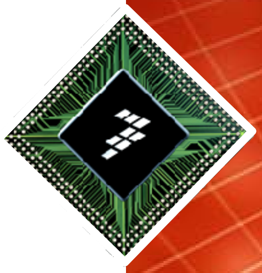
P1010 Errata



Confidential and Proprietary

A-006262 Wrong detection window in UTMI logic for sampling disconnect output

- **Description:** Based on the USB 2.0 spec, the disconnect envelope detector's output must be sampled once during transmission of the last 8 bits of the SOF EOP period at the transceiver output. Due to this erratum, the disconnect detection window is shifted to the left by 10 bits. These disconnect issues only apply to a USB host controller and only apply to the parts with an internal USB PHY.
- **Impact:** It may affect the length of the USB cable.
- **Workaround:** Do not use a USB cable longer than 4 meters.
- **Fix plan:** No plans to fix

eTSEC 11 No parser error for packets containing invalid IPv6 routing header packet

- **Description:** As part of the pseudo-header calculation for the L4 checksum, the controller uses the last destination address from the routing header. It then calculates the L4 checksum and leaves the ETU bit in the RxFCB clear (marking this as a good checksum.) Since the above conditions constitute an invalid IVp6 routing packet, the checksum should not be marked as good and a parse error should be flagged instead.
- The logic will do the following:
  - Stop parsing the packet when the invalid IVP6 routing header is seen
  - CTU bit is clear
  - Indicate packet as bad to the filer via PID 1 bit 11. Note that no parser error will be indicated in the RxFCB (such as PER) or than that which is reported in the filer's PID 1 bit 11.
- **Impact:** A packet with invalid IPv6 routing header will not be flagged as parse error. L4 checksum result (such as ETU) may be invalid.
- **Workaround:** Use one of the following options:
  - Set up the filter to detect the invalid IPV6 routing header using PID 1’s bit 11.
  - If CTU bit is 0, consistency checks for IPv6 routing header packets must be performed in software, or skipped. If a packet does have a valid IPv6 routing header, then L4 checksum result, if enabled, can be considered as valid. Otherwise, software should consider the packet as malformed and should not use the packet's L4 checksum result stored in RxFCB.
- **Fix plan:** No plans to fix

A-004187 Ethernet controller may fail link initialization in SGMII mode

- **Description:** A key function of the Ethernet controller in SGMII mode is link initialization and auto-negotiation (AN), if enabled. The link initialization function does not always properly detect when the link is synchronized. The failure is dependent on physical design so not all products will see failures nor all ports on a device. However, if link initialization is successful the link will remain stable. Link initialization failure is indicated by TBI SR[Link Status] remaining cleared and additionally, if AN is enabled, TBI SR[AN Done] remaining cleared.
- **Impact:** During first initialization or after a loss/change of link event, Ethernet link initialization may fail in SGMII mode resulting in no transmission or reception of packets for the affected port. Failures vary from port to port and device to device because process, voltage, temperature, and frequency are all factors. Small variations in software initialization sequences also affect the failure mode. The failure only refers to the connection between the MAC and the link partner’s SGMII interface (not the local and remote link connection). For 2.5Gbps, the MAC currently only supports AN disabled mode (TBI CR[AN Enable]=0).

A-004187 Ethernet controller may fail link initialization in SGMII mode

- **Workaround**
- AN Enabled case (TBI CR[AN Enable]=1)
  - No guaranteed software workaround for the AN Enabled case (TBI CR[AN Enable]=1).
  - Use steps outlined in AN disabled case if AN Enabled fails.
- AN Disabled case (TBI CR[AN Enable]=0)
  - Please follow these recommended steps to properly establish the link with auto-negotiation disabled. This recommendation requires disabling AN on the SGMII interface between the MAC and the connected device.
  - Execute the following sequence after SerDes clocks are stable, the link partner is sending valid symbols, and prior to clearing IEVENT. Writes to TBI registers require MDC/MDIO accesses.
  - MAC TBI Configuration:
  - Write 0x0020 to TBICON (AN Sense disabled, Clock Select=1)
  - Poll MIIMIND[Busy]=0 (Confirm write cycle is complete)
  - Read TBICON (Confirm value 0x0020 was written)
  - Poll MIIMIND[Busy]=0 (Confirm read cycle is complete)
  - Write 0x8140 to TBI CR (TBI Phy Reset, AN disabled, Full Duplex)
  - Poll MIIMIND[Busy]=0 (Confirm write cycle is complete)
  - Read TBI CR (Confirm value 0x0140 remains after reset)

