S32E2_0P91J

Mask Set Errata

Rev. 3 — 1 July 2025 **Errata**

1 Mask Set Errata for Mask 0P91J

1.1 Revision History

This report applies to mask 0P91J for these products:

- S32E2

Table 1. Mask Specific Information

major_mask_rev_num	0x1
minor_mask_rev_num	0x0
jtag_id	0x0831_B01D, 0x0831_F01D

Table 2. Revision History

Revision	Release Date	Significant Changes
3	7/2025	The following errata were removed. • ERR051655 The following errata were added. • ERR052356 • ERR051779 • ERR052367 • ERR052368 • ERR052374 • ERR052376 • ERR052381 • ERR052384 • ERR052386 • ERR052558 • ERR052805 • ERR052501 • ERR052438 • ERR052453 • ERR052403 • ERR052262 • ERR052455 • ERR052288 • ERR011508 • ERR052663 • ERR052547 • ERR052004 • ERR052289 • ERR052290

Table 2. Revision History*...continued*

Revision	Release Date	Significant Changes
		• ERR052291 • ERR052465 The following errata were revised. • ERR051734 • ERR051617 • ERR051380 • ERR052026
October 2024	12/2024	The following errata were added. • ERR052226 • ERR052026 • ERR052350 • ERR052207 • ERR052243 • ERR052209 • ERR051597 • ERR052223 The following errata were revised. • ERR051476
March 2024	3/2024	Initial Revision

1.2 Errata and Information Summary

Table 3. Errata and Information Summary

Erratum ID	Erratum Title
ERR006425	LINFlexD: FIFO buffer can not be accessed without changing the RFC counter
ERR006992	PSI5: IS_DEBUG_FREEZE bit is not documented
ERR007274	LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state
ERR007996	PSI5: Incorrect SMC message decoding and timestamp generation in case of late last sensor message overlapping with next SYNC period pulse
ERR008075	PSI5-S: Unrecoverable messages are not flagged in the mailbox status register channel 0
ERR008141	PSI5-S: Global mode transition interrupt does not work
ERR009978	eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition
ERR011508	NETC: Transmission Overrun counter defined in IEEE 802.1Qbv is not implemented
ERR050142	ERM_AE: Correctable Error Count Register value is incorrect
ERR050167	eDMA_AE: Uncorrectable memory error in EDMA_AE TCD memory is ignored by EDMA_AE
ERR050366	AE_AHB: Back to back writes of less than 128-bits will corrupt memory
ERR050457	NoC: Trace probe packet counting with length filter may hold incorrect count value.
ERR050501	Core: DFSR.EXTERNAL is not set correctly when waking up from sleep
ERR050502	Core: Execution priority might be wrong for one cycle after AIRCR is changed
ERR050504	Core: Sorting of pending interrupts might be wrong when high latency IRQs are pending
ERR050505	Core: Access permission faults are prioritized over unaligned Device memory faults

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR050509	Cortex-R52 1328481-C The debugger view of the number of cores might be incorrect when switching between Split and Lock modes
ERR050679	NETC: Accesses to 64-bit stats registers must be performed atomically
ERR050786	Core: ATB flush response may be delayed
ERR050787	Core: Data ATB flush may not respond
ERR050956	CEVA_SPF2: OCEM ROM Table is not discovered by Coresight Table Walk
ERR051023	NETC: Excessive collisions are counted as late collisions
ERR051025	NETC: Preemption verify time longer than specified
ERR051034	NETC: HTA SIUNSBESR counter is not incremented
ERR051048	NETC: Management command with search action responds with incorrect NUM_MATCHED
ERR051051	Core: A partially completed VLLDM might leave Secure floating-point data unprotected
ERR051068	CEVA_SPF2: The DDMA incorrectly ends the ongoing IIT task
ERR051080	CEVA_SPF2: Access to the CPM region might not complete according to the AXI protocol
ERR051082	QMAN may set incorrect field in the QMAN_IDM_CROSS register
ERR051129	NETC: Errored frames due to parity error are not counted in TFCS and TERR
ERR051130	NETC: Egress time gate scheduling can get corrupted when functional level reset is applied or when time gating is disabled.
ERR051140	RTU.SRAMCTL: Low power feature cannot be used
ERR051146	FCCU: Back to back faults with functional reset reaction not working properly
ERR051153	FCCU: RTU Faults may get inadvertently set during self-test
ERR051162	I3C: Slave reset not supported when I3C is in slave mode
ERR051163	MC_RGM: Incorrect destructive reset event status
ERR051164	DDRC: Read Data Buffer SRAM Fault Reports Incorrect Device ID
ERR051183	MDM-AP: DTS trigger output cannot be generated inside the MDM-AP
ERR051188	NETC: Register SITSR not accessible for Virtual Station Interfaces
ERR051202	NETC: Possible transmit MAC underrun at low 10M/100M speeds when the NETC switch is operating in cut-through forwarding mode
ERR051218	DDRC: Some timings not properly derated at high temperature
ERR051241	eDMA4: Channel preemption feature can operate incorrectly
ERR051243	NETC: Multi-entry SEARCH table management command returns invalid error code.
ERR051245	NETC: PSIaCFGR1 and SIRBGCR registers are not reset on Function Level Reset (FLR)
ERR051246	NETC: Tx/Rx disable (POR[RXDIS] and POR[TXDIS]) are enabled out of reset
ERR051247	NETC: NETC has no mechanism for system bus errors from VSI initiated transactions to the PSI to notify the PSI of the bus error
ERR051251	DDRC: Violation of tRFCpb
ERR051254	NETC: Administrative gate control list can get configured inadvertently when an exception is detected and notified

Table 3. Errata and Information Summary...*continued*

Erratum ID	Erratum Title
ERR051260	NETC: During initial Initialization of NETC, all ENETCs and Switch PCI functions must be enabled for NETC to accept table management commands for any function
ERR051261	NETC: Egress ports can violate 802.3 half-duplex jam protocol after collision event
ERR051262	NETC: MAC statistics counters are inaccurate when operating in half-duplex
ERR051264	NETC: Transmitted verify mPacket for preemption verification can be corrupted in some rare retry scenarios
ERR051300	NETC: NETC engine watchdogs do not operate effectively under all circumstances
ERR051305	NETC: Entry_id not returned correctly by Stream Gating Instance and Time Gate Scheduling QUERY table management commands
ERR051312	SPF2: RAW match is not detected for a load
ERR051325	eDMA4: The feature of cancelling the remaining data transfer is not working effectively
ERR051326	eDMA4:TCDn_CSR[ESDA] and TCDn_CSR[EEOP] are not functional
ERR051327	eDMA4:Scatter-gather feature does not work when NBYTES is not a multiple of 8 bytes
ERR051336	eDMA4: Swap feature with 8-bit swap size and 16-bit transfer size does not work
ERR051338	NETC: Table 16 (L2 IPv4 Multicast Filter) returns inaccurate KEY_TYPE field
ERR051364	NETC: Cut-through only frames incorrectly incrementing drop counts related to MSDU check
ERR051367	NETC: UDP checksum bytes may be corrupted in transmitted PTP packets with one-step timestamping enabled
ERR051380	LFAST: LFAST PLL Start-up Marginality
ERR051398	NETC: FLR or transmit disable may cause frame transfers to underrun in MAC resulting in bad frame transmission
ERR051400	NETC: Energy Efficient Ethernet (EEE) not supported in MII mode
ERR051401	NETC: Corrupt packets may be transmitted after receiving verify mPacket for preemption capability verification
ERR051402	NETC: MAC Merge stat inaccurate after lost Rx fragments
ERR051411	NETC: Time Gate Scheduling (IEEE 802.1Qbv) for MII half-duplex operation
ERR051413	Precise time-specific-departure operation requires zero_lookahead=0
ERR051432	eDMA4: DMA channel may not terminate correctly if a bus error occurs during transfer.
ERR051441	RTU: Privileged accesses to the peripherals cannot be controlled by the XRDC MDACs
ERR051444	NETC : ICM discards when two priority queues are in use can cause potential memory loss
ERR051449	FlexCAN: Node (connected with CAN HUB) accepting messages without TX ACK
ERR051457	GTM: ARU data trace message does not differentiate between channel 0 and channel 1
ERR051460	NETC: Ingress Port Filter (Table ID 13) can overflow rsp buffer during multi-entry query management command
ERR051476	CANEXCEL: Performance Limitation on Reception side
ERR051484	AES_ACCEL: DMA2_DID_ERR incorrectly asserted
ERR051485	Key property flag DEBUG should not be used
ERR051487	CANEXCEL: Data RAM Error Report Syndrome does not function properly
ERR051488	CANEXCEL: Keep latest feature in RXFIFO does not work

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR051489	CANEXCEL: When a Message Descriptor is locked by the system, the State transition to OVERRUN does not function properly
ERR051495	NETC: Stale context in policer can result in Policer dropping frames
ERR051524	NETC: Ingress Stream Identification Payload construction evaluates incorrectly for frames >1kB
ERR051530	NETC: hash multi-entry search table management continuation command can miss returning entries
ERR051549	CANEXCEL: Message descriptor overrun error bit does not function properly
ERR051551	CANEXCEL: Access to register CANXL_TBS is not protected by PDAC
ERR051557	ATP: Possible Aurora PLL jitter spec violation due to VDD_IO_A_J_POR edge aligned activity
ERR051561	CMU: Automatic Fault Injection mode does not work properly
ERR051587	NETC: Time gate scheduling update command response can be erroneous if the AdminBase Time specified is near the current time
ERR051589	LMEM64: CM33 core can have coherency issues when accessing the LPDDR region
ERR051590	CSTCU/RGM: Destructive reset during STCU self-test causes extraneous FES[ST_DONE] assertion
ERR051591	CM33: Bus faults are not reported in some cases when LMEM cache is enabled
ERR051597	CM33: MCM_FATR detects wrong attributes in cache write buffer error
ERR051607	NETC: NETC PF may override VF bus master enable and MSI-X function mask for MSI-X generation
ERR051609	I3C: May be unreliable in I3C mode.
ERR051613	RTU: GICR_TYPER registers do not reflect affinity configuration inputs
ERR051614	RTU: DTR flags not cleared on external debugger access while leaving Debug state
ERR051616	NETC: FDB (Table ID 15) always overrides Egress Treatment Table access defined by Ingress Stream (Table ID 31)
ERR051617	[I3C] In I2C controller mode generates unintended repeated START before sending STOP
ERR051620	GTM: Interrupt trigger signals CCU0TC_IRQ and CCU1TC_IRQ are delayed by one CMU_CLK period related to the output signals.
ERR051622	NETC: TABLE_BIR of NETC TIMER is wrong
ERR051630	GTM: ((A)TOM) Changing the output signal level synchronously to the period by writing to (A)TOM[i]_CH[x]_CTRL_SR.SL_SR for 0% duty cycle with (A)TOM[i]_CH[x]_CM0.CM0=MAX +1 and (A)TOM[i]_CH[x]_CM1.CM1=0 and (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1 not functional
ERR051631	GTM: (TIO) Interrupts are not cleared by a write access to interrupt mode register
ERR051632	GTM: (ARU) Delivering data via ARU_ACCESS does not take into account the status of the ARU cluster isolation from the requesting module
ERR051633	GTM: (AXIS) Bridge software reset initiated by writing BRIDGE_MODE.BRG_RST=1 will corrupt AXIS slave protocol
ERR051634	GTM: (DPLL) Wrong calculation of pulse generator frequency when number of pulses is too small.
ERR051635	GTM: (GTM_AEI) Write transaction in Split Mode to BRIDGE_MODE register may never terminate

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR051636	GTM: (DTM) HRES output calculation for short input pulses and enabled dead time incorrect
ERR051637	GTM: (MCS) Invalid instruction trace output
ERR051638	GTM: (MCS) Unexpected repeated break point action
ERR051648	SPI: In Continuous Selection Format observed tASC timing differs from the expected one.
ERR051649	NETC: Switch ports support only one ‘802.1p and DEI to internal QoS’ mapping profile
ERR051651	NETC: Use of 10Mbps restricted to PHYs that support octet alignment for preamble with minimum 1B
ERR051663	DDRC: Memory Select Error Can Cause DDRC Failure
ERR051666	GTM: Unexpected write access before instruction break point
ERR051667	GTM: Potential invalid bit field STA.Z for parallel memory read access
ERR051668	GTM: Potential wrong data for register MHB during parallel memory read access
ERR051669	GTM: AEI bridge might not execute an accepted transaction after a software reset.
ERR051673	GTM:Potentially invalid MCS data trace output for parallel memory access.
ERR051674	GTM: Potential wrong data in memory during parallel memory write access while storing the register STA to memory
ERR051675	GTM: Soft reset might not be triggered when issued through Pipeline-Mode
ERR051676	GTM: Unexpected stack pointer decrement
ERR051677	GTM: Unified error behavior
ERR051678	GTM: Missing edge on output signal ATOM/TOM_OUT when CN0 is reset with force update event
ERR051679	GTM: Flags of registers DPLL_STA_FLAG are not set
ERR051680	GTM: The AEI bridge might not execute an accepted write transaction
ERR051681	GTM: Change of the BRIDGE_MODE register might be delayed indefinitely
ERR051682	GTM: Internal compare events are not cleared
ERR051698	CTU : Double buffer reload mechanism is blocked when master reload pulse is not generated by Software
ERR051701	SPI: De-asserting of CONT bit in the Continuous Selection Format is not mandatory for the last frame
ERR051734	Core: DWT comparator match on cycle count is not reported to the ETM if there is no instruction executing on the processor
ERR051777	DMA_CRC: TCDs with a non-zero SMOD attribute lead to invalid checksum calculations if CRC is enabled on the channel
ERR051779	GTM: (TIO) Buffer empty signalization (PL_EVT) is not correct
ERR051780	GTM: Wrong AEI status on accesses to ICM registers while cluster 0 clock is switched off
ERR051781	GTM: (TIM) Potentially wrong capture values
ERR051782	GTM: (AEI) Changing BRIDGE_MODE.MSK_WR_RSP in Pipeline mode can lead to violation of Pipeline protocol
ERR051783	GTM: (DPLL) Wrong DPLL_RDT_S_ACT/DPLL_RDT_T_ACT value in case of overflow correction

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR051784	GTM: (MCS) Unexpected Memory overflow when using RET instruction
ERR051785	GTM: (MCS) Missing Memory overflow detection on literal addresses
ERR051786	GTM: (MCS) Instructions BRDI and BWRI evaluate unused address bits
ERR051787	GTM: (MCS) Unexpected Break Point initiation
ERR051788	GTM: (MCS) Missing resume from Hardware Break Point
ERR051789	GTM: (DPLL) Pulse correction is executed twice
ERR051790	GTM: (TIM) Missing glitch detection interrupt event
ERR051791	GTM: (TIM) Unexpected increment of filter counter
ERR051792	GTM: (TIM) Glitch detection interrupt event of filter is not a single cycle pulse
ERR051793	GTM: (TIO) Compare mode: Output value not correct if two events occur at the same time
ERR051794	GTM: (TIO) Incorrect behaviour on O_OUT in case of consecutive compare events
ERR051796	GTM: (DPLL) Doubled pulse correction
ERR051797	GTM: (TIO) In dual compare mode the initialization of internal compare event enables is not correct
ERR051798	GTM: (TIO) In dual compare mode the update of internal compare event enables is incorrect
ERR051799	GTM: (TIO) Wrong output value on O_OUT in capture mode
ERR051800	GTM: (MCS) Unexpected data break point behavior
ERR051801	GTM: (DPLL) Missing pulse correction in case of DPLL_CTRL_1.SMC=1
ERR051804	GTM: (MCS) Unexpected state of register MCS[i]_HBP[k]_STATUS
ERR051805	GTM: (TIO) Erroneous assertion of PL_EVT when channel is configured as a cyclic buffer
ERR051806	GTM: (DPLL) Stored time stamp values do not consider filter delays
ERR051807	GTM: (TIO) Incorrect initialization of internal compare event enables on cancel trigger
ERR051808	GTM: (TIO) Cancel trigger in single compare mode does not suppress action on O_OUT
ERR051809	GTM: (DPLL) No action calculation
ERR051810	GTM: (DPLL) Missing TOR interrupt and status flag
ERR051811	GTM: (MCS) Missing data break point for parallel memory access
ERR051812	GTM: (DPLL) DPLL_PVT not cleared after direction change
ERR051813	GTM: ((A)TOM) Missing CCU0TC_IRQ interrupt signal for UDMODE>0
ERR051814	GTM: (TOM) Unexpected behaviour of TOM_OUT_T for UDMODE>0
ERR051815	GTM: (MCS) Some internal MCS registers are not controlled by GTM halt logic
ERR051816	GTM: (MCS) Write access to protected bit field reports unexpected AEI status
ERR051817	GTM: (MCS) Missing trace information of WUCE instruction
ERR051821	GTM: (DPLL) DPLL_PSTC, DPLL_PSSC erroneously modified.
ERR051822	GTM: (DPLL) Incorrect values of DPLL_RCDT_TX, DPLL_RCDT_SX
ERR051823	GTM: (MCS) Unexpected instruction execution while disabling of MCS channel
ERR051824	GTM: (DPLL) DPLL_DCGI interrupt not triggered
ERR051825	GTM: (DPLL) Incorrect calculation of DPLL_THVAL, DPLL_THVAL2

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR051832	GTM: (GTM_AEI) Turning off BRIDGE_MODE.MSK_WR_RSP in asynchronous mode might lead to following transactions being corrupted
ERR051833	GTM: Interrupt from DPLL not detected in MCS0
ERR051834	GTM: (DPLL) Wrong value of DPLL_INC_CNT1.INC_CNT1 upon switching to normal mode
ERR051835	GTM: (MCS) Missing hazard detection for parallel memory write access
ERR051848	S32Z2E2: Possible Core and Peripheral PLL jitter spec violation due to edge-aligned I/O activity
ERR051887	NETC: VLAN qualifiers missed in ingress classification lookups
ERR051964	CEVA_SPF2: Vector store instruction data loss
ERR052004	STCU: MBIST reports single bit errors as failures
ERR052024	NETC: Half duplex transmit throughput degraded by up to 2.3%
ERR052026	CAAM: AES-GCM may generate an incorrect MAC
ERR052031	NETC: PTCaTSDR registers are implemented in the wrong order within the memory map
ERR052097	CANEXCEL: NO Underrun reporting for RX MD with match if KL=0
ERR052126	Arm Errata 2918152: [Cortex-R52] Executing LDM instruction might cause data corruption when HSCTLR.FI is set
ERR052133	NETC: MAC does not detect mVerify after start fragment
ERR052207	NETC: SG_DROP_COUNT value in the Ingress Stream Count STSE_DATA response begins at an incorrect bit offset, causing it to be read incorrectly.
ERR052209	CMU: AutoFI (Fault Injection) fixed mode is not executed after running Standalone Fixed FI mode with mask enable bit set.
ERR052223	TMU: Temperature monitor averaging quantization error
ERR052226	SWT: Toggling watchdog enable may cause unexpected timeout in some boundary conditions
ERR052243	TMU : Unexpected temperature readings during monitoring mode
ERR052262	NETC: Minimum-size frames are not preempted
ERR052288	NETC: The first frame after SequenceRecoveryReset (FRER) is dropped and miscounted
ERR052289	CANEXCEL: Unexpected Incorrect Configuration error detected during TX/RX MD Pointer Configuration
ERR052290	CANEXCEL: Incorrect RX-SMB Overrun reporting in case of XL frame with DLC less than 64.
ERR052291	CANEXCEL: Data Inconsistency observed between transmitted and received frame in case of LOA
ERR052350	CANEXCEL: MSG_DESCRIPTORS.TXCTRL_n[PRIO] > 3 is not considered in case of re-insertion.
ERR052356	CANEXCEL: Soft-Reset limitation
ERR052367	GTM: (MCS) Unexpected parallel memory access after MCS error
ERR052368	GTM: (MCS) Unexpected parallel memory access on ECC error
ERR052374	GTM: (DPLL) Unregular pulse generation and wrong PMT results
ERR052376	GTM: ((A)TOM) Missing edge on output signal (A)TOM_OUT
ERR052381	GTM: ((A)TOM) Edge at output signal (A)TOM_OUT does not occur

Table 3. Errata and Information Summary*...continued*

Erratum ID	Erratum Title
ERR052384	GTM: CPU bus access is not acknowledged
ERR052386	GTM: (MCS) No resume action after clearing hardware breakpoints
ERR052403	FlexCAN: CAN frame drops in Enhanced RX FIFO when message buffer (MB) is locked for more than 1 CAN frame time (33 us)
ERR052438	FlexCAN: CAN frame may drop when using Enhanced RX FIFO
ERR052453	FlexCAN: TX frame is not able to transmit when message buffer is programmed in a very small-time window during reception
ERR052455	AES_ACCEL: Missing CRC check on upper part of a 256 bit key
ERR052465	LFAST: LVDS receiver pad fault is latched when receiver is enabled before the common voltage is settled on the line
ERR052501	NETC: PCE does not report correct memory ID for faults from Replication and Egress Packet memory
ERR052547	OMU: Initiator hang may occur when disabling OMU via OMU_OER[OE]
ERR052558	FlexCAN: Message buffer (MB) overrun status is cleared when reading Enhanced RX FIFO (ERF)
ERR052663	NETC: missing support for 802.1Qci PSFP streamBlockedDueToOversizeFrame
ERR052805	DDRC: Can Violate Refresh Recovery for Second Rank

2 Known Errata

ERR006425: LINFlexD: FIFO buffer can not be accessed without changing the RFC counter

Description

When LINFlexD is enabled in First In, First Out (FIFO) mode by setting the Receive FIFO/Buffer mode (RFBM) bit in the Universal Asynchronous Receiver/Transmitter Control Register (UARTCR register), the Reception Data Field Length/receive FIFO Counter (RDFL_RFC) will be decremented on every read access of Buffer Data Register Most Significant (BDRM) register through the debugger.

Workaround

If the user does not want the debugger to decrement the RDFL_RFC, access to the BDRM must be avoided in the debugger.

ERR006992: PSI5: IS_DEBUG_FREEZE bit is not documented

Description

Bit 0 (the most significant bit of the register, MSB) of the Peripheral Sensor Interface 5 General Interrupt Status Register (PSI5_GISR) is not documented. This bit is the IS_DEBUG_FREEZE bit and is set when the PSI5 module is stopped in debug freeze mode. To enable the debug freeze mode, both PSI5 Debug mode Enable and the Debug Freeze Control bits must be set in the PSI5 Channel Control register (PSI5_PCCR[DEBUG_EN] = 0b1 and PSI5_PCCR[DEBUG_FREEZE_CTRL] = 0b1). When thePSI5 module receives the request to enter the debug mode, it finishes the current processing and is stopped coherently. At this stage, the IS_DEBUG_FREEZE bit is set to "1" . This bit automatically gets cleared by the hardware when the debug mode is exited.

Workaround

Expect bit 0 (MSB) of the PSI5_GISR register to be set when the PSI5 module is stopped in debug freeze mode if both DEBUG_EN and DEBUG_FREEZE_CTRL are set. The documentation will be updated.

ERR007274: LINFlexD: Consecutive headers received by LIN Slave triggers the LIN FSM to an unexpected state

Description

As per the Local Interconnect Network (LIN) specification, the processing of one frame should be aborted by the detection of a new header sequence and the LIN Finite State Machine (FSM) should move to the protected identifier (PID) state. In the PID state, the LIN FSM waits for the detection of an eight bit frame identifier value.

In LINFlexD, if the LIN Slave receives a new header instead of data response corresponding to a previous header received, it triggers a framing error during the new header's reception and returns to IDLE state.

Workaround

The following three steps should be followed -

1) Configure slave to Set the MODE bit in the LIN Time-Out Control Status Register (LINTCSR[MODE]) to '0'.

2) Configure slave to Set Idle on Timeout in the LINTCSR[IOT] register to '1'. This causes the LIN Slave to go to an IDLE state before the next header arrives, which will be accepted without any framing error.

3) Configure master to wait for Frame maximum time (T Frame_Maximum as per LIN specifications) before sending the next header.

Note:

THeader_Nominal = 34 * TBit

TResponse_Nominal = 10 * (NData + 1) * TBit

THeader_Maximum = 1.4 * THeader_Nominal

TResponse_Maximum = 1.4 * TResponse_Nominal

TFrame_Maximum = THeader_Maximum + TResponse_Maximum

where TBit is the nominal time required to transmit a bit and NData is number of bits sent.

