the port

Impact

The S-RIO port may continue to report PORT_OK status. RapidIO standard error recovery procedures may time out, causing a PORT_ERR condition to be detected.

Workaround

Change the link speed by completing the following in the Lane n Control CSR prior to performing a port disable. Upon port disable, ensure TX_RATE is set back to the intended speed prior to enabling the port.

- If the TX_RATE bit is greater than 0b01 then set it to 0b00; otherwise, set the bit to 0b11.

Or, perform either one of the following actions:

1. Perform a FORCE_REINIT to ensure the PORT_OK bit is correct.
2. Use disable input/output port, port reset, and enable input/output port instead of disable/enable port.

Status

There is a viable work around to this issue.

Receiving port’s ackID not properly reset upon receipt of link-request/reset-device control symbol

Description

The SPS-1616 may not exit the fatal error state upon receipt of a port reset from its link partner when the SPS-1616 is configured to drain its final buffer: Device Control 1 Register.FATAL_ERR_PKT_MGT = 0 (Drop: default). As a result, the SPS-1616’s output port cannot transmit packets and it will not recover from subsequent port resets.

Impact

If the SPS-1616 receives a link-request/reset-device control symbol, which is often used after a hot swap event, the switch will not completely recover the port from the fatal error condition.

Workaround

Manually align the ackIDs between the SPS-1616 and its link partner, and create a condition whereby the logic can respond to port reset and clear the fatal error conditions so that traffic can resume.

For information on how to manually align the ackIDs, see the “Hot Extraction/Insertion” section of the *SPS-1616 User Manual*.

Status

There is a viable work around for this issue.

Clock frequency tolerance tighter than specification for 6.25 Gbaud

Description

The *RapidIO Specification (Rev. 2.1)* requires devices to support a difference in clock frequency of ±100 PPM. If different clocking sources are used for the SPS-1616 and its link partners, and 6.25 Gbaud operation is used, then the SPS-1616 and its link partners require a frequency accuracy of ±50 PPM.

Impact

- When the link is idle, few errors will be detected.
- When packets are being exchanged, errors will be detected in one direction on a link, ranging from packet CRC errors to protocol/transmission errors.
- Possible link reinitialization or loss of lane alignment or lane sync

Work around

Use clock sources that achieve a frequency accuracy of ±50 PPM.

Status

Viable work around.

Subset of JTAG commands power down SerDes PLL

Description

The following JTAG instructions cause the SPS-1616's SerDes PLL to power down:

- 0x0 – EXTEST
- 0x1 – SAMPLE_PRELOAD
- 0x2 – IDCODE
- 0x3 – HIGHZ
- 0x4 – CLAMP
- 0x5 – EXTEST_PULSE
- 0x6 – EXTEST_TRAIN
- 0x7 – RESERVED

The SerDes PLL will remain powered down until assertion of JTAG reset, TRST_N.

Impact

The S-RIO links will be non-operational. Development of system-level software using JTAG emulators may not be possible. (Note: The RapidFET-JTAG tool is not impacted by this issue.) This issue does not impact 1149.6 SerDes boundary scan testing.

Workaround

Do not place the SPS-1616 in a JTAG chain shared with other devices if non SPS-1616 specific JTAG tools are used.

Status

There is a viable work around for this issue.

False BAD_PORT error detection

Description

A configuration error, BAD_PORT, is detected in the Configuration Block Error Detect Register whenever the following valid values are written to the Device Route Table Register {0..255}:

- 0x40–0x67 (Multicast masks 0–39)

Impact

A false error recovery mechanism is invoked, if enabled.

Workaround

Disable the capture of the BAD_PORT error by setting BAD_PORT_EN to 0 in the Configuration Block Error Capture Enable Register.

Status

There is a viable work around for this issue.

Writing Routing Table Entries Causes Dropped Packets

Description

Assume that the SPS-1616 uses routing table entry “X” to route packets. An application is also writing routing table entry “X”.