A-004187 Ethernet controller may fail link initialization in SGMII mode

- Connected Device (PHY, Switch/MAC, FPGA, etc.) Configuration:
- Please ensure AN is disabled for the SGMII-to-MAC interface and perform an interface reset. MII Management interface should be used for designs using the MDC/MDIO bus. If a PHY with AN ability is used, restart the AN on the link (CAT-5) side and then read the link status and link partner capabilities.
- Continued Initialization:
- When TBI SR[Link Status]=1, perform any other MAC related init/reconfiguration steps as needed along with normal programming of MACCFG2[I/F Mode] and ECNTRL for proper MAC speed and duplex configuration according to link partner capability. If TBI SR[Link Status] does not get set within a reasonable period of time, please repeat from step #5 for Rev 1.0; for Rev 1.1 there may be something else wrong with the link.

- **Fix plan:** No plans to fix

A-006461 USB high-speed compliance tests for HS Test J and Test K fail for unexpected VOL levels

- **Description:** During Test J and Test K compliance tests (EL_9), the VOL level of the high-speed transmitter should be ±10 mV; however, the silicon range can be up to 20 mV. This results in failure of Test J and Test K compliance tests.
- **Impact:** Because the silicon range is 10 mV greater than the 10 mV limit, the common mode of the USB transmitter output shifts 10 mV, which does not impact the USB functional performance because the receiver common mode range is up to 500 mV. There is practically no impact on the functional performance of the USB.
- **Workaround:** There is no work around for this errata. It is recommended to send a part that has a VOL level less than 10 mV in high-speed Test J and Test K compliance tests for external compliance certification.
- **Fix plan:** No plans to fix

A-003827 DATA PID error interrupt issued twice for the same high bandwidth ISO transfer

- **Description:** When receiving an Isochronous OUT transfer for a High Bandwidth endpoint (MULT > 0), if one of the DATA PIDs is corrupted, the controller issues two interrupts for that transaction error, one in the current microframe to signal the DATA PID error, and one fulfillment error in the next microframe.
- **Impact:** The only impact of this issue is that the application discards not only the data transaction for which the DATA PID error occurred but also the next transaction.
- **Workaround:** None
- **Fix plan:** No plans to fix

A-007075 USB host may not detect device connection

- **Description:** The root cause of this issue is that the controller deasserts its soft reset bit (USBCMD[RST]) based on internal counters that do not ensure the ULPI PHY has completed its reset sequence. If a command is sent to ULPI PHY when it is in the reset state, then that command can be corrupted by the ULPI PHY.
- It is seen on LA (Logical Analyzer), a command issued by the controller to write in OTGCONTROL(drivevbus bit) register is getting corrupted sometimes. So, the controller must not have deasserted its USBCMD[RST] bit before PHY reset completion. Thus, when we put a delay before writing in PP bit, it gives enough time to PHY to come out of reset and proper command reaches PHY.
- **Impact**A USB host may not be able to detect device connection. However, this issue rarely occurs (approximately once every 200 iterations).
- **Workaround:** There are two workarounds for this issue:
  - Before writing the Port POWER bit (PP) in the PORTSC register of the host controller, configure a delay of 5 ms so that the ULPI PHY has enough time to come out of reset.
  - Because the USB controller is not able to set the drivevbus bit of the OTGCONTROL register, we can force the drivevbus bit of ULPI PHY using the ULPIVIEWPORT register of controller. After writing PORTSC[PP] = 1, write the ULPIVIEWPORT register with 600b_0020h. This command will set the drivevbus bit in the OTGCONTROL register of the ULPI PHY.
- **Fix plan:** No plans to fix

SEC-A002 The SEC does not correctly use LIODN to classify jobs which can cause sharing-related LIODN error

- **Description:** The SEC internal job management logic should check both the descriptor address and LIODN before classifying a job as belonging to a flow for which other jobs are already resident in the SEC. However, due to this erratum, the SEC erroneously pre-classifies jobs with only matching shared descriptor addresses as belonging to a flow instead of using both the LIODN and descriptor address. In the job execution stage, if the SEC determines that a job's LIODN does not match the rest of the other jobs in the flow (i.e., the job is unrelated) and thus the jobs do not share descriptors, the SEC does not execute the job and reports an LIODN error (JRSTA = 0x4000_xx1F).
- Two or more jobs in the SEC's job queues may end up with the same descriptor address but different LIODNs in environments where SEC's PAMU is configured to perform address translation (e.g., to support two or more guest Operating Systems with overlapping virtualized physical address spaces).