ERR007996: PSI5: Incorrect SMC message decoding and timestamp generation in case of late last sensor message overlapping with next SYNC period pulse

Description

As stated in section 6.6 of Peripheral Sensor Interface (PSI5) Standard v2.0 and v2.1, PSI5 sensor frames should not overlap with the SYNC pulse. This overlap is considered to be an error condition. In extension to the standard requirement, the PSI5 module implemented in this device allow the possibility to manage this overlap error condition.

In case of such overlap condition:

- PSI5 message extraction happens correctly

- Timing bit error [T] and CRC error [C] flags are correctly set within the PSI5 message

- The serial messaging channel (SMC) frame counter gets reset on Sync pulse and the extraction of SMC message does not happen correctly, resulting in loss of SMC message

The PSI5 module correctly handles this error condition for the PSI5 message, but is not able to handle the SMC message correctly.

Also during this overlap condition, the timestamp appended with the overlapped slot is the new sync pulse timestamp. Whenever a sync pulse comes, the internal sync pulse timestamp capture registers are updated with the timestamp of the new sync pulse. Hence if any message overlaps with the sync pulse and that message slot is configured to capture the timestamp of the SYNC pulse (PSI5_SnFCR[TS_CAPT] = 1), then the timestamp appended for that slot is the timestamp corresponding to the new sync pulse and not of the previous sync pulse belonging to the PSI5 frame.

Workaround

The PSI5 message is properly received and analysed with the appropriate error flags set. Application software can identify from the properly received PSI5 message that an error on the bus occurred. If an SMC message is configured to be present in the slot, application software can request an immediate halt and restart of the SMC or it can wait until the SMC reception has finished and check the CRC of the SMC message

- if incorrect, a resend of the SMC message can be re-started at that point in time. Further, application software can check the timestamp of the previous message in the final slot from the previous SYNC period to identify if the wrong timestamp has been captured

- if the difference between the two messages is equivalent to two SYNC periods, application software can correctly identify that an overlap of the final message with the following SYNC pulse has occurred assuming the slot is configured to capture the SYNC pulse timestamp (PSI5_SnFCR[TS_CAPT] = 1) rather than the message timestamp (PSI5_SnFCR[TS_CAPT] = 0)

ERR008075: PSI5-S: Unrecoverable messages are not flagged in the mailbox status register channel 0

Description

In the Peripheral Sensor Interface Support module (PSI5-S), unrecoverable messages are correctly routed to Frame 1 of the special channel 0 but the Frame 1 error bit (F1_ERR) of the Mailbox status register channel 0 (PSI5S_MBOX_SR_CH0) is not set.

Instead, the Frame n error bit (Fn_ERR, n = 0 to 5) of the mailbox status register of the channel y (PSI5S_MBOX_SR_CHy, y = 1 to 7) of the intended channel.

Workaround

Ignore the value of PS_SR_MBOX_CH0[F1_ERR].

Assume that any message that arrives in the Channel 0 Frame 1 always contains some error that can be identified reading the Message Recovery Unit output Buffer 2 register 1 (PSI5S_MRU_BUF2_REG1). Also, when reading the message corresponding to the channel 0 frame 1, the PSI5S_MBOX_SR_CHy[Fn_ERR] of the faulty channel must be cleared.

ERR008141: PSI5-S: Global mode transition interrupt does not work

Description

In the Peripheral Sensor Interface Support module (PSI5-S), the Global Mode Transition Done bit (GL_MODETR_DONE) of the Global Status Register (PSI5S_GLSR) does not get set unless there is a read access to the PSI5-S module.

As a consequence, even if enabled, the associated mode transition interrupt will not be triggered.

Workaround

Do not enable the Global Mode transition enable interrupt. Poll the PSI5S_GLSR[GL_MODETR_DONE] to detect the module entered the desired mode.

ERR009978: eMIOS: Unexpected channel flag assertion during GPIO to MCB mode transition

Description

When changing an Enhanced Modular IO Subsystem (eMIOS) channel mode from General Purpose Input/Output (GPIO) to Modulus Counter Buffered (MCB) mode, the channel flag in the eMIOS Channel Status register (eMIOS_Sn[FLAG]) may incorrectly be asserted. This will cause an unexpected interrupt or DMA request if enabled for that channel.

Workaround

In order to change the channel mode from GPIO to MCB without causing an unexpected interrupt or DMA request, perform the following steps:

(1) Clear the FLAG enable bit in the eMIOS Control register (eMIOS_Cn[FEN] = 0).

(2) Change the channel mode (eMIOS_Cn[MODE]) to the desired MCB mode.

(3) Clear the channel FLAG bit by writing '1' to the eMIOS Channel Status register FLAG field (eMIOS_Sn[FLAG] = 1).

(4) Set the FLAG enable bit (eMIOS_Cn[FEN] = 1) to re-enable the channel interrupt or DMA request reaction.

ERR011508: NETC: Transmission Overrun counter defined in IEEE 802.1Qbv is not implemented

Description

NETC doesn’t support the per-traffic class Transmission Overrun counter specified in the IEEE Std 802.1Q™-2022. As per the standard, this counter is to be incremented when the implementation detects that the transmission gate associated with a queue has closed and a frame that originated from the queue is still being transmitted by the MAC.

Workaround

Overrunning a next gate-close event can be prevented by ensuring that the schedule (gate control list) is created in such way that the time needed for all transmissions can be determined or that the design of the schedule is done in such a way that the intended pattern of transmission will not overrun the next gate-close event. Assuming that the schedule is designed as described above, overrunning a next gate-close event is not expected, as the NETC implementation employs a method that provides a very accurate timing with extreme low delay variance by enforcing a fixed delay between the time gate scheduling timing point and the timing point at which the frame is transferred to the MAC. This delay is configurable via the EaTGSLR/ S0TGSLR[MIN_LOOKAHEAD] register. Furthermore, the PTGSATOR[ADV_TIME_OFFSET] register can be used to adjust for the latency encountered across the MAC plus if needed, delays outside of NETC (e.g. PHY delay).

ERR050142: ERM_AE: Correctable Error Count Register value is incorrect

Description

Error count value read from the Application Extension die Error Reporting Module (ERM_AE) Correctable Error Count Register is incorrect and is reported as twice the actual error count. Each single bit error event reported by AE Platform RAM Controller (PRAM_AE) is counted twice by the ERM_AE Correctable Error Count Register.

Workaround

When reading the contents of the ERM_AE Correctable Error Count Register, divide the returned value by two in order to get the actual error count.

ERR050167: eDMA_AE: Uncorrectable memory error in EDMA_AE TCD memory is ignored by EDMA_AE

Description

Uncorrectable memory errors (double bit error) in the Applications Extensions's (AE) enhanced Direct Memory Access (eDMA) local Transfer Control Descriptor (TCD) memory are ignored by the eDMA_AE. A double-bit error is not corrected and the eDMA transfer is not prevented; the eDMA_AE will use the erroneous data returned from the eDMA_AE TCD memory. Unknown eDMA_AE behavior may result from using the erroneous data. The uncorrectable error is properly reported and captured in the ERM_AE and Fault Collection and Control Unit (FCCU).

Single bit errors are properly corrected by the eDMA_AE.

Workaround

ERM_AE and/or FCCU must be monitored within the Fault Tolerant Time Interval (FTTI) for uncorrectable EDMA_AE TCD memory errors. If any error is detected, corrective action must be taken by the application.

ERR050366: AE_AHB: Back to back writes of less than 128-bits will corrupt memory

Description

Back to back accesses to the Application Extension (AE) that are less than 128-bits and are not 128-bit aligned may produce unexpected results if the memory region is not defined to be in device mode. The AE system covers the address range of 0x5500_0000 - 55FF_FFFF. 8-, 16-, 32- and 64-bit accesses may write to an incorrect address and the remaining bytes of the 128-bit word may be cleared (set to 0).

Workaround

Always configure the address range of 0x5500_0000 - 55FF_FFFF in the Memory Protection Unit (MPU) as "device" mode (non-cachable, (nGnRE mode)).

ERR050457: NoC: Trace probe packet counting with length filter may hold incorrect count value.

Description

When using trace counter with filter in Flexnoc packet probe, the packet count may not be accurately captured in the event of interleaved read response from the slave. In this case, only the first fragment is counted.

Workaround

To count the responses time accurately on basis of the Length filter, count bytes should be used. This is done by setting the counter SRC as FILT_BYTE_EN or LUT_BYTE_EN. In this case, the counter would only increment as per the valid bytes in each response fragment instead of packet count.

ERR050501: Core: DFSR.EXTERNAL is not set correctly when waking up from sleep

Description

Cortex-M33 1367266-C:

An external debug event which causes the processor to enter Debug state or the debug monitor should set DFSR.EXTERNAL. It has been found that this field is not set if the event occurs while the processor is asleep.

Workaround

There is no workaround.

ERR050502: Core: Execution priority might be wrong for one cycle after AIRCR is changed

Description

Cortex-M33 1435973-C:

AIRCR is used in the NVIC active tree to calculate the execution priority, which in turn is used to determine fault escalation, exception preemption, and other NVIC-related behaviors. When the active tree is pipelined and there are high latency IRQs active, there might be a glitch in the active tree output for one cycle after AIRCR is changed. The glitch results in NVIC producing wrong execution priority that is neither based on the old AIRCR value nor the new one.

Workaround

There is no workaround for this erratum.

ERR050504: Core: Sorting of pending interrupts might be wrong when high latency IRQs are pending

Description

Cortex-M33 1540599-C:

The NVIC contains a pending tree which sorts all pending and enabled interrupts based on priorities. If DHCSR.C_DEBUGEN and DHCSR.C_MASKINTS are 1, DHCSR.S_SDE is 0 and halting debug is allowed, then Nonsecure PendSV, Non-secure SysTick, and Non-secure IRQs should be masked off and they should not affect the sorting of pending and enabled secure interrupts. If multiple high latency IRQs are pending and enabled with different security targets and priorities, then Non-secure IRQs which should be masked off might cause the pending tree output to be a pending Secure nterrupt without highest priority. This is because of incorrect masking before doing priority comparisons in the tree.

Workaround

There is no workaround for this erratum.

ERR050505: Core: Access permission faults are prioritized over unaligned Device memory faults

Description

Cortex-M33 1080541-C :

A load or store which causes an unaligned access to Device memory will result in an UNALIGNED UsageFault exception. However, if the region is not accessible because of the MPU access permissions (as specified in MPU_RBAR.AP), then the resulting MemManage fault will be prioritized over the UsageFault.

Workaround

There is no workaround.

However, it is expected that no existing software is relying on this behavior since it was permitted in Armv7-M.

ERR050509: Cortex-R52 1328481-C The debugger view of the number of cores might be incorrect when switching between Split and Lock modes

Description

The Cortex-R52 processor can be configured with Split/Lock, which allows the number of cores that the processor

presents to vary, under Cold reset, between Split mode and Lock mode. The debugger uses the entries in the Cortex-

R52 debug ROM table to determine the number and offsets of the cores in the processor. The processor ROM table

always indicates the current number of cores because of this erratum. If a debugger reads the ROM table and the mode

is subsequently switched, the information that it reads is incorrect.

Workaround

To ensure a debugger has a complete description of the system, the ROM table must be read following each reset of all cores in the Cortex-R52 processor. A reset of a core can be detected by reading EDPRSR.

ERR050679: NETC: Accesses to 64-bit stats registers must be performed atomically

Description

The MAC contains 64-bit stats registers, which are accessed 32 bits at a time as follows:

- Application reads the lower 32 bits: Captures the entire 64-bit stat value into a shadow register and returns the lower 32 bits.

- Application reads the upper 32 bits: Returns the upper 32 bits of the 64-bit stat value from the shadow register.

The MAC only contains one shadow register, so if two application threads are accessing the stats from the same MAC, it is possible for one thread's read of the upper 32-bits to return the upper 32-bit value from the other thread's stat access.

Note: 64-bit register accesses are not supported on this device.

Workaround

If the application software supports multiple threads accessing the same MAC, it must implement a semaphore protecting accesses to the stats registers to ensure that only one thread is accessing stats at a time.

ERR050786: Core: ATB flush response may be delayed

Description

The Embedded Trace Macrocell (ETM) supports an external flush request for each ATB bus. Under certain timing conditions, an AFREADY response from the Cortex-R52 processor may be delayed until a new ATB transfer is generated by the Cortex-R52 processor.

Conditions:

The erratum occurs if the following sequence of conditions are met:

1. The ETM is enabled

2. Trace data is generated on the ATB bus

3. An external flush request is generated

4. Trace data stops being generated due to filtering in the ETM or the ETM being disabled

Implications:

External trace infrastructure which is waiting for trace to be captured may stall forever (for example in a 'flush and stop' scenario). All of the trace data will be captured, but it is not possible to identify this by polling the trace capture device. If the flush is acknowledged, this can be treated as a reliable indication.

If the ETM is enabled again after the erratum has been triggered, the flush logic should become active again. If a flush is generated while trace is being generated, the only effect will be a delay in acknowledging the flush. This should not have

any observable impact.

In a system with multiple trace sources, the delayed flush response may prevent other trace sources from accessing the ATB bus if they are generating trace while the flush is in progress. This could cause trace to be lost from these other

sources.

Workaround

There is no workaround to reliably avoid this erratum. As an alternative to waiting for the flush to be acknowledged, a sufficiently long timeout can be used if it is likely that trace generation has stopped.

ERR050787: Core: Data ATB flush may not respond

Description

The Embedded Trace Macrocell (ETM) supports an external flush request for each ATB bus. If the ETM is active and configured with no data trace, there will be no flush response on the data ATB.

Conditions:

This erratum occurs when all of the following conditions are met:

- The ETM is enabled

- Data trace is disabled (both TRCCONFIGR.DA and TRCCONFIGR.DV are clear)

- The ETM is not in a low-power state due to WFI or WFE

- An ATB flush is requested on the data ATB bus

Implications:

External trace infrastructure which is waiting for trace to be captured may stall forever (for example in a "flush and stop"

scenario). Instruction trace will flush as expected, but this may not be observed at the trace capture device. Disabling the

ETM or entering a low-power state will restore the expected behavior.

Workaround

To avoid this erratum:

- Enable TRCCONFIGR.DA or TRCCONFIGR.DV
- Program all of the following registers to zero if data trace is not required:
- TRCVDARCCTLR
- TRCVDCTLR
- TRCVDSACCTLR
- TRCEVENTCTL1R.DATAEN

This will slightly increase power consumption, but it will not cause any data trace generation.

ERR050956: CEVA_SPF2: OCEM ROM Table is not discovered by Coresight Table Walk

Description

The ROM table component of SPF2 OCEM cannot be found by Coresight discovery.

Workaround

Debug tools must add this component manually to the list of debug components in the system, as it cannot be discovered automatically.

ERR051023: NETC: Excessive collisions are counted as late collisions

Description

The MAC contains separate statistics registers for late collisions and excessive collisions. While the excessive collision counter (PMm_TECOLn) counts correctly, the late collision counter (PMm_TLCOLn) counts both excessive and late collisions.

Workaround

To obtain the actual number of late collisions, read both PMm_TECOLn and PMm_TLCOLn, and subtract PMm_TECOLn from PMm_TLCOLn.

ERR051025: NETC: Preemption verify time longer than specified

Description

The IEEE 802.3 specification, clause 99.4.7.6 specifies verify_timer to be in multiples of 1 ms +/- 20%. The verify_timer sets the verify timeout value - after the timer expires, the MAC merge will retransmit a verify mPacket or, after too many retransmissions, fail.

The counter on this device uses an incorrect rate (400 MHz), so the verify timer is longer than specified by (400-clk)/clk %, where clk=NETC system clock frequency.

Workaround

To program a verify timer of n ms, set MAC_MERGE_MMCSR[VT]=(clk/400)*n, where clk is NETC system clock frequency.

ERR051034: NETC: HTA SIUNSBESR counter is not incremented

Description

System bus errors received in response to read and write transactions issued by an enetc’s SI0 function are not counted in the per-SI error counters.

Workaround

User cannot use RCEC for System Bus errors on SI0. These system bus errors are reported to software in the Rx/Tx BDR descriptors’ status code field and are also visible in the TBSR[SBE] or RBSR[SBE] per-ring status field.

ERR051048: NETC: Management command with search action responds with incorrect NUM_MATCHED

Description

The NUM_MATCHED field in the command NTMP response header may be incorrect when the search access method is used for the following tables: FDB table 15, L2 IPv4 Multicast Filter table 16, VLAN Filter table 18, Ingress Stream Identification table 30, or Ingress Stream Filter table 32.

Workaround

For the query command, limit the message response buffer size (RESPONSE_LENGTH field in the request header) so that NETC returns no more than one entry at a time. Under this condition, the NUM_MATCHED field value is correct.

ERR051051: Core: A partially completed VLLDM might leave Secure floating-point data unprotected

Description

Arm errata 2219175

Affects: Cortex-M33

Fault Type: Programmer Category B

Fault Status: Present in r0p0, r0p1, r0p2, r0p3, r0p4, r1p0. Open.

The VLLDM instruction allows Secure software to restore a floating-point context from memory. Due to this erratum, if this instruction is interrupted or it faults before it completes, then Secure data might be left unprotected in the floating point register file, including the FPSCR.

Configurations affected:

This erratum affects all configurations of the Cortex-M33 processor configured with the Armv8-M Security Extension and the Floating-point Extension.

Conditions:

This erratum occurs when all the following conditions are met:

• There is no active floating-point context, (CONTROL.FPCA==0)

• Secure lazy floating-point state preservation is not active, (FPCCR_S.LSPACT==0)

• The floating-point registers are treated as Secure (FPCCR_S.TS==1)

• Secure floating-point state needs to be restored, (CONTROL_S.SFPA == 1)

• Non-secure state is permitted to access to the floating-point registers, (NSACR.CP10 == 1)

• A VLLDM instruction has loaded at least one register from memory and does not complete due to an interrupt or fault

Implications:

If the floating-point registers contain Secure data, a VLSTM instruction is usually executed before calling a Non-secure function to protect the Secure data. This might cause the data to be transferred to memory (either directly by the VLSTM or indirectly by the triggering of a subsequent lazy state preservation operation). If the data has been transferred to memory, it is restored using VLLDM on return to Secure state. If the VLLDM is interrupted or it faults before it completes and enters a Non-secure handler, the partial register state which has been loaded will be accessible to Non-secure state.

Workaround

To avoid this erratum, software can ensure a floating-point context is active before executing the VLLDM instruction by performing the following sequence:

• Read CONTROL_S.SFPA

• If CONTROL_S.SFPA==1 then execute an instruction which has no functional effect apart from causing context creation (such as VMOV S0, S0)

ERR051068: CEVA_SPF2: The DDMA incorrectly ends the ongoing IIT task

Description

If the following scenario occurs:

1) The Data DMA (DDMA) is configured with a 2D download task with Destination Clipping enabled and,

2) The clipping parameters are such that, in one of the cycles during this download an entire DDMA FIFO line is clipped and,

3) A subsequent Internal-to-Internal Transfer (IIT) task becomes temporarily starved of data due to TCM read contention.

Then:

The DDMA will incorrectly end the ongoing IIT task before all of the data has been transferred, and the subsequent behavior of the DDMA will be unpredictable.

Workaround

Do not configure the DDMA to execute 2D Destination Clipping (MSS_2DCFG1[CLIP_EN] == 0x1) tasks and IIT tasks together.

However, if this is required, then do one of the following to execute the IIT tasks:

● Replace each IIT task with an upload task followed by a download task.

● Replace each IIT task with core load/store instructions (ld/st/vld/vst).

● Program IIT tasks directly from the core, and then ensure that no core or QMAN DTCM accesses contend with the IIT

operations.

Contention can be avoided by doing one of the following

○ Ensure that the IIT accesses a different DTCM block than the core and QMAN.

○ Pause the QMAN (QMAN_PAUSE), and then ensure that all core load/store accesses are complete by polling the DMSS_IDLE field in the MSS_DMBE register.

ERR051080: CEVA_SPF2: Access to the CPM region might not complete according to the AXI protocol

Description

If an EDAP/AXIs port read access to the CPM region (bits 22:21 of the read address are '10') is followed directly by a read access on the same EDAP/AXIs port to the unmapped address space (bits 22:21 of the read address are '11'), then the first access to CPM region might not complete according to the AXI protocol, and the subsequent behavior of the EDAP/AXIs ports will be unpredictable.

Workaround

Do not access the Unmapped address space via EDAP/AXIs.

ERR051082: QMAN may set incorrect field in the QMAN_IDM_CROSS register

Description

If the following scenario occurs:

1) Two or more QMANs are arbitrating to send a task to the DDMA and,

2) The QMAN arbitration parameters (PRI_ABS, PRI_FRAME_NO, PRI_FRAME_ORDER) are identical for the arbitrating queues and,

3) The winning task starts beyond the upper DTCM boundary.

Then:

● The QMAN will detect the IDM crossing error, but might set the incorrect field in the QMAN_IDM_CROSS register (the bit set will be for one of the QMANs that were requesting with equal arbitration).

Workaround

Ensure that no two QMANs have exactly the same arbitration parameters (PRI_ABS, PRI_FRAME_NO, PRI_FRAME_ORDER). Note: The recommended action for a QMAN_IDM_CROSS critical error is a full reset of the core.

ERR051129: NETC: Errored frames due to parity error are not counted in TFCS and TERR

Description

The transmit stats TFCS and TERR should increment for any frame transmitted with an invalid FCS except underflows. If the frame has a forced FCS error due to a detected parity error in the MAC, but no other error condition, TFCS and TERR should increment but do not, and TFRM (count good frames) should not increment, but does.

Workaround

Errored frames due to detected parity error in the MAC are counted in UNIECTR (uncorrectable non-fatal integrity errors). If it is non-zero and UNIESR indicates errors detected in the MAC, then TFCS, TERR, and TFRM may bf off by up to the value of UNIECTR.

ERR051130: NETC: Egress time gate scheduling can get corrupted when functional level reset is applied or when time gating is disabled.

Description

IEEE 802.1Qbv (also known as time gate scheduling) is one of the Time-Sensitive Networking (TSN) standards, which introduces a time-based transmission mechanism operating on a global (offline) configured periodic schedule called the gate control list. Each traffic class (or queue) on an egress port has a gate which controls if the queue is open or closed. Only frames in queues with open gate are eligible for transmission. The gate control list controls the gate state and is executed cyclically based on a global time.

When a gate control list is configured, it is first considered as the administrative gate control list. When its configuration change time is reached, the administrative gate control list is installed and becomes the operational gate control list. In other words, it becomes active. A gate control list is configured by updating the Time Gate Scheduling table. There is one table entry for each port, where a table entry includes both an administrative gate control list and an operational gate control list along with related configuration and status parameters. New gate control lists can be configured at any time

The issue occurs when a functional level reset (FLR) is issued or when time gate scheduling is being disabled (PTGSCR[TGE] is set to 0), and the first configured administrative gate control list is in the process of being installed as the operational gate control list. When an FLR is issued or time gate scheduling is being disabled during this small window of time, one of the internal contexts is not cleared properly resulting in stale context lingering even after FLR or after time gate scheduling has been enabled. This in turn will cause the operational gate control list to be corrupted indefinitely. This issue will not occur for subsequent gate control configuration updates after the first gate control list has become operational.

Workaround

To avoid this issue, a gate control list which sets the gate state of every queue to Open during its entire cycle must be configured immediately after time gate scheduling for the port that has been enabled (TGE=1). As such it would default time gating scheduling conditions.

Suggested settings for the first administration gate control list are:

- Number of gate control list entries = 2

- All gates open in gate list entries

- Administrative gate operation type specified in gate list entries = 0

- Admin cycle time specified as a small value (100usec) , gate time interval in gate list entries = 50usec

- Admin base time set to current time.

Since the first administration gate control list takes effect immediately, it eliminates the possibility of encountering FLR or TGE being disabled at this early stage. Subsequently, admin lists can be installed as desired.

ERR051140: RTU.SRAMCTL: Low power feature cannot be used

Description

The low power control should not be enabled for any of the RTU SRAMs.