Impact

Packets routed using routing table entry “X” may be dropped. A packet may be dropped whether or not the routing table entry “X” value is changed by the write.

Workaround

There are three workarounds for this issue:

1. Temporarily change the routing table entry used to route the packets to allow the original routing table entry to be updated.
   a. Set up the temporary route using the temporary routing table entry.
   b. Change the traffic to use the temporary route.
   c. Ensure that no packets that use the original routing table entry exist in the system by sending a low priority transaction which requires a response. The low priority transaction will push all packets ahead of it, guaranteeing the original routing table entry is now unused. The response will confirm that the low priority transaction has been completed.
   d. Update the original routing table entry.
   e. Test the original routing table entry.
   f. Change the traffic to use the original route.
   g. Ensure that no packets which use the temporary routing table entry exist in the system.
2. Only write routing table entries that are not in use.
3. Temporarily stop accepting packets so that all routing table entries are not used.
   a. Clear the INPUT_PORT_EN and/or OUTPUT_PORT_EN control bits in the Port x Control 1 CSR.
   b. Write the routing table entry.
   c. Restore the INPUT_PORT_EN and/or OUTPUT_PORT_EN control bits.

Status

There are viable work arounds for this issue.

Revision History

Revision Date	Description of Change
April 4, 2019	▪ Updated the status of Protocol error stops packet transmission and IDLE2 defects
June 23, 2017	▪ Added Writing Routing Table Entries Causes Dropped Packets
March 23, 2016	▪ Added False BAD_PORT error detection
November 7, 2014	▪ Updated the Impact section of Dynamic control of link partner transmit emphasis not working correctly ▪ Updated the Description and Impact sections of Subset of JTAG commands power down SerDes PLL
October 30, 2012	▪ Added Subset of JTAG commands power down SerDes PLL
May 4, 2012	▪ Updated the work around for False idle character in packet detection ▪ Added a new issue, Clock frequency tolerance tighter than specification for 6.25 Gbaud
January 18, 2012	▪ Added Receiving port's ackID not properly reset upon receipt of link-request/reset-device control symbol
November 9, 2011	▪ Updated the status of IDLE2 defects and Protocol error stops packet transmission
October 12, 2011	▪ Added PORT_OK may indicate incorrect status ▪ Updated the definition of Improper auto-negotiation of IDLE2 to IDLE1 ▪ Updated all references to field and register names to align with recent changes to the *SPS-1616 User Manual*
August 31, 2011	▪ Revised the definition of Maintenance packet buffer management ▪ Added Maintenance packets congest input buffer ▪ Changed the status of Unable to downgrade in 2x mode when primary lane is disabled, Unable to downgrade when primary lane is inoperable, and Low throughput with maximum sized packets

Corporate Headquarters
6024 Silver Creek Valley Road
San Jose, CA 95138 USA
www.IDT.com

Sales
1-800-345-7015 or 408-284-8200
Fax: 408-284-2775
www.IDT.com/go/sales

Tech Support
www.IDT.com/go/support

DISCLAIMER Integrated Device Technology, Inc. (IDT) and its affiliated companies (herein referred to as “IDT”) reserve the right to modify the products and/or specifications described herein at any time, without notice, at IDT’s sole discretion. Performance specifications and operating parameters of the described products are determined in an independent state and are not guaranteed to perform the same way when installed in customer products. The information contained herein is provided without representation or warranty of any kind, whether express or implied, including, but not limited to, the suitability of IDT's products for any particular purpose, an implied warranty of merchantability, or non-infringement of the intellectual property rights of others. This document is presented only as a guide and does not convey any license under intellectual property rights of IDT or any third parties.

IDT's products are not intended for use in applications involving extreme environmental conditions or in life support systems or similar devices where the failure or malfunction of an IDT product can be reasonably expected to significantly affect the health or safety of users. Anyone using an IDT product in such a manner does so at their own risk, absent an express, written agreement by IDT.