- **Impact:** A job may be classified to an incorrect flow. The job will not be executed and will report an LIODN error via JRSTA=0x4000_xx1F.
- **Workaround:** Avoid the possibility that SEC encounters jobs associated with same shared descriptor address. There are several options to avoid this issue:
  - Do not utilize address translation in SEC's PAMU
  - Place shared descriptors in a memory region for which the SEC PAMU does not perform address translation
  - Ensure that no SEC-dependent partitions use PAMU-virtual address ranges that overlap
  - Ensure that no SEC-dependent partitions use shared descriptors having overlapping PAMU-virtual address ranges

- **Fix plan:** No plans to fix

A-004562 SEC single step mode fails in special cases

- **Description** When running in single step mode (used for debugging purposes), the register which tracks the end of the descriptor may get corrupted by the fifth word of the job descriptor, causing the descriptor size to be improperly updated.
- **Impact:** Low—Single step mode would fail as a function of the fourth word in the descriptor.
- **Workaround:** Every descriptor is made up of a HEADER (which indicates the descriptor type, size, etc) and then a list of commands (up to 64 words long). With this erratum, after the SEC DECO reads the descriptor size value from the header, a read of the 5th word in the job descriptor incorrectly changes the descriptor size value in the SEC registers.

  In order to avoid this descriptor size value getting incorrectly updated after already being read, place the following conditional JUMP command at the 5th word of the job descriptor:

  0xA0000C00.

  The DECO will know not to update the descriptor length as it processes this JUMP command workaround because:

  - The JUMP will not execute because this conditional JUMP command will evaluate as false. This specific conditional JUMP command is evaluating that a condition must be both "zero" and "negative", which is impossible.
  - The JUMP Command Type field (0xA) cannot be mistaken for a HEADER Command Type (0xB).

  Ensure that this JUMP command always is placed as the 5th word in the descriptor. In the case that multiple word commands span the 5th word of the descriptor, NOPs may be necessary. As an example, for a descriptor with a 2 word command at the 4th and 5th words, the programmer needs to insert a NOP at the 4th word and the 0xA0000C00 JUMP command at the 5th word, moving the two word command originally at the 4th and 5th words to the 6th and 7th words. For shorter descriptors, use NOP commands as needed in order to place the JUMP 4 words after the header.
- **Fix plan:** No plans to fix

A-004914 Debug Permissions Register writes take immediate effect

- **Description:** To program any of the security-related fuses, you must first write to the associated Secure Fuse Processor (SFP) mirror register and then program the fuses by writing to the SFP instruction register. However, writing the Debug Permission Register (DPR) takes immediate effect. As a consequence, you will not be able to write the SFP instruction register via a debug controller if you first program the DPR Debug Permission (DP) field to Closed (1xx).
- **Impact:** Cannot set DPR[DP] to 1xx via debug controller.
- **Workaround:** The SFP mirror registers and instruction register are still accessible through software with sufficient clearance. Trusted software can be used to both set DPR[DP] and program the fuses.
- **Fix plan:** No plans to fix

A-006879 SFP cannot be accessed while programming fuses

- **Description:** When the SFP is programming fuses, do not attempt to access any memory-mapped SFP facility. Any read during programming will invariably return 0s, meaning that INGR cannot be polled for a completed status. More importantly, accesses while programming may disrupt the programming operation. For this reason, do not access the SFP until programming is complete.
- **Impact:** User cannot poll the SFP instruction register to determine when fuse programming has completed.
- **Workaround:** SFP documentation recommends polling the instruction register to determine when programming has completed. Because this cannot be done, it is recommended to simply wait the maximum possible programming time before attempting to access the SFP again.
- For 45nm products(QorIQ P series), this time is (5 us prog/bit) x (2048 bits) = 10.24ms
- For 28nm products (QorIQ T series), this time is (12 us prog/bit) x (4096 bits) = 49.15ms
- **Fix plan:** No plans to fix