This applies to both RTU0 and RTU1 and all instances of SRAMCTL_Cn and SRAMCTL_Dn in each.

For all SRAMCTL_C/Dn.RAMPC registers, do not set any bits in field PCUT to 1.

Workaround

The default state for low power control of all RTU SRAMs is disabled. Do not enable.

ERR051146: FCCU: Back to back faults with functional reset reaction not working properly

Description

When a fault programmed to generate a functional reset reaction occurs and then a second fault, also programmed to generate functional reset, occurs while the first fault's reset exit sequence is still in progress, the second fault's status bit will be set but there will be no second reset generated.

Workaround

Option1 : For faults where it is possible for this situation to occur, do not use the functional reset reaction. Instead use a destructive reset or interrupt reaction.

Option2 : On functional reset recovery (detected by reading MC_RGM.FES and MC_RGM.DES registers), if multiple fault status bit are set – issue another functional reset via software.

ERR051153: FCCU: RTU Faults may get inadvertently set during self-test

Description

During self-test, it is possible that some faults may get asserted, though an actual fault hasn't occurred. Software should disable/clear the FCCU faults to avoid unnecessary fault reactions after self-test.

Workaround

Perform the following steps:

1. Before running LBIST on RTU0/1 reset partition, disable faults in RTU LFCCU.

- Write 0x0000_0000 to Fault Enable register, FHFLTENC00.

- Write 0x0000_0000 to Fault Enable register, FHFLTENC01.

2. After self test completes on RTU0 and/or RTU1 Rest partition, software needs to clear the following registers to discard false faults in RTU Local FCCU.

a. Clear false status indications (bits are 'write 1 to clear')

- Write 0xFFFF_FFFF to Fault Status register, FHFLTS00

- Write 0xFFFF_FFFF to Fault Status register, FHFLTS01

b. Clear OVF_DET bit

- Write '1' to bit 8 (OVF_DET) of Global DID FSM Status register, GINTOVFS.

ERR051162: I3C: Slave reset not supported when I3C is in slave mode

Description

When operating in slave mode, the I3C module is unable to reliably detect a slave reset from an external master. As a result this feature is not supported on this device and should be disabled in application software.

Workaround

When I3C is operating in slave mode, the application must mask the slave reset interrupt by writing '1' to the SLVRST field in the I3C Slave Interrupt Clear (SINTCLR) register.

ERR051163: MC_RGM: Incorrect destructive reset event status

Description

If a destructive reset event occurs during functional reset entry sequence, certain destructive reset status bits may incorrectly be set in the MC_RGM.DES register. The affected indicators in MC_RGM.DES are: F_DR_8, F_DR_13, and F_DR_18.

Workaround

The affected indicators in MC_RGM.DES should be ignored when any MC_RGM.FES[F_FR_<n>] field is set at the same time that any of the affected MC_RGM.DES indicators is also set. Clear the indicators by writing 1 to each affected FES and DES field.

ERR051164: DDRC: Read Data Buffer SRAM Fault Reports Incorrect Device ID

Description

When an RDB SRAM Multiple-Bit Error (MBE) is detected by the DDRC, an incorrect Domain ID (DID) may be reported to the FCCU for the fault.

In addition, the error response back to the master may not be reported, or it may be reported back incorrectly on a subsequent read.

Workaround

Since the DID reported by the RDB SRAM MBE fault cannot be relied upon, the reaction for this fault must be configured the same way for all DIDs in the system. In the Central VFCCU (C_VFCCU), DIDs are mapped to Fault Handler IDs in the Global DID-FHID Map (GDFHID_C0/C1) registers. Thus the reactions to this fault in all FHID maps must be the same. If the reaction programmed is an interrupt handler, that interrupt handler must ignore the DID value presented by the C_VFCCU and consider that the fault could have come from any DID source.

Since the error response to the access that detects the MBE may not be reported, the fault reaction must be configured and enabled for all DIDs in order to ensure it is caught and handled.

ERR051183: MDM-AP: DTS trigger output cannot be generated inside the MDM-AP

Description

Development Trigger Semaphore (DTS) output, dts_trigger_out, cannot be generated inside the MDM-AP. Software will have to write to the enable register to control the signal manually.

Workaround

To assert dts_trigger_out:

1. Debugger sets up the enable register.

2. CPU writes to the SEMAPHORE register

3. EXTRA STEP: CPU re-writes the enable register (via external address space) to set the trigger. (Will probably have to also read reg first to preserve existing values).

4. DTS TRIGGER OUT is now asserted

To clear

1. Debugger Read SEMAPHORE register

2. EXTRA STEP: Debugger Re-write to enable register to clear the trigger (debugger should already know the values used, it originally setup the register.)

ERR051188: NETC: Register SITSR not accessible for Virtual Station Interfaces

Description

Register SITSR is a read-only status register which includes the timer function synchronization information. In addition there is a set of interrupt registers that trigger when there is a change in the SYNC bit.

All of these registers are intended to be present in each SI. While the interrupt registers are present, the SITSR register is not available in the VSIs, only in the PSI. The result is that a VSI may know when the SYNC bit changes but will not know the current status.

Workaround

A VSI may send a message to the PSI for general notification or to gain access to hardware resources using the VSIMSG/PSIMSG set of registers. This method can be used to gain access to the PSI SITSR register value when there is a change to the SYNC bit as determined by the VSI interrupt.

ERR051202: NETC: Possible transmit MAC underrun at low 10M/100M speeds when the NETC switch is operating in cut-through forwarding mode

Description

Under the scenario below, a MAC transmit underrun may occur, resulting in either the transmission of invalid frames, or the frame transmission being aborted. A MAC transmit underrun occurs when the MAC transmit FIFO becomes empty during the transmission of a frame.

The scenario where the underrun may occur is when:

− Egress link speed 100M or 10M

− Cut-through operation enabled

Workaround

Software must always configure the TX MAC to wait until 40 bytes of data are built up in the transmit FIFO before beginning transmission on the link (default is 24 bytes). This is achieved by writing the value of 0x0001_0105h to the internal register at offset 0x1020h within the Ethernet_MAC_port memory map for each egress port subject to the conditions of this errata.

Note that this workaround may impact frame transmission latency relative to the default setting. This increase in latency can be up to 32 bytes worth of line time (up to 2,560ns increase on a 100M link) and the precise latency impact will depend on any frame modifications being performed.

ERR051218: DDRC: Some timings not properly derated at high temperature

Description

When using the dynamic refresh rate feature (via DDR_SDRAM_CFG_3[DYN_REF_RATE_EN], the DDR Memory Controller (DDRC) will automatically adjust the refresh rate based on the temperature of the DRAM. This works correctly. However, if the MR4 refresh rate provided back to the controller uses a value of 3'b110, the DDRC should use the value in TIMING_CFG_2[DERATE_VAL] to derate certain timing configs:

* {TIMING_CFG_3[EXT_PRETOACT], TIMING_CFG_1[PRETOACT]}

* {TIMING_CFG_3[EXT_ACTTOPRE], TIMING_CFG_1[ACTTOPRE]}

* {TIMING_CFG_3[EXT_ACTTORW], TIMING_CFG_1[ACTTORW]}

* {TIMING_CFG_3[EXT_ACTTOACT], TIMING_CFG_1[ACTTOACT]}

* TIMING_CFG_8[PRE_ALL_REC]

However, the derating for these specs will not be applied correctly within the DDRC. Note that the dynamic refresh interval updates are not affected by this erratum.

Workaround

If operating in an application with a DRAM that may require AC spec derating (i.e., the DRAM requires derating the effected timing parameters at high temp, and the DRAM will be operating in that range, which is **typically** above 85C), then apply the derating to the following configs when programming the DDRC:

* {TIMING_CFG_3[EXT_PRETOACT], TIMING_CFG_1[PRETOACT]}

* {TIMING_CFG_3[EXT_ACTTOPRE], TIMING_CFG_1[ACTTOPRE]}

* {TIMING_CFG_3[EXT_ACTTORW], TIMING_CFG_1[ACTTORW]}

* {TIMING_CFG_3[EXT_ACTTOACT], TIMING_CFG_1[ACTTOACT]}

* TIMING_CFG_8[PRE_ALL_REC]

Note that the de-rating for these is typically 1.875ns, but the DRAM vendor datasheet should be referenced.

ERR051241: eDMA4: Channel preemption feature can operate incorrectly

Description

Channel preemption can cause incorrect behavior under the following conditions:

• The preempted channel has CHx_PRI[ECP]=1

• The preempting channel has a higher priority than the preempted channel based on the setting of CHx_PRI[APL]

• The preempting channel is activated while the preempted channel has an active transfer on the AXI bus

Workaround

Do not use the preemption feature. For all channels, the TCD register CHn_PRI[31:30] must have at least one of the two settings:

1. Keep CHn_PRI[ECP] = 1'b0 - The channel cannot be suspended by a higher priority channel's service request (default).

2. Set CHn_PRI[DPA] = 1'b1 - The channel can be temporarily suspended by the service request of a higher priority channel.

ERR051243: NETC: Multi-entry SEARCH table management command returns invalid error code.

Description

The response of a successful multi-entry SEARCH table management command could return an error (ERROR=0x8A) when the command did not encounter a real error - i.e. no programming errors or faults.

The error is an indication that the response buffer has been filled, but the SEARCH command did not reach the end of the table.

This should be returned as part of the existing STATUS field, not as an ERROR.

Workaround

On the response of a SEARCH command, software will deem the command successful if the ERROR field is 0x0 OR 0x8A, and then will use the NUM_MATCHED and STATUS field to interpret the response. That is, ERROR=0x8A should not be treated as an error, or as an indication that the command failed.

ERR051245: NETC: PSIaCFGR1 and SIRBGCR registers are not reset on Function Level Reset (FLR)

Description

The PSIaCFGR1 and SIRBGCR registers are in the ENETC and SI memory maps.

They should be reset when the ENETC or SI is reset by a Function Level Reset (FLR), but they are not. They are only reset by a hard reset of the NETC IP block.

A stale value from pre-FLR will persist and be present in the register following FLR.

Workaround

If there is a possibility that the PSIaCFGR1 or SIRBGCR register was written with a non-reset value prior to an FLR, then software must ensure that the register is always written following an FLR.

ERR051246: NETC: Tx/Rx disable (POR[RXDIS] and POR[TXDIS]) are enabled out of reset

Description

The POR register's RXDIS and TXDIS bits reset to 'enabled'. This reset value creates the following hazard scenario for ENETCs connected to internal switch ports:

1) An internal ENETC connected to the switch is reset by FLR (Function Level Reset).

2) The switch is not reset and continues operating.

3) Rx traffic continues to flow into the ENETC throughout its FLR process.

4) The arrival of an Rx frame at the ENETC can collide with its IERB-load event. This simultaneous event results in corruption of the ENETC's Rx byte credit counter. A corrupted Rx byte credit value impairs operation following the FLR. There may be other internal per-ENETC resources that are impacted by traffic arrival during FLR and initialization (memory allocation settings, etc.).

The reset values for RXDIS and TXDIS of the ENETC function should be 1 (disabled) instead of 0 (enabled) so that Rx traffic cannot arrive at the ENETC once its FLR process has begun.

Workaround

Prior to FLR of an internal ENETC, software should disable the internal switch port's POR[TXDIS] register bit to prevent frames from flowing into the internal ENETC during its FLR and bring up.

After ENETC FLR and bring up, the POR[TXDIS] for the internal switch port that the ENETC is connected to may be cleared to allow traffic to flow.

ERR051247: NETC: NETC has no mechanism for system bus errors from VSI initiated transactions to the PSI to notify the PSI of the bus error

Description

If a system bus read or write error occurs during the DMA transfer from a Virtual Station Interface (VSI) to a Physical Station Interface (PSI), the PSI is informed of the message arrival but is not informed of the error event. SIUSBEDR will only reflect the failure for the VSI.

The system bus error is reported through mechanisms outside of the VSI-to-PSI mechanism.

e.g. The DDR controller detecting an ECC error that reports the bus error to ENETC.

However, the detection will be delayed until the PSI SW handling routine has acted on the received message.

If a particular application of NETC cannot tolerate the possibility of PSI software acting on a message that was received with an error during transfer, then an integrity code with zero-fill of the buffer must be used to validate the message received by the PSI.

Workaround

An integrity code with zero-fill of the buffer can be used to prevent the possibility of PSI software acting on a message that is received with error.

ERR051251: DDRC: Violation of tRFCpb

Description

When the DDR Memory Controller (DDRC) is configured for per-bank refresh mode, there are times when it will violate tRFCpb timing by issuing a per-bank refresh to the same bank or an all-bank refresh after a prior per-bank refresh. The DDRC will guarantee that TIMING_CFG_9[REFTOREF_PB] (programmed to tpbR2pbR) is met between any 2 per-bank refreshes. However, there is an erratum in that it should also guarantee TIMING_CFG_9[REFREC_PB] (tRFCpb) when issuing a per-bank refresh to the same bank or when issuing an all-bank refresh after a per-bank refresh.

Workaround

If using per-bank refresh (TIMING_CFG_9[REFREC_PB] != 0x0), then program TIMING_CFG_9[REFTOREF_PB] and TIMING_CFG_9[REFREC_PB] to each meet the maximum of (tRFCpb, tpbR2pbR). tRFCpb is expected to be the maximum of those 2 timing specs.

Another workaround is to disable per-bank refresh by programming TIMING_CFG_9 to 0x0.

ERR051254: NETC: Administrative gate control list can get configured inadvertently when an exception is detected and notified

Description

The IEEE 802.1Qbv implementation (also referred in NXP documentation as time gate scheduling) does not support configuration of an administrative gate control list with a base time that is more than 1 second in the past. The base time specifies the time at which the administrative gate control list is to become operational.

The time gate scheduling configuration and operational information for each port, including the administrative gate control list, is kept in the Time Gate Scheduling table. Control messages (command requests and responses) exchanged between the software and hardware, are used to perform operations on tables, including the Time Gate Scheduling table. When a command request is issued to configure an administrative gate control list with a base time that is more than 1 second in the past, in a Time Gate Scheduling table entry, the hardware returns a response with the error field set to 0x0D6 indicating 'Update action attempted with ADMIN_BASE_TIME specified is more than one second in the past” .

The expectation when receiving this kind of an error code is that the command has not been executed and that the hardware remains in a coherent state. However in this particular case, the hardware inadvertently stores the administrative gate control list in the table, and this gate control list will never become operational.

Workaround

If a command response with an error code set to 0x0D6 is returned in response to an update command request against an Time Gate Scheduling table entry, software must issue another update command request against the same Time Gate Scheduling table entry, to remove the administrative gate control list from the Time Gate Scheduling table entry.

ERR051260: NETC: During initial Initialization of NETC, all ENETCs and Switch PCI functions must be enabled for NETC to accept table management commands for any function

Description

The IERB contains boot-time configuration registers which must be locked before NETC can operate. The IERB should either be locked explicitly by software or be pin strapped to lock automatically at boot-up time. If the IERB is not locked before NETC operation, the first PCIe function that sets (0b1) PCI_CFH_CMD[MEM_ACCESS] will lock the IERB registers. The purpose of locking the IERB or setting PCI_CFH_CMD[MEM_ACCESS] is to trigger a signal for when NETC or a function should sample register values from the IERB. When a function sets PCI_CFH_CMD[MEM_ACCESS] it should also trigger a signal for other functions, regardless if those functions will be used or not.

Upon NETC initialization (i.e. after power-on or soft reset, etc.), the ingress processing engine initializes parts of internal memory associated to the Switch and each ENETC function. This initialization requires the IERB trigger of multiple functions before starting. It will wait for the Switch and all ENETCs to be triggered before it accepts any table management commands for any function. Here, any function that is not used (PCI_CFH_CMD[MEM_ACCESS]=0b) fails to be implicitly triggered by the activation of another function and thus the the initialization will not start.

Workaround

Enable PCI_CFH_CMD[MEM_ACCESS] for all functions regardless if they are to be active or not. This will guarantee that initialization will start for processing engine. After NETC initialization is complete, software may disable functions that are not required.

ERR051261: NETC: Egress ports can violate 802.3 half-duplex jam protocol after collision event

Description

When operating in half-duplex, the NETC egress port may sometimes violate the 802.3 collision protocol. In the case where a collision is detected early during preamble transmission, the 4 octet jam pattern will be transmitted immediately. This violates the 802.3 spec, which states all preamble and SFD bits must be transmitted prior to the jam pattern.

Workaround

None

If the protocol violation occurs, a packet will be transmitted from NETC without a valid SFD octet. The packet will consist of one or more octets of preamble, followed by a 4 octet jam pattern. When this occurs, the link partner will see an invalid packet due to missing SFD, as opposed to an invalid frame due to bad CRC. This protocol violation will not cause any functional impact to NETC itself or to the link partner.

ERR051262: NETC: MAC statistics counters are inaccurate when operating in half-duplex

Description

When operating in half-duplex, there are many MAC statistics counters that may report inaccurate counts. The magnitude of the error in the inaccurate counters depends on the total number of half-duplex deferral and collision events. Below are the lists of potentially inaccurate counters, separated by Rx and Tx. List items using "PMa_" apply to both the express MAC and preemptible MAC. Otherwise, "PM0_" refers to only the express MAC (EMAC) while "PM1_" refers only to the preemptible MAC (PMAC).

Rx Statistic Errors:

1. PMa_RFCSn increments for Rx fragments smaller than 18B (half-duplex collision fragment)

2. PMa_RERRn increments for Rx fragments smaller than 18B (half-duplex collision fragment)

Tx Statistic Errors:

1a. PM0_TDFRn increments for EMAC Tx packets with retry attempts after non-zero backoff delay

1b. PM0_TDFRn increments for PMAC Tx packets with deferral or retry attempts after non-zero backoff delay

1c. PM1_TDFRn increments for PMAC Tx packets with retry attempts after non-zero backoff delay

1d. PM1_TDFRn increments for EMAC Tx packets with deferral or retry attempts after non-zero backoff delay

2. PMa_TFCSn increments for EMAC or PMAC Tx packets with other deferral or collision events

3. PMa_TERRn increments for EMAC or PMAC Tx packets with deferral and/or collision events with successful re-transmission

4a. PM0_TPKTn double-counts EMAC Tx packets with deferral or collision events

4b. PM0_TPKTn increments for PMAC Tx packets with deferral or collision events

4c. PM1_TPKTn double-counts PMAC Tx packets with deferral or collision events

4d. PM1_TPKTn increments for EMAC Tx packets with deferral or collision events

5a. PM0_TEOCTn double-counts octets for EMAC Tx packets with deferral or collision events

5b. PM0_TEOCTn double-counts octets for the last EMAC packet transmitted when PMAC transmits a packet with deferral or collision events

5c. PM1_TEOCTn double-counts octets for PMAC Tx packets with deferral or collision events

5d. PM1_TEOCTn double-counts octets for the last PMAC packet transmitted when EMAC transmits a packet with deferral or collision events

6a. PM0_TMCOLn increments for PMAC Tx packets with multiple collisions

6b. PM1_TMCOLn increments for EMAC Tx packets with multiple collisions

7a. PM0_TSCOLn increments for PMAC Tx packets with single collisions

7b. PM1_TSCOLn increments for EMAC Tx packets with single collisions

8a. PM0_TLCOLn increments for PMAC Tx packets with late collisions

8b. PM1_TLCOLn increments for EMAC Tx packets with late collisions

9a. PM0_TECOLn increments for PMAC Tx packets with excessive collisions

9b. PM1_TECOLn increments for EMAC Tx packets with excessive collisions

10a. PM0_TLCOLn increments for EMAC Tx packets with excessive collisions

10b. PM0_TLCOLn increments for PMAC Tx packets with excessive collisions

10c. PM1_TLCOLn increments for PMAC Tx packets with excessive collisions

10d. PM1_TLCOLn increments for EMAC Tx packets with excessive collisions

11a. The following packet size counters double-count EMAC Tx packets with deferral or collision events:

- PM0_T64n, PM0_T127n, PM0_T255n, PM0_T511n, PM0_T1023n, PM0_T1522n, PM0_T1523Xn

11b. The following packet size counters double-count the most recently transmitted EMAC packets when PMAC packets transmit with deferral or collision events:

- PM0_T64n, PM0_T127n, PM0_T255n, PM0_T511n, PM0_T1023n, PM0_T1522n, PM0_T1523Xn

11c. The following packet size counters double-count PMAC Tx packets with deferral or collision events:

PM1_T64n, PM1_T127n, PM1_T255n, PM1_T511n, PM1_T1023n, PM1_T1522n, PM1_T1523Xn

11d. The following packet size counters double-count the most recently transmitted PMAC packets when EMAC packets transmit with deferral or collision events:

PM1_T64n, PM1_T127n, PM1_T255n, PM1_T511n, PM1_T1023n, PM1_T1522n, PM1_T1523Xn

Workaround

In general, in is not possible to recover fully accurate counts as a result of this issue. Some counters can be partially recovered as described below:

1. PMa_TEOCTn cannot be reliably recovered, use PMa_TOCTn instead.

2. PMa_TFCSn is partially recoverable by subtracting the sum of the single, multiple, late, and excessive collision counters:

PM0_TFCSn' = PM0_TFCSn - (PM0_TMCOLn + PM0_TSCOLn + PM0_TLCOLn + PM0_TECOLn)

PM1_TFCSn' = PM1_TFCSn - (PM1_TMCOLn + PM1_TSCOLn + PM1_TLCOLn + PM1_TECOLn)

Some error will remain in PMa_TFCSn' after this adjustment. The remaining error will be equal to the number of deferred packets (subset of those counted in PMa_TDFRn) transmitted without collision. If the total number of collisions is low relative to the number of deferred packets, then the following corrections will be more accurate:

PM0_TFCSn' = PM0_TFCSn - PM0_TDFRn

PM1_TFCSn' = PM1_TFCSn - PM1_TDFRn

For this correction, the remaining error will be equal to the number of total collisions that occurred without any deferral before or after the collision.

3. PMa_TERRn is partially recoverable. Use the same correction as described for PMa_TFCSn.

4. PMa_TPKTn is partially recoverable. Use the same correction as described for PMa_TFCSn.

5. PMa_T{64,127,255,511,1023,1522,1523X}n cannot be reliably recovered. The total combined error spread across all counters will be equal to the total error found in PMa_TFCSn as described above.

6. PMa_RFCSn is not recoverable since there's no other way to determine how many small fragments (<18B) have been received due to half-duplex collisions.

7. PMa_RERRn is not recoverable since there's no other way to determine how many small fragments (<18B) have been received due to half-duplex collisions.

ERR051264: NETC: Transmitted verify mPacket for preemption verification can be corrupted in some rare retry scenarios

Description

When preemption is enabled in NETC, up to 3 verify mPackets may be transmitted as part of the preemption verification handshake. The second or third verify mPackets can be corrupted in an extreme cornercase where a valid respond mPacket is received after the initial verify timeout (as configured in MMCSR[VT]), but before the next verify mPacket has been transmitted. If this cornercase occurs, the SMD-V (0x07) octet will be missing from the affected verify mPacket.

This can only occur in the case where a valid respond mPacket has just be received, processed successfully, with preemption capability now verified and enabled. The corrupted verify mPacket is no longer needed since we've already received a valid response and verification is complete. The corrupted verify mPacket should not cause any issues downstream since it is missing the SMD-V octet and also has bad mCRC.

Workaround

No workaround recommended. If necessary, the verify time limit (MMCSR[VT]) can be increased to avoid the possibility of a late arriving respond mPacket.

ERR051300: NETC: NETC engine watchdogs do not operate effectively under all circumstances

Description

TxMacClient watchdog timeout does not account for extra transmission time due to Half Duplex collisions or Transmission Pause

The problem occurs in two cases:

Case 1: MAC works in half-duplex mode and multiple collisions occurs. The TX MAC client watchdog timeout value is not large enough to account for backoff and retry attempts in.

Case 2: MAC receives a pause frame. In the case that the Tx MAC has received a start of frame from NETC but has not yet begun transmission on the line, and MAC enters pause mode, the watchdog on TX MAC client does not similarly pause and could potentially expire.

PCE watchdog timer does not account for hold/release behaviour. If there is a hold window without express traffic, the watchdog can timeout.

Workaround

All NETC watchdogs should be disabled by writing the NETC watch dog timer register (NETCWDTR) to all 1's.

ERR051305: NETC: Entry_id not returned correctly by Stream Gating Instance and Time Gate Scheduling QUERY table management commands

Description

entry_id is provided in the table management command, but not returned correctly in the Query response for the following tables;

- Table ID 36, Stream Gating Instance table:

- Table ID 5, Time Gate Scheduling table:

Entry_id in the response buffer will always be zero.

Workaround

No loss of functionality. Entry_id is not required in the response as it is provided by Software as part of the command request.

ERR051312: SPF2: RAW match is not detected for a load

Description

If the following scenario occurs:

1) External stores over the EDP have been backed up by one of the following:

● AXI backpressure (WREADY/AWREADY=0) for two stores

● PRAW buffers full (lack of bresp) for four stores

And,

2) Another three external stores are issued by the core to the EDP.

Then:

● A RAW match will not be detected for a load (for example, ls/vld/pop/vpop) that matches the last external store in the above sequence, when one of the following is true:

○ There are three or four cycles (nop instructions) between the load and last scalar store (for example, st/push).

○ There are between three and 11 cycles between the load and last vector store (for example, vst/vpush).

Workaround

Do the following:

● For each active region of the DACU with PRAW enabled, Strong Ordering must also be enabled, as follows:

○ If D_ADD0_START[DPRAW] = 1’b1, then set D_ADD0_ATT0[MOM] = 1’b1

○ If D_ADDx_START[INACTIVE] = 1’b0 and D_ADDx_START[DPRAW] = 1’b1, then set D_ADDX_ATT0[MOM] = 1’b1

Notes:

● No action needs to be taken for inactive regions (D_ADDx_START[INACTIVE] = 1’b1).

● No action needs to be taken for regions with DPRAW disabled (D_ADDx_START[DPRAW] = 1’b0) because RAW checking is not supported for this region.

ERR051325: eDMA4: The feature of cancelling the remaining data transfer is not working effectively

Description

eDMA4 does not cancel the current bus activity. Bus transactions will continue normally and complete the transfer size requested by the NBYTES parameter.

Workaround

Do not use the cancel feature.

ERR051326: eDMA4:TCDn_CSR[ESDA] and TCDn_CSR[EEOP] are not functional

Description

If Store Destination Address is enabled (ESDA = 1) and End of Packet Processing is enabled (EEOP = 1), the address stored will be invalid if the EOP signal is triggered. It will not correspond correctly to the last address transferred when EOP is signalled.

Workaround

TCDn_CSR[EEOP] must be disabled.

ERR051327: eDMA4:Scatter-gather feature does not work when NBYTES is not a multiple of 8 bytes

Description

If the current TCD that will initiate the copy has a total transfer size (TCDn_NBYTES_MLOFFYES[NBYTES]) which is not a multiple of 8 bytes, data will not be copied correctly.

Workaround

For cases where the data is 8-byte aligned, use DMA and set NBYTES=8.

For cases where the data is not aligned, use a CPU core to conduct the transfer instead.

ERR051336: eDMA4: Swap feature with 8-bit swap size and 16-bit transfer size does not work

Description

The DMA SWAP function that allows software to select data portions to be swapped between the read data to the write data is not working in the following cases:

1. CHn_CSR[15:12]=0001 and SSIZE=001

2. CHn_CSR[15:12]=1001 and DSIZE=001

Workaround

Do not use SWAP function for the described cases.

ERR051338: NETC: Table 16 (L2 IPv4 Multicast Filter) returns inaccurate KEY_TYPE field

Description

The Table ID 16: L2 IPv4 Multicast Filter (IPv4MF) table management command is used to for both ASM and SSM entries. To differentiate between the two types, or to specify the type when adding, there is a KEY_TYPE field in the KEYE_DATA element. This KEY_TYPE field is returned as part of a Query operation however it may not always be returned accurately when ACCESS_METHOD is Entry ID Match or Search. Note if the ACCESS_METHOD is Exact Match Key Element Match then the KEY_TYPE is provided as part of the command and is accurate in the Query response data.

Workaround

When querying the KEYE_DATA (KEY_TYPE, IPV4_SRC_ADDR, IPV4_DEST_ADDR) from a table management command doing a Query operation (using ENTRY_ID or SEARCH), the issue of an additional command is required using the returned KEYE_DATA:

- issue query with ACCESS_METHOD set to Exact Match Key Element Match(KEY_ELEMENT)

- Use the IPV4_SRC_ADDR and IPV4_DEST_ADDR returned from the previous command

- Set the KEY_TYPE field to SSM

- If command returns with a match, the entry has a KEY_TYPE of SSM.

- If the command returns without a match, the entry has the KEY_TYPE of ASM.

ERR051364: NETC: Cut-through only frames incorrectly incrementing drop counts related to MSDU check

Description

During the Ingress Packet Processing (IPP) stage, for each forwarding destination of a frame, it is determined (independently) if the frame is forwarded as cut-through or as store-and-forward. Cut-through copies of a frame are not intended to be subject to a MSDU check. The MSDU check is only applied to store-and-forward destinations, and if the check fails, then frames destined to those destinations are discarded. If such a discard occurs, then the count in PRXDCR will increment and the MSDUEDR flag will be set in PRXDCRR0. If enabled, the MSDU_DROP_COUNT statistic in the Ingress Stream Count (ISC) entry associated to the frame will also be incremented.

In the case when a frame only has cut-through destinations, the MSDU is erroneously checked. If the MSDU check fails in this case, the above registers and ISC entry statistic will be updated incorrectly.

Workaround

In order to ensure the accuracy of PRXDCR, PRXDCRR0 and MSDU_DROP_COUNT, the MSDU checking has to be disabled if there is the possibility of cut-through transfers in that stream.

The MSDU check is disabled in the Ingress Stream, and Ingress Stream Filter table entries.

ERR051367: NETC: UDP checksum bytes may be corrupted in transmitted PTP packets with one-step timestamping enabled

Description

In some rare cases, the UDP checksum bytes may become corrupted in transmitted PTP packets with one-step timestamping enabled. The corruption occurs only when specific combinations of values are present within the sub-fields of the initial or adjusted 48-bit correction field.

Workaround

1. Use GPTP (PTP over Ethernet).

2. Corruption will never occur provided the upper 16 bits of the initial 48-bit correction field are 0.

3. Receiver should discard any PTP packet with corrupt checksum.

ERR051380: LFAST: LFAST PLL Start-up Marginality

Description

Corner cases have been discovered where LVDS Fast Asynchronous Serial Transmission (LFAST) PLL may not start following the PLL configuration sequence described in the device reference manual.

This is caused by a non-optimal initial control voltage to the Voltage Controlled Oscillator (VCO) inside the LFAST PLL

Workaround

To ensure the correct start-up of the VCO and the LFAST PLL, the user must perform the following sequence while initializing the LFAST PLL. This sequence replaces the PLL configuration example in the LFAST chapter of the device reference manual:

1. Make sure PLL is off by writing:

PLLCR[SWPOFF] = 0x1

2. Prepare to discharge Vctrl to 0V by writing:

PLLCR = 0xC000_xxxx (where x is the user configuration of register fields, FDIVEN must be 0b)

a. If 28 <= FBDIV <= 31, write PLLCR[LPCFG] = 01b

b. If 21 < FBDIV < 28, write PLLCR[LPCFG] = 00b

c. If 16 <= FBDIV <= 21, write PLLCR[LPCFG] = 11b

FBDIV values outside above ranges are not to be used. Note that the restrictions relate to the register value of FBDIV. For example, the real value 32 is allowed, since 32 is programmed within FBDIV as 31 (0x1F).

3. Begin discharge of Vctrl by turning on the PLL by writing:

PLLCR[SWPON] = 0x1

4. Wait for 10us to allow full discharge of Vctrl to 0V

5. Enable VCO startup with optimal Vctrl control voltage by writing:

PLLCR = 0xE000_xxxx (where x is the user configuration of register fields, FDIVEN must be 0b)

a. If 28 <= FBDIV <= 31, write PLLCR[LPCFG] = 01b

b. If 21 < FBDIV < 28, write PLLCR[LPCFG] = 00b

c. If 16 <= FBDIV <= 21, write PLLCR[LPCFG] = 11b

FBDIV values outside above ranges are not to be used

6. Wait for 10us to allow VCO to stabilize

7. Switch to functional mode by writing

PLLCR = 0x0000_xxxx (where x is the user configuration of register fields, FDIVEN must be 0b)

8. Set PLLCR[LPCFG] = 00b to return charge pump setting to default which may have been modified above

The user must wait for:

1. PLLLSR[PLLDIS] to become 0, indicating that the PLL is enabled.

2. PLLLSR[PLDCR] to become 1, indicating that the PLL is locked and ready for use.

After initialization sequence is complete, PLLCR must not be modified unless PLL is reinitialized following the entire sequence.

For scenarios where the software running on the slave device cannot be changed, this workaround procedure can also be achieved using SIPI writes from the Master device only. Upon establishing low speed communications and SIPI being initialized on the slave side, the user can then configure SIPI on the master side to perform the following write sequence to the Slave target (where x is the user configuration of register fields, FDIVEN must be 0b):

1. PLLCR write (0xC000_xxxx)

2. PLLCR write (0xC002_xxxx)

3. PLLCR write (0xC001_xxxx)

4. Wait 10 us

5. PLLCR write (0xE000_xxxx)

6. Wait 10 us

7. PLLCR WRITE (0x0000_xxxx)

8. PLLCR WRITE (0x0000_ xxxx & 0xFFFF_9FFF)

After completion and the PLLs are active on both master and slave side, the rest of the normal startup procedure can be followed as per the reference manual.

ERR051398: NETC: FLR or transmit disable may cause frame transfers to underrun in MAC resulting in bad frame transmission

Description

Assertion of FLR or transmit disable (POR[TXDIS]=1) on a given port would result in halting frame transmissions and discard any frames queued in egress queues (aka flush) for that port. This flush action interferes in the frame transmission of other ports, and may cause MAC underrun on these other ports.

Workaround

1.For each egress class queue used on the transmit port to be disabled

a.Configure its congestion group to enable tail drop (with corresponding threshold set to 0) for each DR level used b.The above is achieved by issuing an update operation against a Congestion Group table entry; Each entry corresponds to a congestion group on given switch port. Each class queue within a port is mapped to a single congestion group. Entry ID:

-Bits 3-0: Congestion Group ID; by default each class queue is mapped to a separate congestion group, class queue 0 map to congestion 0, class queue 1 map to congestion 1, and so on

-Bits 8-4: Switch Port ID

2.Wait for PMa_IEVENT[TX_EMPTY] to be set. If preemption is enabled, wait for TX_EMPTY from both PM0 and PM1.

3.Set POR[TXDIS] to 1b,

4.Disable the congestion group tail drop(s) that were enabled in step 1

5.Wait 64 byte time for Tx packet transmission to complete.

6.Write PMa_COMMAND_CONFIG[TX_EN]=0b.

ERR051400: NETC: Energy Efficient Ethernet (EEE) not supported in MII mode

Description

Energy Efficient Ethernet (EEE) is not supported in MII mode. The Low Power Idle (LPI) signal is not properly sent to the PHY to support EEE. After setting PM0_COMMAND_CONFIG[TX_LOWP] or PM0_SLEEP_TIMER, the PHY does not receive the expected LPI request when in MII mode.

Note that EEE is not defined in the RMII spec. RMII EEE is not supported by NETC.

Workaround

1. Do not use PM0_COMMAND_CONFIG[TX_LOWP] or PM0_SLEEP_TIMER EEE controls in MII mode.

ERR051401: NETC: Corrupt packets may be transmitted after receiving verify mPacket for preemption capability verification

Description

NETC may receive one or more verify mPackets from the link partner as part of the preemption capability verification process defined in 802.3 Clause 99. After receipt of a valid verify mPacket, a respond mPacket is transmitted to the link partner to complete the preemption capability verification handshake. In the case where Tx traffic has been started on the preemptable MAC (pMAC), and a verify mPacket is then received from the link partner, NETC may transmit one or more of the following types of corrupt packets with bad FCS:

1. Respond mPacket
2. Continuation fragment of a preempted frame
3. Start fragment of a preemptable frame

Workaround

1. Disable preemption verification for the link partner.
2. Wait for link partner to verify preemption capability before starting pMAC traffic.

ERR051402: NETC: MAC Merge stat inaccurate after lost Rx fragments

Description

When frame preemption is enabled, MAC Merge stat counter MAC_MERGE_MMFAECR may record an inaccurate count of Rx frame assembly errors in some scenarios where continuation fragments are lost and/or received out of order.

Depending on the exact sequence of fragments skipped and/or reordered, the MAC_MERGE_MMFAECR stat counter may be higher or lower than expected.

Workaround

There is no functional impact to frame assembly. Expect and/or allow small discrepancies in the MAC_MERGE_MMFAECR counter if any fragments are lost or received out of order.

ERR051411: NETC: Time Gate Scheduling (IEEE 802.1Qbv) for MII half-duplex operation

Description

Time Gate Scheduling (IEEE 802.1Qbv) does not work properly when enabled on ports configured for MII half-duplex operation. The scheduler will be over-conservative when scheduling traffic on ports configured for MII half-duplex operation, resulting in significantly less throughput than expected.

Workaround

Choose RMII or RGMII mode when time-gating is enabled (PTGSCR[TGE] = 1) for half-duplex operation (PM0_IF_MODE[HD] = 1).

ERR051413: Precise time-specific-departure operation requires zero_lookahead=0

Description

The IERB default setting of the EaTGSLR[ZERO_LOOKAHEAD] time-based scheduling attribute for ENETCs is 1 (i.e. a lookahead time of 0 ns is used).

Using a lookahead time of 0 ns causes imprecise TSD operation on the pseudo-link. Frames will transmit later than their specified departure time and with higher jitter than is possible with a proper lookahead value configured.

Because the ZERO_LOOKAHEAD value is configured in the IERB space, it may not be possible for run time software to change its setting. This effectively makes it impossible to both obtain the benefit of a zero-lookahead time and perform precise TSD.in a given configuration.

Workaround

In an application where the IERB register space can be accessed during operation (i.e. IERB access can be unlocked), the ZERO_LOOKAHEAD bit can simply be cleared to avoid the issue.

Alternatively, a gate control list that maintains all traffic class gates in an open state can be loaded, which will clear the EaTGSLR[ZERO_LOOKAHEAD] bit and allow precise TSD to be performed. The EaTGSLR[ZERO_LOOKAHEAD] bit is cleared when the ENETC is configured with a Qbv gate control list.

ERR051432: eDMA4: DMA channel may not terminate correctly if a bus error occurs during transfer.

Description

During a DMA transfer, if a bus error is returned on a read or a write, the following actions are expected:

(1) The channel execution should terminate immediately.

(2) The last address (which caused the error) should be written back to the RAM TCD SADDR and DADDR.

(3) The error should be reported to the DMA status registers CHn_ES[SBE:DBE].

The issue is that steps 1 and 2 above may not be performed correctly. Channel execution could continue and/or the last address stored in TCD SADDR/DADDR could be offset from the correct address. Step 3 does occur correctly and DMA records the event in CHn_ES[SBE:DBE] and MP_ES[SBE:DBE] registers.

Workaround

Software can check for bus error events via DMA CHn_ES or MP_ES, but it should not rely on the TCD information at the end of the transfer to determine the bus address on which the error occurred. Software should consider the entire transfer may be incorrect and, if needed, reinitialize and retry the TCD execution.

ERR051441: RTU: Privileged accesses to the peripherals cannot be controlled by the XRDC MDACs

Description

RTU CR52 accesses to the peripherals on the Low-Latency Peripheral Port (LLPP) will control the privilege bit attribute that is driven on the bus. The LLPP MDAC8, MDAC9, MDAC10, MDAC11 are ineffective in changing this attribute.

Workaround

Software must treat the R52 cores as the sole masters for the privilege mode when accessing the peripherals.

ERR051444: NETC : ICM discards when two priority queues are in use can cause potential memory loss

Description

- Issue is related to the Ingress ENETC Ingress Congestion Manager (ICM) which provides the main internal buffering/queuing point on the ENETC receive datapath.

- At the ICM queuing point, frames can be optionally arranged by their schedule priority (priority queuing) and drop resiliency (drop resiliency queuing).

- When priority queuing is used (2 priority levels supported), a frame with high priority is scheduled to HTA Receive before a frame with low priority. Within a priority, first-in-first-out (FIFO) ordering is followed.

- When drop resiliency queuing (4 levels) is used, under congestion (queue buildup), a frame with lower drop resiliency is dropped first before frames with higher drop resiliency. Within a drop resiliency level, first-in-first-out (FIFO) ordering is followed, with dropping occurring at the head of the queue.

- Both priority and drop resiliency queuing are optional and not used when ENETC comes out of reset.

- Issue occurs when priority queueing is used, and ICM becomes congested, and under a combination of rare events, internal buffering memory is depleted/loss when frames are being dropped to relieve congestion.

Workaround

To avoid this defect, ICM priority queueing should not be used; always use single priority by keeping the Receive IPV to ICM priority mapping register 0 (IPV2ICMPMR0) setting to its power on reset value.

ERR051449: FlexCAN: Node (connected with CAN HUB) accepting messages without TX ACK

Description

When FlexCAN is connected though CAN HUB in receiving mode, FlexCAN will accept messages even if there is no ACK in ACK slot.

Workaround

No workaround is possible when using CAN HUB on this device. Software must be able to tolerate this limitation.

ERR051457: GTM: ARU data trace message does not differentiate between channel 0 and channel 1

Description

GTM’s Advance Routing Unit (ARU) module provides two independent debug channel interfaces. The CHN bit of the ARU data trace message should indicate whether the trace message is for channel 0 or channel 1. This bit should be ‘0’ for channel 0 and ‘1’ for channel 1. Instead, this bit is tied to ’0' for both channels of ARU. Hence, messages for ARU channel 1 will incorrectly identify the channel as 0.

Workaround

Each ARU channel can be configured with a dedicated read address in register ARU_DBG_ACCESS0 and ARU_DBG_ACCESS1 of the GTM module which would eventually transfer ARU data on each respective debug channel interface. To avoid the issue described in this erratum, configure only ARU channel 0 or ARU channel 1 to debug all ARU data. Do not configure both channels for operation. Accordingly, configure WMC1 or WMC2 bit of GTMDI module for generation of watchpoint messages for the respective ARU channel 0 or ARU channel 1. Similarly, configure DMC1 or DMC2 bits of GTMDI module for generation of data trace messages for the respective ARU channel 0 or ARU channel 1.

ERR051460: NETC: Ingress Port Filter (Table ID 13) can overflow rsp buffer during multi-entry query management command

Description

The Ingress Port Filter table which is implemented using a Ternary Content Addressable Memory (TCAM), supports a search query-operation command (ACCESS_METHOD=Search) which is used to return either the “Entry ID” (QUERY_ACTIONS=1) or the full content (QUERY_ACTIONS=0) of every entry in the table. When the full content of every entry in the table is queried (QUERY_ACTIONS=0), the command may return more response data than will fit in the allocated buffer if one or more entry is compressed and the response buffer size is too small to accommodate all query entries. The extra data will be written beyond the end of allocated buffer.

Compression is used to reduce the amount of TCAM space required to store each table entry. A narrow (48-bit wide), but deep TCAM organization is used, where a table entry can occupy multiple sequential lines of the TCAM. TCAM lines for which all fields are masked out (their corresponding mask field set to all 0s) will not actually exist in the TCAM, thus what is being referred above as compression

Workaround

Option A:

Issue a query table management command, with ACCESS_METHOD=Search and QUERY_ACTIONS=1. This returns only the list of entry IDs currently in use.

For each entry ID returned, issue a query table management command, with ACCESS_METHOD=Entry ID Match and QUERY_ACTIONS=0. This returns the full entry data of each entry.

Option B:

Allocate a large enough response buffer to fit all potential entries. The response buffer needs to be 4+(N*232) bytes, where N is the max possible number of entries (96 entries in RT1180, 64 entries in S32Z1/Z2/E2 devices).

ERR051476: CANEXCEL: Performance Limitation on Reception side

Description

CANEXCEL can drop frames while receiving back-to-back frames if the number of Receive Message Descriptors is more than 64 and the number of XL Receive Message Descriptors is more than 32 with a Nominal baud rate greater than 250 Kbps. In this scenario CANEXCEL correctly notifies an RX SMB Overrun by setting SIC.SYSS[CRXOERR] = 0b1,

Workaround

It is a performance bottleneck. Use one of the following three options as a workaround:

Common Configuration:

- CHI Clock = 267 Mhz.

- XL Data Baud-rate configured up to 16 Mbps.

- The number of Tx and Rx MD’s are 64 (GRP_CTRL.DSCCTRL[TXDSC] = 63).

- TX Message Descriptor count (SIC.SYSMCFG[MAXTXMB]) <= 64.

- Non used RX-MD ID filtering criteria should be configured as NO MATCH

Option-1: Nominal Baud-rate configured up to 1 Mbps

- Only XL frames

- Configure one CANEXCEL instance as Tx only node and Other instance as Rx only node. The RX node needs to configure the related filter to filter out the frames transmitted from same board’s Tx node.

- RX Message Descriptor count (SIC.SYSMCFG[MAXRXMB]) <=32.

- Non-Overlapping ID’s configured for both RX MD and RXFIFO - For each MD, only ID filtering criteria is such that it is unique with respect to other MDs (non-overlapping ID filtering configuration including Mask/Range for each MD), such that the received frame has a match in either single MD or RXFIFO.

- Filter Bank-1 is disabled (SIC.SYSMCFG[FB1EN] = 0’b0).

Option-2: Nominal Baud-rate configured up to 500 Kbps.

- RX Message Descriptor count (SIC.SYSMCFG[MAXRXMB]) <=64.

- Up to 4 Receive Message Descriptors can have overlapping ID filtering criteria out of all configured Receive Message Descriptors.

Option-3: Nominal Baud-rate configured up to 250 Kbps.

- RX Message Descriptor count (SIC.SYSMCFG[MAXRXMB]) <=64.

- Maximum Reception MD’s for XL Message Descriptor can be <= 32.

ERR051484: AES_ACCEL: DMA2_DID_ERR incorrectly asserted

Description

When accessing the AES Accelerator (AES_ACCEL) via an application core, the error flag DMA2_DID_ERR may be set on a subsequent access by the Result DMA.

Workaround

After accessing AES_ACCEL via an application core, access the AES_ACCEL again via the Result DMA and clear the DMA2_DID_ERR flag, if it has been set

.

Alternatively: solely access the AES_ACCEL registers using the Result DMA

ERR051485: Key property flag DEBUG should not be used

Description

The DEBUG flag for the keyslots in the ACE keystore should bet set to '0'. This bit is reserved and on a future device will be used for a ‘NO DEBUG’ flag.

Workaround

n/a

ERR051487: CANEXCEL: Data RAM Error Report Syndrome does not function properly

Description

The Data RAM Error Report Syndrome (DRRERRSYN.BEn) bit is driven by the result of OR-ing of syndrome bits (DRRERRSYN.SYNDn) instead of byte enable value requested in 32bit read transaction

Workaround

The information in DRRERRSYN cannot be used as the byte enables do not properly reflect which bytes were accessed when the error occurred.

ERR051488: CANEXCEL: Keep latest feature in RXFIFO does not work

Description

RXFIFO does not work in following senarios when keep latest feature is enabled (RXFFCTR[KEEPLST] bit is set) :

1. Incorrect pointer (RXFCSTA[SYSPOINTER] and RXFCSTA[HWPOINTER]) and state (RXFCSTA[STATE]) notification may occur in RXFIFO when CANEXCEL pushes (RXFHWPUSH[RXPUSH]) on keep latest pointer and system performs the pop (RXFSYSPOP[RXPOP]) operation. This may occur when RXFIFO goes into OVERRUN state (RXFCSTA[STATE]).

2. If RXFIFO is in OVERRUN state (RXFCSTA[STATE]), system write one to clear RxFIFO Error Flag (RXFS[RXFEF]), RXFS[RXFEF] does not get cleared.

3. If RXFIFO is in OVERRUN state (RXFCSTA[STATE]), hardware pointer RXFCSTA[HWPOINTER]) updates incorrectly.

Workaround

System must not enable the keep latest feature. The only legal value allowed for RXFFCTR[KEEPLST] is 0b0.