A-006022 PMGC0 and UPMGC0[TBSEL and TBEE] can not be read

- **Description:** PMGC0[TBSEL and TBEE] can be correctly written but will always return 0 when read. UPMGC0[TBSEL and TBEE] will also always contain 0. No other PMGC0 and UPMGC0 bits are impacted by this erratum.
- **Impact:** Software can correctly write PMGC0[TBSEL and TBEE] and they will function as specified. However, software must remember what values it wrote and not rely on the value it reads from PMGC0 or UPMGC0[TBSEL and TBEE].
- **Workaround:** None
- **Fix plan:** No plans to fix

eTSEC 3 Multiple BD frame may cause hang

- **Description:** In the “Transmit Data Buffer Descriptors (TxBD)” section of the device reference manual, it states the following:
    - Software must expect eTSEC to prefetch multiple TxBDs, and for TCP/IP checksumming an entire frame must be read from memory before a checksum can be computed. Accordingly, the R bit of the first TxBD in a frame must not be set until at least one entire frame can be fetched from this TxBD onwards. If eTSEC prefetches TxBDs and fails to reach a last TxBD (with bit L set), it halts further transmission from the current TxBD ring and report an underrun error as IEVENT[XFUN]; this indicates that an incomplete frame was fetched, but remained unprocessed.
    - If software sets up a frame with multiple BDs, and sets the first BD READY bit before the remaining BDs are marked ready, and if the controller happens to prefetch the BDs when some are marked ready and some marked unready, the controller may not halt or set IEVENT[XFUN], hanging the transmit.
- **Impact:** If software does not follow the guidelines for setting the ready bit of the first BD of a multiple TxBD frame, the Ethernet controller may hang.
- **Workaround:** Software must ensure that the ready bit of the first BD in a multiple TxBD frame is not set until after the remaining BDs of the frame are set ready.
- **Fix plan:** No plans to fix

A-006293 Mixing TOE = 0 and TOE = 1 frames may cause data corruption

- **Description:** eTSEC supports several TCP offload functions on transmitted frames, including IP checksum generation, TCP checksum generation, and VLAN tag insertion. If the controller is processing a mixture of frames with TxBD[TOE] = 1 and TxBD[TOE] = 0, it may use an incorrect Transmit Frame Control Block (TxFCB) for a frame and, therefore, incorrectly process the packet data.
  - The effects of incorrect TxFCB processing are as follows: TOE function executed on frame with TOE = 0.
  - TOE function not executed on frame with TOE = 1.
  - Different TOE function executed than intended (one or more fields of TxFCB perform an unintended action).
  - In most cases, this corrupts the frame in a way that is detected by the receiver (for example, checksum error). However, in some cases, the frame corruption may not be detected by the receiver.
- **Impact:** Mixing TOE = 1 and TOE = 0 frames may cause corrupted packets.
  - Whether a particular frame is affected is dependent on the TxBD[TOE] settings and sizes of the frame(s) preceding the frame in question, as well as the transient state of the internal Tx FIFO SRAM.
- **Workaround:** Do not mix TCP offload usage for frames. Write either TxBD[TOE] = 1 on all frames or TOE = 0 on all frames.
- **Fix plan:** No plans to fix

freescale™

A-005255 Failure to Detect Single SYNC Primitive

- **Description:** When a single SYNC primitive is sent between the WTRM and XRDY primitives, the host controller may fail to detect this primitive. This can cause the host controller link state machine to remain in the "ending" state. As a result, the controller will not be able to return to the "idle“ state, will not respond to the XRDY primitive stream and will not set corresponding command completion event.
- The SYNC primitive after detection in the Rx domain is normally stretched to two Rx clock pulses, the arrival of the XRDY primitive in the clock directly after the SYNC will cause the stretching effect to be nullified thus leading to a single SYNC detect pulse in the Rx domain being passed to the Tx domain. This is an unsafe structure and will lead to the non detection of the SYNC pulse in the Tx domain.
- **Impact:** SATA devices that send a single SYNC primitive after a read transaction may fail proper operation. However, there are many SATA devices will send more than one SYNC primitive after a read transaction and are not impacted by this erratum.
- **Workaround:** There is no workaround. If the device sends a single SYNC primitive and the host controller fails to detect it, the software will time out waiting for command completion event. To recover, the software can re-initialize the SATA Link by clearing HCONTROL[HC_ON] to 0x0 and then setting it to 0x1.
- **Fix plan:** No plans to fix

freescale™