ERR051489: CANEXCEL: When a Message Descriptor is locked by the system, the State transition to OVERRUN does not function properly

Description

When a Message Descriptor is locked by the system (DCSYSLOCKn[SYSLOCK] bit is set) and the system pushes (DCSYSPUSHn[PUSH]) greater than the Message FIFO Depth (FIFOCTRLn[FIFODPHn]) , the Message Descriptor does not transition to OVERRUN State (DCSTAn[STATE])

Workaround

Either the system must not push (DCSYSPUSHn[PUSH]) greater than the Message FIFO Depth or the system must configure Message FIFO Depth FIFOCTRLn[FIFODPHn] = 0b1111

ERR051495: NETC: Stale context in policer can result in Policer dropping frames

Description

The rate policer is an optional function that is used to control the amount of traffic (or bytes) that is allowed to go through for a particular received flow of frames. Specifically, it monitors a received flow of frames, and if the incoming rate of this traffic exceeds the maximum configured rate (i.e., policing rate), the excess traffic is dropped (or remarked).

The rate policer is implemented using the token-bucket algorithm and relying on the arrival times of frames to compute the number of tokens to be added to the bucket. A given rate policer instance keeps track of the time (timestamp) of the last frame received and updates the number of tokens in a bucket only when a new frame arrives, computing the number of tokens to add, as the elapsed time between the stored timestamp and the timestamp of the frame currently being received, multiplied by the policer bandwidth rate. A 32 bits timestamp is used (i.e. least significant 32 bits of the 1588 timer or default nanosecond timer. Because the 32-bit timestamp covers a time window of approximately 4 seconds, a refresh mechanism is implemented to update the token bucket in case there are no frames received during the 4 seconds time window. Thus the implementation updates timestamp upon a frame arrival event or refresh event.

There are two issues in the Rate policer :

Issue 1 -

The refresh mechanism is a periodic event that occurs approximately every 0.5 seconds to ensure that the stored timestamp value does not go stale in case of a long idle time between frame arrivals. The issue is that the timestamp captured can be erroneous if a rate policer instance is enabled (add or update operation against the Rate Policer table entry) during certain periods of time and if a refresh event is to occur before a frame arrival. As a result of stale timestamp capture, the policer becomes dysfunctional indefinitely.

Issue 2:

Issuing an update operation that only enables/disables a rate policer instance (i.e. Functional Enable Element Update (FEEU) action) doesn’t work properly.

Workaround

Workaround for issue 1:

The policer can work correctly if the policer can be configured and becomes operational within a certain period of time. That is if an add or update action is specified to enable a policer needs to be issued after least significant 32 bits of the timer is 0 seconds and must be executed by hardware no later than 1.60 seconds. (i.e. when TIMER_SRT_L/TIMER_DEF_CNT_L is between 0x0 and 0x5FFF_FFFF). To leave some margin it is recommended that the time window is kept between 0 and 1.34 seconds (i.e. between 0x0 and 0x5000_0000).

Note that the timer can be read in timer register space TIMER_SRT_L/TIMER_DEF_CNT) or in the ENETC SI register address space - Read to the SICTR0 register followed immediately by a read to the SICTR1 register.

Workaround for issue 2:

A functional disable operation can be achieved by issuing an update operation with the action to update the configuration element (i.e. Configuration Element Update (CFGEU) action). In other words, re-configuring the rate policer instance. After an update to the configuration element, the rate policer instance is disabled.

A functional enable operation can be achieved by issuing an update operation with the following two update actions set at the same time

- Updating the configuration element (i.e. Configuration Element Update (CFGEU) action). In other words, re-configuring the rate policer instance, and

- Updating the functional enable element (i.e. Functional Enable Element Update (FEEU) action) to enable the rate policer instance

ERR051524: NETC: Ingress Stream Identification Payload construction evaluates incorrectly for frames >1kB

Description

The Ingress Stream Identification table (Table ID 30) lookup key is built using key construction registers. Payload data can be optionally included to the lookup key.

When present, the payload key field is taken from an offset relative to the Payload Ethertype. Payload Ethertype starts immediately after source MAC address or VLAN tag(s)/RTAG/HSR (if any). The payload key field must be within the frame, and within 116 bytes from the Payload Ethertype. Hardware checks these two conditions before constructing the Ingress Stream Identification table lookup key.

This check may evaluate incorrectly (indicating an invalid key construction) when the frame is received from a pseudo port (internal port) bound to ENETC or the switch, and is 1024 bytes or larger in size.

Workaround

Disable payload as part of the key construction if being used on a pseudo port (bound to ENETC or switch), when maximum frame sizes is expected (or set) to be 1kB or larger.

ERR051530: NETC: hash multi-entry search table management continuation command can miss returning entries

Description

NETC implements a common hash space that is used for multiple tables, across functions. The hash table management commands support a multi-entry Search command (ACCESS_METHOD = Search). The Search command completes when the full hash space has been searched, or the response buffer has been filled with matching entries. For the latter, Hardware returns a resume entry ID which Software uses to continue the Search command.

When the resume point of a continued Search command is in the middle of a collision chain (i.e. the last entry returned in the previous Search command segment, that filled the response buffer, was not the last entry in the collision chain), and the resume point cannot be found, Hardware may not return the matching entries of the next valid collision chain.

Workaround

OPTION A:

- provide a large enough buffer to hold all entries, avoiding the need to resume.

OPTION B:

- When resuming a search command, zero the lower 4 bits of the returned resume entry ID, and issue a resume search. NOTE: this may return some duplicates overlapping with the end of the previous search, up to HBTCAPR[MAX_COL] entries.

ERR051549: CANEXCEL: Message descriptor overrun error bit does not function properly

Description

Message descriptor overrun error SYSS[CMDOERR] bit is not being set in case of message descriptor state DCSTn[STATE] transition to OVERRUN . During the internal arbitration process if the CANEXCEL finds any message descriptor state DCSTn[STATE] in OVERRUN then it sets the overrun error SYSS[CMDOERR]. If no push occurs after a Message Descriptor is moved to OVERRUN, then the SYSS[CMDOERR] bit is not set.

Workaround

Software should ensure to not push more than available pointers.

ERR051551: CANEXCEL: Access to register CANXL_TBS is not protected by PDAC

Description

There is no Peripheral Domain Access Control (PDAC) to control access to register CANXL_TBS.

Workaround

Safety checking software should conduct periodic reads of CANXL_TBS to ensure integrity of this register, if desired.

ERR051557: ATP: Possible Aurora PLL jitter spec violation due to VDD_IO_A_J_POR edge aligned activity

Description

Aurora Trace Port (ATP) PLL can exceed jitter specs when pads on VDD_IO_A_J_POR supply are configured as output and toggle in an edge aligned fashion. With random IO activity, there is no spec violation with Aurora PLL Jitter

Workaround

Ensure that no more than 2 pads in the VDD_IO_A_J_POR domain are configured as outputs which toggle in an edge aligned manner i.e multiple (3+) pins from the same module configured to the same frequency which toggle at the same time.

Also, Ensure sufficient isolation between VDD_IO_AJ_POR and Aurora PLL supply on the board

ERR051561: CMU: Automatic Fault Injection mode does not work properly

Description

If Automatic Fault Injection mode is enabled (GCR[FI_AUTO_EN]=1) after any other mode (functional or standalone mode), then it does not give correct output.

Workaround

Do not use Automatic Fault Injection mode. Use Standalone Fault Injection mode to cover safety checks of the module.

ERR051587: NETC: Time gate scheduling update command response can be erroneous if the AdminBaseTime specified is near the current time

Description

Under the following circumstances, the Time Gate Scheduling Table may not execute configuration commands with the update action correctly:

(1) If an operational gate control list is installed and active (I.E. PTGAGLSR[TG] corresponding to the port is 1), and the ADMIN_BASE_TIME specified in the command is in the range [current time, current time + advance time + the command processing duration + the minimum duration before admin gate list installation] then the command may unexpectedly fail and return the error code “0x0D6” instead of succeeding.

(2) If an operational gate control list is installed and active (I.E. PTGAGLSR[TG] corresponding to the port is 1), and the ADMIN_BASE_TIME specified in the command is in the range (current time – 2*2^30 nanoseconds, current time), then the command may unexpectedly succeeds instead of failing and returning the error code “0x0D6”

Where “advance time” is:

The duration that the time reference used by HW scheduler is moved forward to adjust for latency encountered in the transmit processing pipeline (~ 10 usec for ENETC,   0. 1 usec for the switch)

Where “current time” is:

The timestamp of the 1588 (synchronized) timer at the moment the driver (software) begins to set up and dispatch the command.

Where “the command processing duration" is:

The sum of the duration it takes software to set up and dispatch the command, and the duration it takes hardware to process the command.

Where "the duration it takes software to set up and dispatch the command" is:

The sum of the durations it takes to setup the CBD command, dispatch the CBD command, and complete the CBD command by updating the BDR producer index.

Where "the duration it takes hardware to process the command" is:

(300 + (7 * ADMIN_CONTROL_LIST_LENGTH) + (5 * (ADMIN_CONTROL_LIST_LENGTH)^2)) * the NETC platform clock period

Where “the minimum duration before admin gate installation” is:

The sum of the cycle duration of the active operational gate list and the duration remaining in the current gate list cycle.

Workaround

If an operational gate control list is installed and active (I.E. PTGAGLSR[TG] corresponding to the port is 1), and the ADMIN_BASE_TIME that would be specified in an update command is within the range (current time – 2*2^30 nanoseconds, current time + advance time + the command processing duration + the minimum duration before admin gate list installation), then software must increase the ADMIN_BASE_TIME that would be specified in increments of ADMIN_CYCLE_TIME until it is no longer within that range.

Functionally this workaround has minimum impact on the config change time computation performed by hardware (time at which the admin gate control list is installed to become operational). With the workaround the “minimum time before admin gate control list can be installed as the operation list” is increased

From: time left to end of the current operation cycle + one entire operation cycle time

To: 2 × operational cycle times

ERR051589: LMEM64: CM33 core can have coherency issues when accessing the LPDDR region

Description

When LMEM64 cache is enabled in write-back mode, cache write-back operations to the LPDDR address space will bypass the Last Level Cache (LLC), updating physical memory without updating an entry in the LLC. This may cause incorrect data to be read back from the same address later since the LLC is no longer coherent with the underlying physical memory.

Workaround

Configure the LPDDR address range for write-through operation in the CM33 MPU. Do not use write-back mode for this address space.

ERR051590: CSTCU/RGM: Destructive reset during STCU self-test causes extraneous FES[ST_DONE] assertion

Description

In the case of destructive reset while STCU is running, the STCU still asserts Functional Reset request to RGM. Since RGM knows that it is in Destructive reset state, it ignores this Functional Reset request, but this gets latched in FES[ST_DONE] bit.

Workaround

Software should ignore the RGM_FES[ST_DONE] bit after self-test execution.

ERR051591: CM33: Bus faults are not reported in some cases when LMEM cache is enabled

Description

Bus errors on instruction or data fetches issued by the Cortex-M33 may not be notified precisely. On burst reads which result in an error response, the Cortex-M33 does not receive the fault until the last payload regardless of which payload within the burst resulted in the error. Burst reads may occur when the LMEM cache is enabled for the address being accessed.

Workaround

Configure the Cortex-M33 MPU to disallow access to reserved memory regions and avoid the error response.

ERR051597: CM33: MCM_FATR detects wrong attributes in cache write buffer error

Description

When a cache write buffer error is generated when a cache write access is made on xRDC protected memory location, the following attributes are wrongly detected in MCM_FATR register:

1. BEWT : Observed value is 0 but expected is 1
2. BEDA : Observed value is 0 but expected is 1

Workaround

Write software to expect opposite polarity for FATR bit[7] and bit[0] values as compared to Reference Manual description.

ERR051607: NETC: NETC PF may override VF bus master enable and MSI-X function mask for MSI-X generation

Description

Each PCIe function within NETC allocates its MSI-X table resources from a single larger MSI-X table that serves the entire NETC. An ENETC instance acquires a total table size that includes both PF and VF(s) as defined by IERB register EaMCR[NUM_MSIX], which is split between PF and VF(s) as defined by PSI0CFGR[NUM_MSIX] and PSIaCFGR[NUM_MSIX] (a>0) respectively.

When ENETC PF sets the PCIe MSI-X Function Mask (PCI_CFC_MSIX_MSG_CTL[FUNC_MASK]=1), it will also override the VF function's setting of the same bit, making it appear to the VF MSI-X table as always set. This has the potential side effect of making VF MSI-X interrupt go to the PBA (Pending Bit Array) instead of generating an MSI-X interrupt.

When ENETC PF sets the PCIe Bus Master Enable bit (PCI_CFH_CMD[BUS_MASTER_EN]=1), it will also override the VF function's setting of the same bit, making it appear to the VF MSI-X table as always set. This has the potential side effect of generating an MSI-X interrupt by VF when in fact is has disabled it by setting PCI_CFH_CMD[BUS_MASTER_EN]=0.

Workaround

For an ENETC PF that requires the use of masking an MSI-X interrupt, it should rely on the per-vector masking instead of the global MSI-X table function mask. This is a bit part of the MSI-X table entry VC Control field.

Optionally if the ENETC PF uses the function mask during processing of an interrupt, it should only temporarily set this bit, allowing the VF to log interrupts in the PBA and when PF clear the function mask, VF interrupts will again be generated. This could result in a delay of VF interrupt notification.

ERR051609: I3C: May be unreliable in I3C mode.

Description

The I3C module is intended to operate in either I3C or legacy I2C (Fm /Fm+) protocol modes. Due to internal structural issues, I3C mode functionality is not guaranteed. Legacy I2C mode is fully functional as specified.

Workaround

Do not use the I3C module in I3C mode. Use I2C mode instead.

ERR051613: RTU: GICR_TYPER registers do not reflect affinity configuration inputs

Description

The Cortex-R52 processor has an identifier which is set via the processor top-level configuration inputs CFGMPIDRAFF1 and CFGMPIDRAFF2. These are referred to as the Aff1 and Aff2 identifier fields. The Aff3 identifier field is always zero. These affinity fields are reported through the MPIDR system register for each core and the GICR_TYPER register for each associated Redistributor. Because of this erratum, the GICR_TYPER.Aff1 and GICR_TYPER.Aff2 fields are always zero

Workaround

Software which identifies Cortex-R52 Redistributors by matching affinity values should compare the Aff0 value only and ignore the Aff1 and Aff2 values.

The Redistributors' registers within a Cortex-R52 processor cluster are only accessible to the cores within that cluster and all the Redistributors within a cluster are associated with cores within the cluster or the GIC export port for that cluster. Therefore, it is only necessary to compare the Aff0 values to precisely identify the correct Redistributor.

ERR051614: RTU: DTR flags not cleared on external debugger access while leaving Debug state

Description

The Data Transfer Registers (DTRs) provide a mechanism to transfer data between an external debugger and the core. They consist of write-only registers to transmit data (DBGDTRTX_EL0 and DBGDTRTXint), read-only registers to receive data (DBGDTRRX_EL0 and DBGDTRRXint), and associated data control flags.

Due to this erratum, if these registers are accessed by the external debugger while the debug exit procedure is in progress, then the accesses will go ahead but the flow control flags will not be updated correctly.

Workaround

The external debugger should not access the DTRs after requesting a debug exit until that procedure completes.

ERR051616: NETC: FDB (Table ID 15) always overrides Egress Treatment Table access defined by Ingress Stream (Table ID 31)

Description

Background:

There are 2 methods, via the primary Egress Treatment group assignment, to direct a frame to Egress Sequence Recovery Function used for duplicate frame elimination:

a) using Ingress Stream (IS) table

The ingress stream method to direct to the sequence recovery function is used when stream identifier is encoded in the Ethernet MAC Source address (MAC SA) and VLAN identifier

b) using Bridge method using VLAN and FDB (or IPv4 Multicast Filter) tables.

Can be used in case where the stream identifier is encoded in the Ethernet MAC Destination Address (MAC DA) and the VLAN identifier.

There are 2 methods to forward a frame to egress switch port:

a) Stream forwarding using Ingress Stream and

b) Bridge forwarding using VLAN and FDB (or IPv4 Multicast Filter) tables

Issue:

Using Ingress streams (Table ID 31) to identify Egress Sequence Recovery Function and using Bridging with FDB (Table ID 15) entries (with the Override Egress Treatment Entry ID (OETEID) option field is set to 00b) to forward frame to egress switch ports incorrectly overrides the primary Egress Treatment group assignment (i.e. the Egress Sequence Recovery Function) specified by the Ingress Stream.

This breaks duplicate frame elimination capability provided by Egress Sequence Recovery Function.

This is of particular concern when the Ingress stream is defined using MAC SA and the Bridging method is used to identify egress switch port.

Note: Egress Sequence Recovery is incorrectly overridden only if the bridging (FDB or IPv4 Multicast Filter) lookup finds an entry.

Egress Sequence Recovery is not overridden if FDB (or IPv4 Multicast Filter) entry is not found causing the frame to be flooded.

Workaround

Goal: Avoid having Ingress Stream (IS) specify ET_EID and using FDB entry to forward frame. 2 acceptable workaround.

1) Use IS to specify ET_EID but don't use FDB entry

For each {VLAN, MAC SA} or {MAC SA} stream, add 2 Ingress Stream (IS) entries:

a) IS for terminating flows: Based on {VLAN, MAC SA, MAC DA} with FA=StreamFwd (010b) and destination port being ENETC's Host MAC

b) IS for transiting flows: Based on {VLAN, MAC SA} with FA=Bridging (011b).

FDB entry will flood to all ports except the one received.

Another option: specify FA=StreamFwd with SPPD=0 (source port pruning enabled) & fwd to all ports. This eliminate need to do Bridging and use VLAN filter table.

Both IS entries would have same configuration except for FA field. That is, ET_EID & EGRESS_PORT_BITMAP are the same.

Note: IS for terminating flow has a higher precedence than transiting flows during Ingress Stream lookup (using either EM ISID or IPF TCAM lookup).

Pro: simple to implement.

Con: scalability concern if multiple MAC address needs to be migrated from FDB to IS table (#MAC SA streams * # MAC DA)

Note: For HSR, there would be 1 and maybe 2 MAC Addr. So scalability should not a concern.

Note: HSR implementing this workaround.

2) Use IS to identify internal VLAN which specifies ET_EID entry. FDB used to forward.

a) For each {VLAN, MAC SA}, {MAC SA} stream use a reserved VLAN (eg: VLAN 3000-4095)

b) IS adds interim VLAN (Ingress Frame Modification)

c) VLAN specifies the ET_EID entry responsible for sequence recovery function and removes the interim VLAN.

Pro: address scalability concern of Workaround 1

Con: more configuration needed.

ERR051617: [I3C] In I2C controller mode generates unintended repeated START before sending STOP

Description

When I3C module is used as an I2C controller, repeated START may be randomly generated before STOP. That is, when MCTRL.REQUEST = STOP and MCTRL.TYPE = I2C, a STOP signal may or may not be preceded by a repeated START signal.

Workaround

In I2C compatibility mode, set to MCONFIG[SKEW] = 1

ERR051620: GTM: Interrupt trigger signals CCU0TC_IRQ and CCU1TC_IRQ are delayed by one CMU_CLK period related to the output signals.

Description

Interrupt trigger signals CCU0TC_IRQ and CCU1TC_IRQ are delayed by one CMU_CLK period if the following configurations are used:

1. Both CCU0TC_IRQ and CCU1TC_IRQ are affected (ATOM: in SOMP mode) when the channel is configured in up-down counter mode ((A)TOM[i]_CH[x]_CTRL.UDMODE>0).

2. CCU1TC_IRQ only is affected (ATOM: in SOMP mode) when the channel is configured in up-counter mode ((A)TOM[i]_CH[x]_CTRL.UDMODE==0) and (A)TOM[i]_CH[x]_CTRL.SR0_TRIG is enabled.

Workaround

This feature cannot be used.

ERR051622: NETC: TABLE_BIR of NETC TIMER is wrong

Description

The NETC PCI MSI-X Table Offset/BIR register (PCI_CFC_MSIX_TABLE_OFF_BIR) and PCI MSI-X PBA Offset/BIR register (PCI_CFC_MSIX_PBA_OFF_BIR) include the field BIR which indicates which one of a Function’s Base Address

registers, located beginning at 10h in Configuration Space, or entry in the Enhanced Allocation capability with a matching BEI, is used to map the Function’s MSI-X Table/PBA into Memory Space.

NETC supports the PCIe Enhanced Allocation (EA) capability which replaces the BARs located in the Configuration Space, starting at 10h. The EA Capability uses fixed base addresses and includes the entries described below. Note that the 4-bit BEI value is supposed to be used to match the BIR values to find an entry.

- EA Entry 0 (BEI 0): Physical Function (PF)
- EA Entry 1 (BEI 2): Physical Function (PF) MSI-X Table/PBA
- EA Entry 2 (BEI 9): Physical Function's first Virtual Function (VF) (ENETC only if applicable)
- EA Entry 3 (BEI 11): Physical Function's first Virtual Function (VF) MSI-C Table/PBA (ENETC only if applicable)

This bug affects all NETC physical/virtual functions specifying an MSI-X table/PBA BIR value of 1 or 3, which are not BEI values found in the EA capability entries, only BIR values of 0 and 2 are valid (matches PF BEI 0/2 and VF BEI 9/11). Any standard PCIe driver that tries to find the correct EA capability entry for the MSI-X table/PBA, will not be able to do so by matching the MSI-X BIR value with an EA entry's BEI value.

Workaround

Software can use the MSI-X Table/PBA BIR value as the index to find the Enhanced Allocation entry. For example, if MSI-X TABLE_BIR=1, EA per-entry registers PCI_CFC_EA_PE{BIR}_* would be used.

ERR051630: GTM: ((A)TOM) Changing the output signal level synchronously to the period by writing to (A)TOM[i]_CH[x]_CTRL_SR.SL_SR for 0% duty cycle

with (A)TOM[i]_CH[x]_CM0.CM0=MAX+1 and (A)TOM[i]_CH[x]_CM1.CM1=0 and (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1 not functional

Description

(A)TOM channel is configured to be triggered by a preceding channel with (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1. Additionally the channel is configured to 0 % duty cycle by setting of (A)TOM[i]_CH[x]_CM0.CM0=MAX+1 and (A)TOM[i]_CH[x]_CM1.CM1=0. In this case, the output signal (A)TOM_OUT is set to inverse signal level SL.

Expected behavior: By writing a value to (A)TOM[i]_CH[x]_CTRL_SR.SL_SR, the value of the output signal can be changed synchronously with the period.

Observed behavior: Changing the signal level by writing a value to (A)TOM[i]_CH[x]_CTRL_SR.SL_SR is intermittent not taken into account in the next period on the output signal (A)TOM_OUT.

Workaround

First Workaround is changing the constant signal level of the output signal (A)TOM_OUT synchronously to the period can be achieved by configuring 100 % duty cycle with (A)TOM[i]_CH[x]_CM0.CM0=0 and (A)TOM[i]_CH[x]_CM1.CM1=MAX. In this case, the inverted signal level has to be written to (A)TOM[i]_CH[x]_CTRL_SR.SL_SR to reach the same constant output level on (A)TOM_OUT.

Second workaround is to not change the value of (A)TOM[i]_CH[x]_CTRL_SR.SL_SR but to switch between 0 % duty cycle ((A)TOM[i]_CH[x]_CM0.CM0>MAX, (A)TOM[i]_CH[x]_CM1.CM1=0) and 100 % duty cycle ((A)TOM[i]_CH[x]_CM0.CM0=0, (A)TOM[i]_CH[x]_CM1.CM1=MAX) to get the required signal level on the output signal (A)TOM_OUT.

ERR051631: GTM: (TIO) Interrupts are not cleared by a write access to interrupt mode register

Description

A write access to a register TIO[i]_G[g]_CH[c]_IRQ_MODE was intended to clear the associated IRQ notify bits of registers TIO[i]_G[g]_CH[c]_IRQ_NOTIFY. However, this functionality is not implemented.

Workaround

The software must reset the registers TIO[i]_G[g]_CH[c]_IRQ_NOTIFY manually after a write access to TIO[i]_G[g]_CH[c]_IRQ_MODE.

ERR051632: GTM: (ARU) Delivering data via ARU_ACCESS does not take into account the status of the ARU cluster isolation from the requesting module

Description

It is possible to write (ARU_ACCESS.WREQ) or to read (ARU_ACCESS.RREQ) data to/from each module, which is connected to the ARU via the ARU registers ARU_ACCESS, ARU_DATA_H and ARU_DATA_L.

If the ARU cluster isolation of the cluster from the requesting module is active by setting of GTM_ARU_COM_DIS(k)=1, the read or write request from ARU_ACCESS must not be served.

This works correctly for the ARU_ACCESS read access but not for the write access. The write access will always deliver valid data and does not take into account the status of the cluster isolation from the requesting module.

Workaround

Do not deliver data via the ARU registers ARU_ACCESS, ARU_DATA_H and ARU_DATA_L as long as the cluster of the addressed module (ARU_ACCESS.ADDR) is isolated (GTM_ARU_COM_DIS(k)=1).

ERR051633: GTM: (AXIS) Bridge software reset initiated by writing BRIDGE_MODE.BRG_RST=1 will corrupt AXIS slave protocol

Description

The invocation of the AEI Bridge software reset (writing BRIDGE_MODE.BRG_RST=1), also resets the AXIS transaction ID. Due to this, the GTM_AXIS module is unable to serve the response of this transfer. This AXIS transaction will not be terminated at all.

Software reset is not allowed to reset the AXIS transaction IDs in the GTM_AEI write buffer.

Workaround

Use asynchronous reset AXIS_ARESETN instead of writing BRIDGE_MODE.BRG_RST=1 (synchronous reset).

ERR051634: GTM: (DPLL) Wrong calculation of pulse generator frequency when number of pulses is too small.

Description

When the number of pulses per increment DPLL_CTRL_0.MLT is smaller than 127, or DPLL_MLS1/2.MLS1/2 is smaller than 128 and the correction of physical deviations is used (DPLL_CTRL_0.AMT/AMS=1 and DPLL_CTRL_11.ADT/ADS=1), the calculation of internal values such as DPLL_DT_T/S_ACT.DT_T/S_ACT, DPLL_RDT_T/S_ACT.RDT_T/S_ACT, and DPLL_ADD_IN_CAL1/2.ADD_IN_CAL_1/2 is wrong. The resulting frequency of the generated sub increment pulses of the DPLL is too small.

Workaround

First Workaround: Don`t use pulse numbers DPLL_CTRL_0.MLT < 127 and/or DPLL_MLS1/2.MLS1/2 < 128, when using correction of physical deviation (DPLL_CTRL_11.ADT/ADS=1 when DPLL_CTRL_0.AMT/AMS=1).

Second Workaround: When first configuration cannot be applied then use configuration DPLL_CTRL_11.ADT/ADS=0 when DPLL_CTRL_0.AMT/AMS=1 is used.

ERR051635: GTM: (GTM_AEI) Write transaction in Split Mode to BRIDGE_MODE register may never terminate

Description

AEI Bridge is in async_bridge mode (BRIDGE_MODE.BRG_MODE='1'). Issuing an AEI Bridge software reset while switching to Mask write response mode through a split mode access (writing BRIDGE_MODE=h#10003) can under certain circumstances prevent the aei_response_ready-Signal from being set.

The split transaction is therefore not properly terminated and the protocol is corrupted.

Workaround

Issue the Reset in a separate transaction: Write BRIDGE_MODE=h#10001 then BRIDGE_MODE=h#3

ERR051636: GTM: (DTM) HRES output calculation for short input pulses and enabled dead time incorrect

Description

DTM channel is configured for dead time calculation. High resolution PWM support is enabled.

Configuration:

CDTM[i]_DTM[d]_CH_CTRL2.DT[0/1]_[x]=1

CDTM[i]_DTM[d]_CH[x]_DTV.HRES=1

Expected behaviour:

The HRES output value must be calculated correctly after three cluster clock cycles, even if the subsequent input edge on DTM_IN or DTM_IN_T or DTM_IN_PREV occurs after less than three cluster clocks.

Observed behaviour:

The calculated value of the HRES output signal for the current input signal edge is incorrect if the subsequent input signal edge on DTM_IN or DTM_IN_T or DTM_IN_PREV occurs after less than 3 cluster clock cycles.

Workaround

Avoid pulses from connected module TOM, ATOM or TIO which are shorter than 3 cluster clock cycles together with enabled dead time calculation.

No workaround available for shorter input pulses on DTM_IN or DTM_IN_T or DTM_IN_PREV.

ERR051637: GTM: (MCS) Invalid instruction trace output

Description

The MCS instruction trace output does not signalize a WURM, WURMX or WURCX instruction, if the corresponding match event for that instruction occurs one cluster clock cycle after the MCS channel enters its suspended state.

Workaround

No workaround

ERR051638: GTM: (MCS) Unexpected repeated break point action

Description

After the MCS is getting resumed from an instruction break point that is configured at a memory location containing a suspending instruction (e.g. WURM), the MCS is executing the instruction at the break point correctly but without incrementing the Program Counter. Afterwards the instruction at the activated break point is executed again.

Workaround

Disable the actual break point at MCS channel x by clearing the appropriate MCS[i]_HBP[k]_CTRL.EN_CH[x] first, and then resume the MCS channel x by a write access to MCS[i]_HBP[k]_STATUS.HALT_CH[x].

ERR051648: SPI: In Continuous Selection Format observed tASC timing differs from the expected one.

Description

In Continuous Selection Format (CONT) mode, the expected timing for After SCK Delay (tASC) is by the description of the Clock and Transfer Attributes Register in the ASC field (SPI.CTARn.[ASC]). The actual observed tASC time differs from expected time as follows, depending on Clock Phase (SPI.CTARn.CPHA) settings:

For CPHA=1:

- If expected tASC < 8 cycles, then observed tASC = 10 cycles of protocol clock .

- If expected tASC >= 8 cycles, then observed tASC = expected tASC + 2 cycles of protocol clock.

For CPHA=0:

- If expected tASC <8 cycles, then observed tASC = 9 cycles of protocol clock.

- If expected tASC >= 8 cycles, then observed tASC = expected tASC + 2 cycles of protocol clock

SPI module register description is also applicable to DSPI module.

Workaround

To properly match the requirements of the target SPI device, adjust configuration of SPI.CTARn fields PASC and ASC as needed to account for the difference in observed timing as given in errata description.

ERR051649: NETC: Switch ports support only one ‘802.1p and DEI to internal QoS’ mapping profile

Description

One of the methods to set the internal QoS of an incoming frame is to use the '802.1p and DEI to internal QoS' mapping function. The internal QoS represents the local QoS values of frame within the switch, namely the internal priority value (IPV) and drop resilience (DR) of the frame.

The '802.1p and DEI to internal QoS' mapping method can be selected on per switch port, along with specifying the '802.1p and DEI to internal QoS' mapping profile to be used. Two '802.1p and DEI to internal QoS' mapping profiles global to the switch that can be configured. Registers VLANIPVMPaR0/1 and VLANDRMPaR are used to configured the profiles where 'a' identifies the mapping profile instance, and can take the value of 0 or 1.

Only mapping profile instance 0 can be used. Mapping profile instance 1 cannot be used due the decoding of the selected profile is not correct. Therefore, if any switch port is configured to use the second mapping profile (1), all switch ports will not assign the default QoS correctly.

Note, the first profile (0) works correctly. The errata is only when at least one of the switch ports is not using profile 0.

Workaround

All switch ports must use the first VLAN QoS Mapping Profile (PQOSMR[VQMP]=0).

If multiple profiles is critical, then port classification lookups (Ingress Port Filter or Ingress Stream) can be used to provide alternative methods to set internal QoS values

ERR051651: NETC: Use of 10Mbps restricted to PHYs that support octet alignment for preamble with minimum 1B

Description

802.3 frames are octet based, but preambles are not necessarily in multiples of octets, depending on interface data width. When operating at 10Mbps, MII or RMII PHYs may remove bits of the preamble in multiples of 4b, leaving the packet non-octet aligned.

This device only supports octet-aligned packets. When running 10Mbps MII or RMII on a PHY that does not support octet-only alignment, the device may discard or flag an error on packets with a non-octet aligned preamble, as it would be unable to recognize the SFD/SMD. For some PHYs, that may impact the majority of packets.

Some 10 Mbps PHYs may also strip the entire preamble. This device does not support that mode of operation, but requires minimum 1B of preamble.

Workaround

1. Use 100Mbps operation instead of 10 Mbps, or

2. Use a PHY that supports octet-only alignment for 10 Mbps and does not strip all preamble bytes

ERR051663: DDRC: Memory Select Error Can Cause DDRC Failure

Description

A memory select error can be detected by the DDRC if a memory transaction is received by the DDRC that is outside the programmed CSn_BNDS register(s). This is considered a programming error.

Under most conditions, the memory controller will detect and report the memory select error, and it will continue operating.

However, if using page mode, all-bank refreshes, and dynamic refresh rate, then a memory select error can cause the DDRC to fail. In this scenario, a reset of the part would be required to recover after this time.

The memory select error will still be reported as expected.

Workaround

It is not expected to have memory select errors in production software. However, this scenario can still be avoided if any of the following conditions are true:

- Auto precharge mode is used (i.e., DDR_SDRAM_INTERVAL[BSTOPRE] is 0)

- Per bank refreshes are used via TIMING_CFG_9

- DDR_SDRAM_CFG_3[DYN_REF_RATE_EN] is 0

If none of the above conditions are used to avoid the potential failure, then the DDRC can be tested after a memory select error is detected. If it is non-functional, then a reset would be required.

ERR051666: GTM: Unexpected write access before instruction break point

Description

If an MCS program sequence has an instruction break point activated there are two scenarios at which unexpected write accesses are executed:

1) A subsequent instruction is executing a parallel memory write exactly 2 cluster clock cycles after the instruction at the break point.

2) A subsequent instruction is executing an AEI bus master write access exactly 1 cluster clock cycle after the instruction at the break point.

Note: Errata does not apply when the Round Robin scheduling mode is active.

Workaround

Add one NOP in scenario 2 and respectively two NOP instructions in scenario 1 between the instruction at the break point and the subsequent write access instructions.

ERR051667: GTM: Potential invalid bit field STA.Z for parallel memory read access

Description

If a memory read instruction that is executing a parallel read access is followed by an instruction that is reading the STA register for the evaluation of the bit field STA.Z and the delay between both instructions is exactly one clock cycle, the MCS architecture must initiate a pipeline flush, since the result from the memory read access is not yet available for the correct determination of the flag STA.Z. However the pipeline flush is not initiated.

Note: This errata does not apply when the Round Robin scheduling mode is active.

Workaround

Do either of the following to work around this issue:

Workaround 1: Re-schedule instruction sequence to avoid critical data dependency

Workaround 2: Add one NOP instruction between the parallel memory read access instruction and the instruction that is reading the register STA.

ERR051668: GTM: Potential wrong data for register MHB during parallel memory read access

Description

A memory read instruction is executing a parallel read access and the 8 LSBs of the read data are stored in the register MHB. The MCS reflects wrong data for the register MHB if one of the following scenarios about data dependencies to the subsequent instruction occurs:

A) The subsequent instruction is reading the register MHB in any argument and the delay between both instructions is exactly two cluster clock cycles.

B) The subsequent instruction is a parallel memory write instruction that is writing register MHB to the memory and the delay between both instructions is exactly three cluster clock cycles.

C) The subsequent instruction is a parallel memory read or write instruction that is reading register MHB in any argument and the delay between both instructions is exactly three cluster clock cycles.

Note: This errata does not apply when the Round Robin scheduling mode is active.

Workaround

Workaround :

1) Re-Schedule instruction sequence to avoid critical data dependency

2) Add two NOP in scenario A or three NOP instructions in scenario B and C between the parallel memory read access instruction at the subsequent instruction that is reading the register MHB.

ERR051669: GTM: AEI bridge might not execute an accepted transaction after a software reset.

Description

When the AEI Bridge operates in sync_bridge mode (BRIDGE_MODE.BRG_MODE='0') or bypasses synchronizer flip-flops (BRIDGE_MODE.BYPASS_SYNC='1') while a soft-reset is issued by writing BRIDGE_MODE.BRG_RST = '1', an upcoming transaction might be accepted by the bridge but never actually executed.

Workaround

Switch to async_mode and turn off bypassing synchronizer flip-flops before issuing a soft reset.

OR

Issue always a read access directly after issuing a software reset.

ERR051673: GTM:Potentially invalid MCS data trace output for parallel memory access.

Description

If an MCS instruction executes a parallel memory access and more than one MCS channel is active, the data trace interface potentially reports invalid data concerning the following information about the memory access instruction:

- channel number

- memory address

- transferred data

- data direction (read or write).

Workaround

No workaround is available.

ERR051674: GTM: Potential wrong data in memory during parallel memory write access while storing the register STA to memory

Description

Assume that an MCS instruction executes a parallel memory access in order to write the register STA to the memory. If its preceding instruction is updating at least one of the following status flags: STA.CY, STA.Z, STA.N, STA.V, or STA.CWT and the delay between both instructions is exactly one cluster clock cycle potentially wrong data of the associated status flags are written to the memory. The content of the register STA itself remains correct.

Note: This errata does not apply when the Round Robin scheduling mode is active.

Workaround

1) Re-schedule instruction sequence to avoid critical data dependency

OR

2) Add one NOP instruction between the concerned memory write instruction and its predecessor instruction.

ERR051675: GTM: Soft reset might not be triggered when issued through Pipeline-Mode

Description

A pipeline write access setting BRIDGE_MODE.BRG_RST to '1' might not trigger a soft-reset if the preceding transaction has already been in pipeline mode.

Workaround

1) Issue the soft reset with a write command in any mode other than pipeline mode.

OR

2) Issue a read transaction in standard mode before writing BRIDGE_MODE.BRG_RST in pipeline mode.

ERR051676: GTM: Unexpected stack pointer decrement

Description

If an RET or an POP instruction is executing a parallel data read access and an ECC error is signaled during the read access, the associated MCS channel signals the error correctly and it correctly stops the channel execution. However, the corresponding stack pointer register R7 is decremented, which is not expected during error handling.

Workaround

No workaround is available.

ERR051677: GTM: Unified error behavior

Description

In order to provide a unified error behavior for the various types of MCS errors, the specification has been refined. In addition to the already existing error behavior (namely stopping the associated MCS channel [x], setting bit field STA.ERR and MCS[i]_ERR.ERR[x], and updating bit field MCS[i]_CTRL_STAT.ERR_SRC_ID) no other MCS registers or memory cells are updated.

Workaround

No workaround is available. Application should be aware of the error behavior in the description of this erratum.

ERR051678: GTM: Missing edge on output signal ATOM/TOM_OUT when CN0 is reset with force update event

Description

The channel is configured in continuous up-counter mode. Then a new period is started with a force update event and reset of CN0 is activated.

Expected behavior:

After the counter ATOM/TOM[i]_CH[x]_CN0.CN0 has been reset and therefore a new period has to be started and the output signal (A)TOM_OUT has to be set immediately to SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL) and after the counter reaches ATOM/TOM[i]_CH[x]_CM1.CM1. an edge on ATOM/TOM_OUT to inverted SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL) is expected.

Observed behavior:

An edge on the output signal ATOM/TOM_OUT to SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL) at the beginning of the new period does not happen. Instead, the output signal ATOM/TOM_OUT holds its last value.

A second observation is in case of the SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL) changes synchronously together with the force update event, an edge on ATOM/TOM_OUT to the inverted SL value (ATOM[i]_CH[x]_CTRL_SOMP.SL, TOM[i]_CH[x]_CTRL.SL) when ATOM/TOM[i]_CH[x]_CN0.CN0 reaches ATOM/TOM[i]_CH[x]_CM1.CM1 does not happen.

Workaround

No workaround is available.

ERR051679: GTM: Flags of registers DPLL_STA_FLAG are not set

Description

The flags of register DPLL_STA_FLAG (STA_FLAG_T, STA_FLAG_S, INC_CNT1_FLAG, INC_CNT2_FLAG) are not set when one of these flags is cleared by a write operation of MCS register DSTAX within the same cluster clock cycle.

Workaround

If DPLL_CTRL_0.RMO = 0, only use DPLL_STA_FLAG.STA_FLAG_T or DPLL_STA_FLAG.INC_CNT1_FLAG.

If DPLL_CTRL_0.RMO = 1 and DPLL_CTRL_SMC = 0, only use DPLL_STA_FLAG.STA_FLAG_S or DPLL_STA_FLAG.INC_CNT1_FLAG.

If DPLL_CTRL_0.RMO = 1 and DPLL_CTRL_SMC = 1, only use one of DPLL_STA_FLAG.STA_FLAG_S, DPLL_STA_FLAG.INC_CNT2_FLAG, DPLL_STA_FLAG.STA_FLAG_T, or DPLL_STA_FLAG.INC_CNT1_FLAG.

ERR051680: GTM: The AEI bridge might not execute an accepted write transaction

Description

If the AEI Bridge operates in pipeline mode while a soft-reset is issued (writing BRIDGE_MODE.BRG_RST = '1'), upcoming write transactions primed in the buffer although accepted may never be actually executed.

Workaround

Ensure the first access across the AEI bridge after a soft reset is a read access rather than a write access. Any read access is effective.

ERR051681: GTM: Change of the BRIDGE_MODE register might be delayed indefinitely

Description

The bit fields BRG_MODE and BYPASS_SYNC will not be updated after making a write access to the BRIDGE_MODE register until the transaction buffer is empty. In split mode, the bridge allows new transactions to be added to the buffer, even when an update of these bits is pending.

Polling the register in split mode might prevent the buffer from getting empty and hence prevents the actual update of the described bit fields.

Workaround

Avoid constant polling of this register. If polling is necessary add sufficient wait time between reads of the register to allow the transaction buffer to empty.

ERR051682: GTM: Internal compare events are not cleared

Description

If a dual compare command is disabled by writing TIO[i]_ENDIS.CH[x:x]=0 and enabled again by writing TIO[i]_ENDIS.CH[x:x]=1, compare events which have been detected during the first activation are not cleared.

Workaround

It is possible to do a software reset at both affected channels and to setup a new compare instruction as before.

ERR051698: CTU : Double buffer reload mechanism is blocked when master reload pulse is not generated by Software

Description

CTU operates on two clock domains. The Bus Interface Clock (BIC) domain, used for SW communication and the Module Clock (MC) domain, used for its own functionality. The FGRE bit of CR register is set with first write into double buffered registers in BIC domain and the synchronization logic propagates this set bit into the MC domain. FGRE bit is cleared in both domains when the MR occurs and FGRE bit is equal to GRE bit in the MC domain. Double buffered registers are reloaded only when MR occurs and FGRE and GRE bit are set in the MC domain. But when the MR occurs at the same time as FGRE bit set, generated by first double buffered register write, is propagated into the MC domain the FGRE bits are cleared in both.

Two possible cases can happen:

1) In case of updating one double buffered register the reload doesn’t occur and the CR register is kept in 0x2 value which blocks further double buffered register update.

2) In case of updating more than one double buffered register the reload doesn’t occur and the CR register is kept in 0xA when the delay between first and second double buffered register write is less than the synchronization propagation time which blocks further register update.

Workaround

1) Generate the Master reload pulse by SW (by setting the bit CR[MRS_SG] after setting GRE at the end of update sequence)

2) Master reload pulse is not generated by SW and one double buffered register to be updated:

The register needs to be written twice with the delay between the writes longer than synchronization propagation time.

3) Master reload pulse is not generated by SW and more than one double buffered register to be updated:

The delay between write to first and second double buffered register needs to be longer than synchronization propagation time.

Synchronization propagation time = (6*BIC) cycles + (5*MC) cycles, where BIC is Bus Interface Clock period and MC is Module Clock period.

ERR051701: SPI: De-asserting of CONT bit in the Continuous Selection Format is not mandatory for the last frame

Description

In Continuous Selection Format it is requested the Continuous Peripheral Chip Select Enable bit (SPI.PUSHR[CONT]) is de-asserted for the last frame to be transmitted. However when SPI.PUSHR[CONT] is not de-asserted for the last frame, data transfer happens correctly according to corresponding SPI.CTARE[DTCP] configuration.

SPI module register description is also applicable to DSPI module.

Workaround

De-assertion of the SPI.PUSHR[CONT] bit in the last frame to be transmitted is not mandatory. Data transfer works correctly as programmed in SPI.CTARE[DTCP] regardless the SPI.PUSHR[CONT] configuration in the last frame to be transmitted.

ERR051734: Core: DWT comparator match on cycle count is not reported to the ETM if there is no instruction executing on the processor

Description

Cortex-M33 2435965-C

The Cortex-M33 Data Watchpoint and Trace (DWT) unit supports a Cycle count match event which can be used to trigger the Embedded Trace Macrocell (ETM) to generate a trace packet from the processor. Due to this erratum the event signal is only propagated when an instruction is executing in the pipeline and so no event will be transferred to the ETM if the processor is idle.

Workaround

There is no workaround for this erratum, however, non-debug operation of the core is not affected.

ERR051777: DMA_CRC: TCDs with a non-zero SMOD attribute lead to invalid checksum calculations if CRC is enabled on the channel

Description

An invalid checksum value will be calculated whenever Cyclic Redundancy Check (CRC) is enabled on an eDMA channel if the Source Address Modulo (SMOD) attribute of the Transfer Control Descriptor (TCD) is any number greater than 0.

Conditions:

- Global Enable bit of the DMA_CRC Global Enable CRC Register (GEC[GBL_EN]) is set to 1
- Enable bit of the DMA channel CRC Control Register (CTLn[EN]) is set to 1
- Source Address Modulo field of the eDMA TCD Transfer Attributes Register (TCDn_ATTR[SMOD]) is a non-zero number

If all conditions are met, an incorrect CRC checksum will be calculated for the TCD.

Implications:

If a circular queue buffer is implemented on a given eDMA channel, CRC cannot be used.

Workaround

Either one of the following workarounds can be used:

1. If CRC is enabled on an eDMA channel, disable circular queue buffer support on the channel by setting the Source Address Modulo field of the eDMA TCD Transfer Attributes Register (TCDn_ATTR[SMOD]) to zero (0). This workaround should be chosen for a safety relevant applications.

2. Disable CRC on the eDMA channel to enable the use of a circular queue buffer on that channel. This workaround can only be used for any non-safety relevant applications.

ERR051779: GTM: (TIO) Buffer empty signalization (PL_EVT) is not correct

Description

If a buffer is reloaded by its predecessor's buffer which has been written with TIO[i]_G[g]_CH[c]_SCMD.SCMD=0x0 or TIO[i]_G[g]_CH[c]_OCMD.OCMD=0x0, then the buffer filled flag is not set to 0 (0 = buffer empty).

Examples of affected reloading: Reloading a buffer e.g. at instruction end with its predecessors buffer TIO[i]_G[g]_CH[c]_SCMD.SCMD = TIO[i]_G[g]_CH[c-1]_OCMD.OCMD with TIO[i]_G[g]_CH[c-1]_OCMD.OCMD = 0x0 or TIO[i]_G[g]_CH[c]_OCMD.OCMD = TIO[i]_G[g]_CH[c]_SCMD.SCMD with TIO[i]_G[g]_CH[c]_SCMD.SCMD = 0x0.

Workaround

Do not use instruction sequences with command value 0x0 for TIO[i]_G[g]_CH[c]_SCMD.SCMD or TIO[i]_G[g]_CH[c]_OCMD.OCMD

ERR051780: GTM: Wrong AEI status on accesses to ICM registers while cluster 0 clock is switched off

Description

Bus accesses to ICM register while cluster 0 clock is switched off by setting of GTM_CLS_CLK_CFG.CLS0_CLK_DIV=0b00 should return AEI status value of 0b10 but it returns always AEI status value of 0b00.

Workaround

1) Do not access ICM while cluster 0 clock is disabled.

OR

2) Avoid switching off Cluster 0.

ERR051781: GTM: (TIM) Potentially wrong capture values

Description

Configuration:

The TIM channel is configured in TIEM, TIPM, TGPS or TSSM mode by setting of TIM[i]_CH[x]_CTRL.TIM_MODE=0b010, 0b011, 0b101, 0b110. The TIM channel is disabled (TIM[i]_CH[x]_CTRL.TIM_EN=0) and later enabled again (TIM[i]_CH[x]_CTRL.TIM_EN=1).

Expected behaviour for TIEM/TIPM/TGPS mode:

The registers TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_ECNT.ECNT[15:1], TIM[i]_CH[x]_GPR0 and TIM[i]_CH[x]_GPR1 are set to their reset values. In case of an input signal edge or an input capture event or an active selected CMU clock (TGPS mode) at the same time as the channel is enabled, this event has to be taken into account and the TIM[i]_CH[x]_CNT register must be updated/incremented based on its reset value. Due to this a capture event can happen depending on the configured TIM mode and the register values.

Expected behaviour for TSSM mode:

The registers TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_ECNT.ECNT[15:1], TIM[i]_CH[x]_GPR0 and TIM[i]_CH[x]_GPR1 are set to their initial values. The initial value for TIM[i]_CH[x]_CNT register depends on TIM[i]_CH[x]_CTRL.ISL and TIM[i]_CH[x]_CNTS.CNTS(22). If TIM[i]_CH[x]_CNTS.CNTS(22) is set to 0 and TIM[i]_CH[x]_CTRL.ISL is set to 0 the initial value of TIM[i]_CH[x]_CNT is 0x000000. An input signal event simultaneously to the channel enable is not taken into account.

Observed behaviour for TIEM/TIPM/TGPS mode:

If no input signal event or input capture event or active selected CMU clock (TGPS mode) occurs, the registers TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_ECNT.ECNT[15:1], TIM[i]_CH[x]_GPR0 and TIM[i]_CH[x]_GPR1 are set to their reset values as expected.

If an input signal event or an input capture event or an active selected CMU clock (TGPS mode) occurs at same time as the channel gets enabled, the TIM[i]_CH[x]_CNT register continues to count (or update) based on the previous (old) value. As a result, a capture could be performed too early and/or with the wrong values.

The TIM[i]_CH[x]_ECNT.ECNT[15:1] register is set to its reset value as expected.

Observed behaviour for TSSM mode:

The registers TIM[i]_CH[x]_CNT is not set to its initial value of 0x000000 on channel enabling when TIM[i]_CH[x]_CNTS.CNTS(22) is set to 0 and TIM[i]_CH[x]_CTRL.ISL is set to 0.

Note: The TIM channel modes TPWM, TPIM and TBCM (TIM[i]_CH[x]_CTRL.TIM_MODE=0b000, 0b001, 0b100) are not affected.

Workaround

Workaround 1:

Reset the TIM channel by setting of TIM[i]_RST.RST_CH[x]=1 before enabling the TIM channel.

Workaround 2:

The following sequence has to be executed on the disabled channel but before the actual enabling of the channel, to ensure that the TIM[i]_CH[x]_CNT register is set to its reset value when the channel is enabled:

1. Configure TIM[i]_CH[x]_CNTS=0

2. Enable the TIM channel with the following configuration inside the TIM[i]_CH[x]_CTRL register:

- TIM_EN=1

- TIM_MODE=0b101 (TGPS)

- ISL=1

- OSM=1

- ARU_EN=0

- select a fast CMU_CLK_RES, e.g. CLK_SEL=0b000

3. Wait until an edge on the selected CMU_CLK_RES occurs. This can be observed on the NEWVAL IRQ notify register. This event sets the TIM[i]_CH[x]_CNT register to its reset value.

4. Disable TIM channel (TIM[i]_CH[x]_CTRL.TIM_EN=0)

5. Configure the former TIM channel configuration in TIM[i]_CH[x]_CTRL register and enable the TIM channel again.

ERR051782: GTM: (AEI) Changing BRIDGE_MODE.MSK_WR_RSP in Pipeline mode can lead to violation of Pipeline protocol

Description

In Pipeline mode, a reconfiguration of the BRIDGE_MODE.MSK_WR_RSP directly after another write transaction can lead to a hang of following write transactions by not setting the AEI_READY.

Workaround

1) Make sure the transaction preceding the write of BRIDGE_MODE.MSK_WR_RSP is a read transaction.

OR

2) Integration dependent: Issue the write to BRIDGE_MODE.MSK_WR_RSP in standard mode instead of pipeline mode.

ERR051783: GTM: (DPLL) Wrong DPLL_RDT_S_ACT/DPLL_RDT_T_ACT value in case of overflow correction

Description

The wrong overflow correction occurs for DPLL_RDT_S_ACT when the DPLL is in normal mode (DPLL_CTRL_0.RMO=0, DPLL_CTRL_1.SMC=0) or for

DPLL_RDT_T_ACT when the DPLL is in emergency mode (DPLL_CTRL_0.RMO=1, DPLL_CTRL_1.SMC=0).

Instead of 0xFFFFFF the value 0x000000 is written in both cases. A problem in calculation of pulse frequency (settling behaviour) or for PMT values may occur, when the mode DPLL_CTRL_0.RMO is switched to the other mode (normal mode <-> emergency mode). If the overflow value was not yet overwritten (due to engine revolution happening before mode's switch) the wrong value might come into use for the described calculations

Workaround

Modification of DPLL_RDT_T_ACT (emergency mode) or DPLL_RDT_S_ACT (normal mode) after detection of overflow condition is not possible but does not cause any negative effects on pulse generation or PMT calculation at all.

The values stored to DPLL_RDT_T of RAM2 or DPLL_RDT_S of RAM1bc need to be corrected by following workaround sequence:

1) Check if relevant overflow on either DPLL_RDT_T_ACT or DPLL_RDT_S_ACT occurred.

This can be done by observation of DPLL_STATUS.CRO when interrupt DPLL_IRQ_NOTIFY.EI occurred.

2) Check which of the interrupts DPLL_IRQ_NOTIFY.TASI/SASI has occurred next and based on that DPLL_RDT_T or DPLL_RDT_S has to be corrected.

3) For DPLL_CTRL_0.RMO=0 and DPLL_CTRL_1.SMC=0, DPLL_RDT_S[DPLL_APS.APS -1] has to be written to 0xFFFFFF.

For DPLL_CTRL_0.RMO=1 and DPLL_CTRL_1.SMC=0, DPLL_RDT_T[DPLL_APT.APT -1] has to be written to 0xFFFFFF.

ERR051784: GTM: (MCS) Unexpected Memory overflow when using RET instruction

Description

If a RET instruction is executed at memory location MP1-4 (last valid memory location) the associated MCS channel stops and signals a memory overflow instead of executing the instruction.

Workaround

Reschedule MCS program in a way that memory location MP1-4 has no RET instruction.

ERR051785: GTM: (MCS) Missing Memory overflow detection on literal addresses

Description

For a memory access at memory location x using a direct memory access (instructions MRD, MWR, or MWRL) or an indirect memory access with literal offset (instruction MRDI or MWRI), an address overflow is not detected for the range 2^(RAW + USR) <= x < 2^14.

Workaround

Software needs to avoid accesses to invalid addresses.

Otherwise no workaround in hardware possible.

ERR051786: GTM: (MCS) Instructions BRDI and BWRI evaluate unused address bits

Description

Bus master instructions with indirect addressing (BRDI and BWRI) use the bits 2 to 15 of register B for defining its target address. However, if the bit slice B[1:0] is unequal to 0 the current implementation of the GTM-IP behaves as follows:

bit field CCM[i]_AEIM_STA.AEIM_XPT_STA is updated with value 1 and CCM[i]_AEIM_STA.AEIM_XPT_ADDR is updated with the address of the selected register B. Further, if bit field MCS[i]_CTRL_STAT.HLT_AEIM_ERR is set then the associated MCS channel stops and indicates an unexpected bus master error.

Workaround

The use of the lower address bits are forbidden for indirect addressing. Hence this error needs to be avoided by the application.

ERR051787: GTM: (MCS) Unexpected Break Point initiation

Description

The current implementation of an MCS hardware break point h is as follows: Potential break point match events

that are defined by the bit fields MCS[i]_HBP[h]_CTRL.TYPE, MCS[i]_HBP[h]_CTRL.DATA, MCS[i]_HBP[h]_CTRL.AND, and MCS[i]_HBP[h]_CTRL.NOT

and the input port DBG_MCS[i]_BP_EN are stored in an internal pipeline register for all enabled MCS channels but without considering the current values of the bitfields MCS[i]_HBP[h]_CTRL.EN_CH[x]. However, such a potential break point match event of an MCS channel x is actually fired if it is reaching the end of the pipeline register and the corresponding bit field MCS[i]_HBP[h]_CTRL.EN_CH[x] is currently set to 1.

This means, the functional behavior of the h-th hardware break point is only correct, if this hardware break point is not re-configured in a way that other running MCS channels y are changing their break point enable behavior from MCS[i]_HBP[h]_CTRL.EN_CH[y]=0 to MCS[i]_HBP[h]_CTRL.EN_CH[y]=1.

Workaround

No workaround in hardware possible. Debugger applications must try to avoid such critical reconfigurations.

ERR051788: GTM: (MCS) Missing resume from Hardware Break Point

Description

The hardware break point functionality is defined that the i-th MCS instance shall resume from any pending hardware break point if GTM-IP input port DBG_MCS[i]_BP_EN is set to 0.

However, this feature is not functional.

Workaround

Resuming from a pending break point can be done by a write access to clear the bit fields MCS[i]_HBP[h]_STATUS.HALT_CH[x].

ERR051789: GTM: (DPLL) Pulse correction is executed twice

Description

If DPLL_CTRL_1.PCM1/2 is set during DPLL_CTRL_1.DEN=0 or DPLL_CTRL_1.DEN changes from 1 to 0, these values are immediately transferred to the respective shadow registers DPLL_CTRL_1_TRIGGER_SHADOW and DPLL_CTRL_1_STATE_SHADOW. Since the DPLL_CTRL_1.DEN=0 the DPLL_CTRL_1.PCM1/2 registers are not cleared when transferred to the shadow registers.

When DPLL_CTRL_1.DEN=1 and the next relevant input signal on either STATE or TRIGGER arrives the pulse corrections are executed due to the state of the named shadow registers.

As the DPLL_CTRL_1.PCM1/2 were not cleared this is leading to an additional pulse correction for the next following input signal (TRIGGER / STATE).

Workaround

Avoid setting DPLL_CTRL_1.PCM1/2 when DPLL_CTRL_1.DEN=0.

ERR051790: GTM: (TIM) Missing glitch detection interrupt event

Description

Configuration:

TIM filter is configured in immediate edge propagation mode by setting TIM[i]_CH[x]_CTRL.FLT_MODE_RE = 0 or TIM[i]_CH[x]_CTRL.FLT_MODE_FE = 0. The filter is enabled by setting TIM[i]_CH[x]_CTRL.FLT_EN = 1.

Expected behaviour:

As long as the filter threshold is not reached and the input signal level unexpectedly changes, it is an input glitch occurs, the internal glitch detection interrupt event signal (TIM_GLITCHDET_IRQ) should have a HIGH pulse of one cluster clock cycle.

Observed behaviour:

When the input signal glitch occurs at the same time the filter counter reaches its threshold, the internal glitch detection interrupt event signal (TIM_GLITCHDET_IRQ) does not occur.

Workaround

The filter counter threshold can be set to the next higher value. Thus, a former not detected glitch would be detected. In that case, the output signal would be changed (one clock cycle longer), when the input signal is a single cycle pulse.

ERR051791: GTM: (TIM) Unexpected increment of filter counter

Description

Configuration:

TIM filter is configured in immediate edge propagation mode by setting TIM[i]_CH[x]_CTRL.FLT_MODE_RE = 0 and/or TIM[i]_CH[x]_CTRL.FLT_MODE_FE = 0. The filter is enabled by setting TIM[i]_CH[x]_CTRL.FLT_EN = 1. The filter counter threshold is set to zero by setting TIM[i]_CH[x]_FLT_RE.FTL_RE = 0 and/or TIM[i]_CH[x]_FLT_FE.FTL_FE = 0.

Expected behaviour:

When the input signal level changes, the filter counter should not increment.

Observed behaviour:

When the input signal level changes, the filter counter increments by one and is not reset.

Workaround

If acceptable, use a threshold greater than zero. Otherwise there is no workaround available.

However, there is a possibility of minimizing the absolute error, deriving from this bug. If possible, a faster CMU clock can be selected. This leads to a shorter absolute time difference between the expected and actual output signal change. Additionally when applying this, the filter counter thresholds need to be assimilated proportionally in order to make the filter work as before.

ERR051792: GTM: (TIM) Glitch detection interrupt event of filter is not a single cycle pulse

Description

The TIM filter must be enabled by setting TIM[i]_CH[x]_CTRL.FLT_EN = 1.

Expected behaviour:

As long as the filter threshold is not reached and the input signal level changes unexpectedly, the glitch detection interrupt event signal (GLITCHDET_IRQ) should have a single cycle HIGH pulse.

Observed behaviour:

When the input signal level changes unexpectedly for longer than one clock cycle, the glitch detection interrupt event signal (GLITCHDET_IRQ) is HIGH for as many cluster clock cycles as the unexpected signal change is present.

Workaround

No workaround in hardware.

For the unexpected retrigger of the interrupt directly after an interrupt clear step, the interrupt routine has to consider that the interrupt might be invalid.

ERR051793: GTM: (TIO) Compare mode: Output value not correct if two events occur at the same time

Description

Action on O_OUT[c:c] (driving TIO_G[g]_OUT[c:c]) can be missing if in compare mode the current channel is configured as cyclic buffer and the trigger events of PL_TRIG_OUT[c:c] and O_TRIG_OUT[c:c] occur in the same cluster clock cycle.

The action defined by O_TRIG_OUT[c:c] should be applied here.

Workaround

No workaround in hardware possible. The application should avoid this scenario by disabling one of the two triggers.

ERR051794: GTM: (TIO) Incorrect behaviour on O_OUT in case of consecutive compare events

Description

Output O_OUT[c:c] (driving TIO_G[g]_OUT[c:c]) can be incorrect if the two compare events of a dual compare instruction occur back-to-back. That is, if the second compare event occurs in the next cluster clock after the first compare event.

The error can only occur if the first compare event actually changes the O_OUT[c:c] value.

Workaround

Workaround1:

Ensure that the resolution of master and slave channel is identical (PL_UPDATE[c:c] =PL_UPDATE[c-1:c-1]).

Workaround2:

Ensure that the 2 resolutions PL_UPDATE[c:c] and PL_UPDATE[c-1:c-1] can never occur back-to-back.

ERR051796: GTM: (DPLL) Doubled pulse correction

Description

In automatic end mode (DPLL_CTRL_1.DMO=0) using DPLL_CTRL_1.PCM1/2=1 and DPLL_CTRL_11.INCF1/2=1, when DPLL_CTRL_1.DEN=0->1 changes and the first input signal on either TRIGGER/STATE arrives the pulse correction is executed two times with DPLL_MPVAL1/DPLL_MPVAL2.

Workaround

Workaround 1: Postpone the pulse correction under the described conditions to the second increment.

Workaround 2: For even values of pulse corrections half of the original DPLL_MPVAL1/DPLL_MPVAL2 can be used. With that the effective number of corrected pulses will match the original DPLL_MPVAL1/DPLL_MPVAL2.

ERR051797: GTM: (TIO) In dual compare mode the initialization of internal compare event enables is not correct

Description

In the dual compare mode the initialization of internal compare event enable flags at state transition DISABLED to ENABLED is not performed, because the inner logic depends on TIO[i]_G[g]_CH[c]_OCMD_COMP.CMD_ACTIVE_CC=b01 which is not intended.

Workaround

No workaround in hardware possible.

The application should avoid state transitions from DISABLED to ACTIVATED via ENABLED, but instead implement state transfers from DISABLED directly to ACTIVATED.

ERR051798: GTM: (TIO) In dual compare mode the update of internal compare event enables is incorrect

Description

In dual compare mode the update of internal compare event enables is done even if instruction is not activated (TIO[i]_G[g]_CH[c]_OCMD_COMP.CMD_ACTIVE_CC=b00).

Workaround

No workaround in hardware possible.

The application should avoid this corner case of

deactivating an instruction and activating it again (via writing TIO[i]_G[g]_CH[c]_OCMD_COMP.CMD_ACTIVE_CC).

ERR051799: GTM: (TIO) Wrong output value on O_OUT in capture mode

Description

In capture mode the output O_OUT[c:c] (driving TIO_G[g]_OUT[c:c]) of channel [x] might have a wrong value if a write access to TIO_O.CH[x] is done one cluster clock cycle before the instruction terminates.

Workaround

Do not write TIO[i]_O.CH[x] if the capture instruction in channel [x] is activated.

ERR051800: GTM: (MCS) Unexpected data break point behavior

Description

The hardware break point functionality has the following unexpected behavior if a break point is configured with MCS[i]_HBP[h]_CTRL.TYPE = DADR or MCS[i]_HBP[h]_CTRL.TYPE = DPAT and a serial data access occurs:

1) In the case of the execution of a prefetched instruction that is following any serial data access, the prefetched instruction might be misinterpreted as data resulting in an unexpected break point.

2) Serial data accesses resulting from stack operations of the instructions CALL, CALLI, or RET do not initiate any break point.

Workaround

No workaround in hardware possible.

ERR051801: GTM: (DPLL) Missing pulse correction in case of DPLL_CTRL_1.SMC=1

Description

If DPLL_CTRL_1.SMC=1 and DPLL_CTRL_0.RMO=1 and DPLL_CTRL_11.PCMF1=0 no pulse correction on CCM[0]_TBU_TS1 is executed.

Workaround

1) Use DPLL_CTRL_0.RMO=0 when DPLL_CTRL_1.SMC=1. In this case fast pulse corrections via DPLL_CTRL_11.PCMF1 are possible.

2) Use DPLL_CTRL_11.PCMF1=1 for pulse corrections. In this case no negative values for DPLL_MPVAL1 can be used

ERR051804: GTM: (MCS) Unexpected state of register MCS[i]_HBP[k]_STATUS

Description

Assume that the k-th MCS break point is configured with MCS[i]_HBP[k]_CTRL.SCOPE = 2 and this break point is enabled for an MCS channel x by setting MCS[i]_HBP[k]_CTRL.EN_CH[x] = 1. If the MCS channel x reaches that break point, the register MCS[i]_HBP[k]_STATUS signalizes the break point correctly by setting MCS[i]_HBP[k]_STATUS.HALT_CH[x] = 1.

Further, if we assume that the k-th break point is reconfigured again with MCS[i]_HBP[k]_CTRL.SCOPE = 2 and it is enabled for another channel y (with x unequal to y) by setting MCS[i]_HBP[k]_CTRL.EN_CH[y] = 1 the following unexpected behavior might happen:

The register MCS[i]_HBP[k]_STATUS still signalizes MCS[i]_HBP[k]_STATUS.HALT_CH[x] = 1 from the first break point although the second break point has already been reached. The second break point is nevertheless signalized correctly by setting bit field MCS[i]_HBP[k]_STATUS.HALT_CH[y] = 1.

Workaround

No workaround in hardware possible.

ERR051805: GTM: (TIO) Erroneous assertion of PL_EVT when channel is configured as a cyclic buffer

Description

Configuration: The channel is configured as a cyclic buffer with S resource configured as a count instruction or a buffer and the O resource configured as a buffer.

The PL_EVT[c:c] is unexpectedly asserted on every exchange that loads the instruction with 0x0 into the O resource.

Workaround

For the described configuration, do not use the PL_EVT[c:c] as an update source (UPDATE[c:c], PL_UPDATE[c:c]), as an interrupt source (TIO_IRQ[c:c]), or as a trigger source (TRIG_OUT[c:c], PL_TRIG_OUT[c:c]).

ERR051806: GTM: (DPLL) Stored time stamp values do not consider filter delays

Description

For the case where the filter delay values should be considered (DPLL_CTRL_0.IDT/IDS=1) the data values of the time stamp fields in RAM1c (DPLL_TSF_S) and RAM2 (DPLL_TSF_T) actually do not take them into account for the input signals TRIGGER/STATE.

Workaround

The entry of DPLL_TSF_T[p]/_S[p] can be read, corrected (by DPLL_FTV_T/_S), and written back.

The correction needs to be done after the DPLL has received new input data. For this reason it is necessary to read and store the filter value of the last but one DPLL input signal, which then will be used for the correction.

ERR051807: GTM: (TIO) Incorrect initialization of internal compare event enables on cancel trigger

Description

In the dual compare mode the initialization of internal compare event enable flags is not performed when a cancel trigger O_INSTR_PULL_NEXT occurs.

Workaround

No workaround in hardware possible.

The application should avoid canceling an instruction by using O_INSTR_PULL_NEXT.

ERR051808: GTM: (TIO) Cancel trigger in single compare mode does not suppress action on O_OUT

Description

If in single compare mode an instruction terminates regularly and the cancel trigger O_INSTR_PULL_NEXT occurs at the same time the instruction is canceled, instruction end (signal INSTR_END=1) is correctly suppressed, but not the action on O_OUT[c:c] (driving TIO_G[g]_OUT[c:c]).

Workaround

No workaround in hardware possible.

The application should avoid canceling an instruction by using O_INSTR_PULL_NEXT

ERR051809: GTM: (DPLL) No action calculation

Description

If DPLL_CTRL_1.SMC=1 and DPLL_CTRL_0.RMO=0 no action calculation is done in STATE processing unit for action channels NOAC/2 to NOAC-1 (NOAC: number of action channels).

Note: Starting with V4.1.* NOAC=32, while in previous versions NOAC may be set to either 32 or 24.

Workaround

None

ERR051810: GTM: (DPLL) Missing TOR interrupt and status flag

Description

If DPLL_CTRL_0.RMO=1 and DPLL_CTRL_1.SMC=0, the TOR interrupt (DPLL_IRQ_NOTIFY.TORI) is not triggered and the status flag (DPLL_STATUS.TOR) is not set on encountering an out of range TRIGGER.

Workaround

No workaround available in hardware.

Nevertheless the application can detect the trigger out of range interrupt by observing TBU_TS0:

If the current TRIGGER time stamp (DPLL_TS_T.TRIGGER_TS / DPLL_TS_T_OLD.TRIGGER_TS) + DPLL_DT_T.DT_T_ACT * DPLL_TLR.TLR > TBU_TS0 and no active TRIGGER input was encountered, the CPU/MCS can force a TOR interrupt by writing a one to DPLL_IRQ_FORCINT.TRG_TORI.

ERR051811: GTM: (MCS) Missing data break point for parallel memory access

Description

If a hardware break point with index 0 is configured with type DPAT-R or DPAT-RW a match event resulting from a parallel memory read access does not trigger a break point

Workaround

No workaround in hardware possible.

ERR051812: GTM: (DPLL) DPLL_PVT not cleared after direction change

Description

For settings of DPLL_CTRL_1.SMC=1 or alternatively DPLL_CTRL_1.SMC=0 and DPLL_CTRL_1.IDDS=1 the direction change on TRIGGER channel is done via DPLL input port "TDIR" (generated via the control path SPE or TIM to MAP to DPLL).

If there is a direction change the RAM parameter DPLL_PVT is not cleared as specified.

Workaround

Reset DPLL_PVT by CPU or MCS0 write operation, when direction change is detected via DPLL_IRQ_NOTIFY.DCGI interrupt.

ERR051813: GTM: ((A)TOM) Missing CCU0TC_IRQ interrupt signal for UDMODE>0

Description

Configuration:

The channel is configured in SOMP (ATOM) up-down counter mode with (A)TOM[i]_CH[x]_CTRL.UDMODE>0 and will be triggered by an preceding channel with configuration of (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1.

Expected behaviour:

When the counter (A)TOM[i]_CH[x]_CN0.CN0 reaches in the up-counting phase the value of (A)TOM[i]_CH[x]_CM0.CM0, the interrupt signal CCU0TC_IRQ must be triggered.

Observed behaviour:

In the first period after (A)TOM[i]_CH[x]_CM0.CM0 is changed to the value 0, the CCU0TC_IRQ interrupt signal is triggered but not in the following periods with unchanged value of (A)TOM[i]_CH[x]_CM0.CM0=0.

A second observation is that the CCU0TC_IRQ interrupt signal is not triggered in the first period after the value of (A)TOM[i]_CH[x]_CM0.CM0 is changed from 0 to 1. Note: in this case, the CCU0TC_IRQ interrupt is triggered in the following periods with unchanged value of 1 for (A)TOM[i]_CH[x]_CM0.CM0.

Workaround

No workaround available.

If applicable use the interrupt indication from the preceding channel, which is always generated half a period earlier.

ERR051814: GTM: (TOM) Unexpected behaviour of TOM_OUT_T for UDMODE>0

Description

Configuration:

The channel is configured in up-down counter mode with TOM[i]_CH[x]_CTRL.UDMODE>0 and will be triggered by an preceding channel with configuration of TOM[i]_CH[x]_CTRL.RST_CCU0=1.

Expected behaviour:

The output signal TOM_OUT_T has to be set to TOM[i]_CH[x]_CTRL.SL value as long as the condition TOM[i]_CH[x]_CN0.CN0 >= TOM[i]_CH[x]_CM0.CM0 is true.

Observed behaviour for TOM[i]_CH[x]_CM0.CM0=0:

The output signal TOM_OUT_T is set to TOM[i]_CH[x]_CTRL.SL value only for one clock period of the selected CMU clock when TOM[i]_CH[x]_CN0.CN0 has reached 0. Afterwards TOM_OUT_T is set unexpectedly to the inverted value of TOM[i]_CH[x]_CTRL.SL.

Observed behaviour for TOM[i]_CH[x]_CM0.CM0=1:

An unexpected pulse on the output signal TOM_OUT_T with the length of one clock period of the selected CMU clock to the inverted value of TOM[i]_CH[x]_CTRL.SL can be observed when the trigger input signal TRIGIN occurs and the counter TOM[i]_CH[x]_CN0.CN0 starts to count down.

Workaround

No workaround available.

ERR051815: GTM: (MCS) Some internal MCS registers are not controlled by GTM halt logic

Description

MCS processing is not reliable with respect to the mentioned effects after resuming from a GTM halt event.

Description :

Some internal MCS registers (not mentioned in the specification) are not controlled by the GTM halt logic. Therefore, the following side effects might occur:

1) If an MCS is getting resumed after a GTM halt event a previous error of type write protection (MCS[i]_CTRL_STAT.ERR_SRC_ID = 6), ECC error (MCS[i]_CTRL_STAT.ERR_SRC_ID = 1), or memory overflow (MCS[i]_CTRL_STAT.ERR_SRC_ID = 2) might not be signalized and the MCS will not halt due to that error.

2) If an MCS is getting resumed after a GTM halt event a halted parallel memory access might be corrupt.

Workaround

No workaround in hardware possible. A resume after GTM halt is not recommended.

ERR051816: GTM: (MCS) Write access to protected bit field reports unexpected AEI status

Description

If the protection enable condition of bit field MCS[i]_CTRL_STAT.RAM_RST is active any write access to the register MCS[i]_CTRL_STAT nevertheless must return an AEI status 0. However, if this protection enable condition is active and a write access with value 1 to bitfield MCS[i]_CTRL_STAT.RAM_RST is executed an AEI status 2 is returned.

Workaround

No workaround in hardware possible.

The application needs to consider the unexpected status

ERR051817: GTM: (MCS) Missing trace information of WUCE instruction

Description

If the match event for a WUCE instruction arrives a few system clock cycles after the corresponding MCS channel has entered the suspended state, the WUCE instruction is executed correctly, but the instruction trace interface does not signalize this instruction

Workaround

No workaround in hardware possible.

ERR051821: GTM: (DPLL) DPLL_PSTC, DPLL_PSSC erroneously modified.

Description

If a direction change happens while the TRIGGER processing unit is not yet synchronized (DPLL_STATUS.SYT = 0) then DPLL_PSTC is erroneously overwritten.

If a direction change happens while the STATE processing unit is not yet synchronized (DPLL_STATUS.SYS = 0) then DPLL_PSSC is erroneously overwritten.

Workaround

Store the DPLL_PSTC, DPLL_PSSC values outside the DPLL, each time a TRIGGER/STATE input occurs.

If a direction change is detected, overwrite the newly calculated value by the value stored earlier. This is necessary as long as the DPLL is not yet synchronized (DPLL_STATUS.SYT=0 for DPLL_PSTC and/or DPLL_STATUS.SYS=0 for DPLL_PSSC).

ERR051822: GTM: (DPLL) Incorrect values of DPLL_RCDT_TX, DPLL_RCDT_SX

Description

If during the reciprocal value calculation an overflow happens then the parameters DPLL_RCDT_TX.RCDT_TX and DPLL_RCDT_SX.RCDT_SX are set erroneously to 0x000000. The specified value is 0xFFFFFF.

Workaround

If a different settling behaviour of the DPLL control loop is acceptable no specific countermeasure is necessary.

ERR051823: GTM: (MCS) Unexpected instruction execution while disabling of MCS channel

Description

A disable request initiated by a write access MCS[i]_CH[x]_CTRL.EN = 0 might cause the following unexpected side effects if the MCS is not configured in Round Robin Scheduling mode and the following conditions are met:

1) Assume that an MCS channel x is disabled after the execution of an instruction instr1. If a potential successor instruction instr2 of instruction instr1 is a memory instruction executing a parallel memory write access and the delay between instr2 and instr1 is up to 3 cluster clock cycles, the write access of instruction instr2 might be executed unexpectedly after the MCS channel is already disabled.

2) Assume that an MCS channel x is disabled after the execution of an instruction instr1. If a potential successor instruction instr2 of instruction instr1 is a bus master instruction executing a bus access and the delay between instr2 and instr1 is up to 2 cluster clock cycles, the access of instruction instr2 might be executed unexpectedly after the MCS channel is already disabled.

Workaround

Provide a disabling feature by MCS program, e.g.:

1) Reserve a memory cell in MCS RAM and define value 1 as a request for a MCS channel disable.

2) Instead of writing MCS[i]_CH[x]_CTRL.EN = 0 write value 1 to reserved memory cell.

3) Poll the reserved memory cell during idle time of MCS program and switch off the MCS channel with instruction

MOVL STA 0x0

if the reserved memory cell contains value 1.

ERR051824: GTM: (DPLL) DPLL_DCGI interrupt not triggered

Description

When synchronous motor control mode is active (DPLL_CTRL_1.SMC=1): If a first direction change together with an input signal change (active edge) has happened, then for a consecutive direction change together with the next following input signal change the interrupt DPLL_DCGI does not occur.

Workaround

When a direction change is detected by DPLL_IRQ_NOTIFY.DCGI the Register DPLL_STATUS.BWD1 can be checked after the next relevant input signal edge on TRIGGER. If a second direction change is detected with the very next relevant input signal, the DPLL_DCGI can be set by writing DPLL_IRQ_FORCINT.DCGI =1. The next relevant input signal edge is the next input signal edge for DPLL_CTRL_1.SMC=1 (In contrast to the next inactive input signal edge when DPLL_CTRL_1.SMC=0).

ERR051825: GTM: (DPLL) Incorrect calculation of DPLL_THVAL, DPLL_THVAL2

Description

In case of LOW_RES=1, DPLL_CTRL_1.SMC=0, DPLL_CTRL_0.IDT=1, and DPLL_CTRL_0.TS0_HRT=0 the values of DPLL_THVAL, DPLL_THVAL2 are calculated incorrectly because the filter values are not divided by 8 as specified.

Workaround

If a negative effect on the direction decision is not expected no workaround is necessary. If a negative effect cannot be excluded the use of the filter values can be switched off by setting DPLL_CTRL_0.IDT=0.

ERR051832: GTM: (GTM_AEI) Turning off BRIDGE_MODE.MSK_WR_RSP in asynchronous mode might lead to following transactions being corrupted

Description

If the AEI bridge operates in asynchronous mode and in pipelined protocol, with Mask-Write-Response turned on (BRIDGE_MODE[2:0]=="011") and the BRIDGE_MODE.MSK_WR_RSP is turned off (by writing BRIDGE_MODE[2:0]="001"), the following transaction might be corrupted by the AEI_READY not being set.

Workaround

Change BRIDGE_MODE.MSK_WR_RSP together with setting the soft-Reset (pipeline writing BRIDGE_MODE[16:0]=h#10001)

ERR051833: GTM: Interrupt from DPLL not detected in MCS0

Description

Some of the DPLL interrupts can be detected in the internal registers DSTA and DSTAX of MCS0.

If the clock divider of cluster 0 is configured to 0b10 (2:1 clock ratio) inside GTM_CLS_CLK_CFG.CLS0_CLK_CFG and DPLL_IRQ_MODE.IRQ_MODE is configured to 0b01, 0b10, or

0b11, the interrupt pulse cannot be detected inside MCS0 due to an connectivity failure on GTM_IP toplevel and gets lost.

Workaround

Workaround 1:

Use interrupt level mode in DPLL by setting of DPLL_IRQ_MODE.IRQ_MODE=0b00.

Workaround 2:

Configure the cluster clock divider of cluster 0 to 1 by setting of GTM_CLS_CLK_CFG.CLS0_CLK_CFG=0b01.

ERR051834: GTM: (DPLL) Wrong value of DPLL_INC_CNT1.INC_CNT1 upon switching to normal mode

Description

DPLL_CTRL_0.RMO 1 -> 0:

Upon switching from emergency to normal mode (with DPLL_CTRL_1.SGE1 set to 1), DPLL_INC_CNT1.INC_CNT1 increments by DPLL_MLS1.MLS1 micro ticks every time an active STATE input is encountered till the first active TRIGGER input is encountered. The extra micro ticks accumulated in DPLL_INC_CNT1.INC_CNT1 will only be generated after encountering the first active TRIGGER.

The described behavior is not intended because the STATE input is not supposed to contribute to the pulse generation in normal mode.

Workaround

Two possible workarounds for DPLL_CTRL_0.RMO: 1 -> 0:

(1) Defer setting DPLL_CTRL_1.SGE1 to 1 till the first DPLL_TASI interrupt is encountered (signaling the arrival of the first active TRIGGER).

(2) Make sure that DPLL_MLS1.MLS1 is set to zero upon switching the mode. The user may then alter it on encountering the first DPLL_TASI interrupt.

ERR051835: GTM: (MCS) Missing hazard detection for parallel memory write access

Description

The MCS is configured in a scheduling mode unequal to round robin (MCS[i]_CTRL_STAT.SCD_MODE != 0b01) and is executing an instruction instr2 immediately after instruction instr1, which means that there is no delay in between both instructions. If instruction instr1 is a bus master read instruction that is executing a fast access (e.g. BRD R1, TIM_CH0_CTRL) and instruction instr2 is a memory write access that is executing a parallel memory access (e.g. MWRI R1, R2) via harvard architecture:

In this case the MCS does not resolve any hazard induced by a write after read data dependency between both instructions. As a consequence the data that is written by the parallel memory write access does not reflect the read data of instr1.

Example for such a sequence:

MOVL R1, 0xBAD7

...

BRD R1, TIM_CH0_CTRL #instr1

MWRI R1, R2, 0 #instr2

In the case of a fail in this example, the memory write access is writing the value 0xBAD7 instead of the content of configuration register TIM_CH0_CTRL.

The following instructions are memory write access instructions that are affected by this erratum: MWR, MWRI, MWRIO, PUSH, CALL, and CALLI.

Workaround

Add an additional NOP instruction between bus master read instruction and memory write instruction.

ERR051848: S32Z2E2: Possible Core and Peripheral PLL jitter spec violation due to edge-aligned I/O activity

Description

Core & peripheral PLLs can exceed jitter specifications when I/O pads on the VDD_IO_CD supply domain are configured as outputs and toggle in an edge-aligned fashion. “Edge-aligned” means that multiple I/Os are transitioning at the same time and aligned to the same clock edge. The PLLs are more affected by this issue at lower VCO frequencies.

Workaround

Recommended Workaround: Configure a VCO Frequency of 2400MHz for both Core and Peripheral PLL. In this case, there is no limitation on the number of I/Os which may be configured as outputs in domain VDD_IO_CD.

Alternate Workaround: Lower VCO frequencies may be used, if necessary, but in this case there is a limitation on the number of I/Os which may be configured as outputs (output buffer enabled). This limitation is based on the lesser of the two VCO frequencies for the Core and Peripheral PLLs.

For 2400MHz > VCO >= 2000MHz, up to 16 I/Os in domain VDD_IO_CD may be configured as outputs.

For 2000MHz > VCO >= 1800MHz, up to 6 I/Os in domain VDD_IO_CD may be configured as outputs.

For 1800MHz > VCO >= 1600MHz, up to 4 I/Os in domain VDD_IO_CD may be configured as outputs.

For 1600MHz > VCO >= 1300MHz, up to 3 I/Os in domain VDD_IO_CD may be configured as outputs.

Note: I/Os which do not toggle during operation of other I/Os in the segment (e.g. enable/disable level signals) may be excluded from this total limitation on outputs.

ERR051887: NETC: VLAN qualifiers missed in ingress classification lookups

Description

The Ingress Port Filter (Table ID 13) and Ingress Stream Identification (Table ID 30) lookups can optionally include in the matching key outer and inner VLAN tag information from the received fame.

For the Ingress Port Filter tale, when Hardware constructs the lookup key for a frame header field for which the frame header is not present in the frame, it sets the key field to zero. If zero corresponds to a valid value in a frame header field, then this could lead to a false positive match. To avoid these false positive matches, present frame header (or field) 1-bit key matching fields (or frame attribute flags) are defined in the table entry to explicitly specify whether the presence of a frame header (or field) must be present or not present in a frame header, for an entry to match.

For VLAN headers, an Outer VLAN Tag Present attribute flag and Inner VLAN Tag Present attribute flag have been defined. The expectation is that each of these present matching key flag will be set if parser has found a VLAN tag in their respective position (outer/inner) in the received frame, and it has passed the TPID acceptance criteria for the tag position in the frame (defined in Port TPID acceptance register (PTAR)). Currently the issue (or behaviour) is that each of these present matching key flag will be set if parser has found a VLAN tag in their respective position (outer/inner), regardless of the TPID acceptance criteria for the tag position in the frame. This is applicable for frames received from either the Switch or an ENETC.

For the Ingress Stream Identification table, when Hardware constructs the lookup key for a frame the same behaviour is used as the Ingress Port Filter table - the VLAN tag is found in the frame and has passed the acceptance criteria. Currently the issue (or behaviour) is that if VLAN is not found in the frame or hasn’t passed the acceptance criteria, the default VLAN information is used in the lookup key. This is only applicable for frames received from an ENETC.

Workaround

Option A (Ingress Stream Identification only): Do not use the VLAN tag present/not present attribute in the Ingress Stream Identification lookup.

Option B: Add an entry with higher precedence (Ingress Port Filter) or using the first key construction (Ingress Stream Identification) that contains the VLAN attributes (VID, PCP) of the default VLAN. Configure these entries with the applicable result actions for frames that use the default VLAN information. Frames that do not contain the default VLAN information will match an entry with a lower precedence (Ingress Port Filter) or during the second Ingress Stream Identification lookup.

ERR051964: CEVA_SPF2: Vector store instruction data loss

Description

When a scalar store instruction occurs after a vector store instruction and the gap between the two instructions is less than seven cycles, then the scalar and vector stores will collide, which might cause a Write-after-Write (WAW) hazard and a loss of data for the vector store instruction.

Workaround

Ensure that there are three NOP instructions between a vector store instruction and a subsequent scalar store instruction.

All of the code inside the functions is already protected by the compiler; this workaround is required only on code between functions.

ERR052004: STCU: MBIST reports single bit errors as failures

Description

The Self-Test Control Unit's Memory Built-In Self-Test (STCU MBIST) does not distinguish between single and multiple bit failures. As a result, it will report failure for single bit errors even if the errors can be corrected by ECC. This deviates from the intention of relying on ECC to protect against single bit latent reliability issues.

Workaround

After the STCU reports an MBIST failure, run additional diagnostic software to determine if the failure is due to single bit or multiple bit errors. The detailed workaround procedure is supplied, for each affected device, in NXP’s Safety Peripheral Drivers (SPD) package which is available on NXP’s website (https://www.nxp.com). This package is provided free of charge and is also included as part of the premium Safety Software Framework (SAF).

ERR052024: NETC: Half duplex transmit throughput degraded by up to 2.3%

Description

When configured for half duplex mode, the theoretical maximum transmit throughput is degraded by up to 2.3%. This issue can occur when transmitting back-to-back frames in the absence of any receive traffic and/or collisions. When transmitting back-to-back frames in half duplex mode, the minimum Inter-packet Gap (IPG) is 14 rather than 12 bytes times, resulting in the degradation. The transmit performance degradation decreases for larger frame sizes as follows:

Frame Length (B)	% Degradation
64	2.3
128	1.3
256	0.7
512	0.4
1024	0.2
1518	0.1

Workaround

For devices such as RT1180 supporting flexible preamble, the preamble length can be decreased by 2 if possible to recover the lost IPG time. There is no workaround on other devices without flexible preamble support. The application must tolerate the reduced transmit throughput when configured for half duplex.

ERR052026: CAAM: AES-GCM may generate an incorrect MAC

Description

When running AES-GCM (Galois Counter Mode) if the length of the IV (Initialization Vector) is not 12 bytes, the Seq Num value (32-bit counter value) may roll over. When this occurs (unless the counter starts at -1), the AES operation will generate an incorrect MAC, because the Seq Num overflows and corrupts the MAC. This occurs even when -1 is used as the counter for the last block (16-byte).

Workaround

The AES operation must be stopped after initialization to determine the Seq Num (32-bit counter value) to check whether and when it will roll over. Then, before the offending block is processed, the context needs to be saved. The one offending block gets processed twice:

1. Perform AES-GCM

2. Restore MAC then generate GMAC over its ciphertext

3. Restore the message and AAD lengths from the saved context and continue as normal

ERR052031: NETC: PTCaTSDR registers are implemented in the wrong order within the memory map

Description

The PTCaTSDR registers have been implemented at the following address offsets within ENETC as follows:

0x390 = traffic class 4

0x394 = traffic class 5

0x398 = traffic class 6

0x39C = traffic class 7

0x3A0 = traffic class 0

0x3A4 = traffic class 1

0x3A8 = traffic class 2

0x3AC = traffic class 3

The intended register order as per the reference manual should be:

0x390 = traffic class 0

0x394 = traffic class 1

0x398 = traffic class 2

0x39C = traffic class 3

0x3A0 = traffic class 4

0x3A4 = traffic class 5

0x3A8 = traffic class 6

0x3AC = traffic class 7

Workaround

Access the PTCaTSDR registers using the implemented register ordering described above rather than the intended order specified in the documentation, or set/clear the registers' TSDE bits to the same value for all 8 classes.

ERR052097: CANEXCEL: NO Underrun reporting for RX MD with match if KL=0

Description

If CANEXCEL receives a frame and an RXMD match is available such that the RXMD state is EMPTY but KL=0 (DCSTAn[STATE]=0b1 and RXCTRLn[KEEPLST]=0b0), then CANEXCEL does not issue UNDERRUN notification (URUNACKn) to the system.

Workaround

If KL=1 (RXCTRLn[KEEPLST]=0b1), RX MD, state must not become EMPTY (DCSTAn[STATE]=0b1) and LOCK must be available for CANEXCEL to receive incoming frame in this RXMD along with UNDERRUN notification.

If KL=0 (RXCTRLn[KEEPLST]=0b0), RX-MD state must not become EMPTY (DCSTAn[STATE]=0b1) to avoid UNDERRUN condition. System must program FIFOCTRLn[FIFWTMn] < FIFOCTRLn[FIFODPHn] and Lock must be available for CANEXCEL, so that UNDERRUN condition does not occur. System must perform Push for more pointers DCSYSPUSHn as soon as it receives MSGIFLAGn notification.

ERR052126: Arm Errata 2918152: [Cortex-R52] Executing LDM instruction might cause data corruption when HSCTLR.FI is set

Description

Affects: Cortex-R52

Fault Type: Programmer Category B

The Cortex-R52 processor supports a variety of asynchronous exceptions, such as interrupts and System Error aborts. The processor can also be optionally configured with floating-point instructions support. Although Cortex-R52 is an in-order processor, the floating-point divisions and square root operations complete out-of-order with respect to other instructions. Because of this erratum, a rare combination of a floating-point division (or square root) instruction, a load-multiple instruction, and an asynchronous exception might cause the contents of one of the integer registers to be corrupted.

Workaround

You can avoid this erratum by doing either of the following.

Workaround 1:

Disable out-of-order floating-point divisions and square roots, by setting CPUACTLR.OOODIVDIS to 1. There might be a minimal performance overhead. Indicative measurements show that the performance of several known benchmarks are impacted by less than 0.1%.

Workaround 2:

Disable Fast Interrupts, by clearing HSCTLR.FI to 0. The processor interrupt latency might be affected, at the worst case by waiting for 128 bytes of memory to be read.

ERR052133: NETC: MAC does not detect mVerify after start fragment

Description

A link partner should be able to initiate preemption verification at any time, e.g. for a port reset or SW reconfiguration of a port. If the MAC receives an mVerify packet after a start fragment, and before the corresponding ending continuation fragment, it will not recognize the mVerify packet, and the link partner verify process will time out.

Workaround

The recommended workaround in a closed system is to disable preemption verification. On NETC, that is done by setting MAC_MERGE_MMCSR[VDIS]=1 before setting [ME]=1.

In an open system where VDIS must be 0, system SW must provide an alternate mechanism for signaling when preemption verify has failed on the link partner. SW can then clear the MAC receiver state by toggling PM1_COMMAND_CONFIG[RX_EN], which will abort the preemptable packet in progress, after which the MAC will be able to receive mVerify packets again.

ERR052207: NETC: SG_DROP_COUNT value in the Ingress Stream Count STSE_DATA response begins at an incorrect bit offset, causing it to be read incorrectly.

Description

The Ingress Stream Count (Table ID 38) contains multiple statistic counters. When the statistics are queried by the software, one of the counters in the response buffer, namely SG_DROP_COUNT is returned in the response buffer shifted. No information is lost, but the placement of the information in the response buffer is sifted compared to expectation per the reference manual.

Workaround

Adjust the starting offset for the SG_DROP_COUNT statistic in the response buffer from the definedby 7-bits

The actual offset of the SG_DROP_COUNT in the STSE_DATA element is bit 199

Or the offset within the entire response buffer format is bit 231

ERR052209: CMU: AutoFI (Fault Injection) fixed mode is not executed after running Standalone Fixed FI mode with mask enable bit set.

Description

When Standalone Fixed FI (Fault Injection) mode is executed on CMU keeping GCR[FI_INT_MASK] bit set, after completion of Standalone Fixed FI mode there is no FHH/FLL seen in status register since GCR[FI_INT_MASK] is set but FHH/FLL is generated in reference clock domain which is expected. Because of this when Automatic Fixed FI mode is initiated after Standalone Fault Injection Mode, it doesn't start.

The issue is not seen when GCR[FI_INT_MASK] is not set and same sequence is used.

Workaround

Do not set GCR[FI_INT_MASK] during Standalone Fixed FI configuration whenever software wants to run AutoFI Fixed mode after Standalone Fixed FI mode.

ERR052223: TMU: Temperature monitor averaging quantization error

Description

The TMU Mode Register TMR[ALPF], if set to a non-zero value, may introduce quantization errors that can exceed the allowable temperature sensor error range (TERR) specified in the datasheet. If ALPF is set to 1, TERR maximum is increased and the minimum is decreased by 1 DegC. If ALPF is set to 2, TERR maximum is increased and the minimum is decreased by 2 DegC. If ALPF is set to 3, TERR maximum is increased and the minimum is decreased by 4 DegC.

Workaround

If the increased TERR specification cannot be tolerated in the application, TMR[ALPF] must be set to 0 to disable averaging. Software may use the immediate temperature reported and calculate a running average. The TMU averaging interrupts (AHTT, AHTCT, ALTT, and ALTCT) may still be used, but software needs to be aware that these interrupts are reacting to immediate temperature readings as averaging has been disabled per this workaround.

ERR052226: SWT: Toggling watchdog enable may cause unexpected timeout in some boundary conditions

Description

The Software Watchdog Timer (SWT) may timeout unexpectedly when loading a new timeout value. This can occur when the SWT is paused (CR[WEN]=0b0) to update the TO[WTO], while the counter is less than 0x14 (CO[CNT] = 0x14). When SWT is re-enabled (CR[WEN]=0b1), the SWT resumes the cycle count, but the counter is not updated (CO[CNT]) with the new timeout value before the cycle counter reaches zero.

Workaround

Before setting a new timeout value (TO[WTO]) the SWT must be updated with the watchdog keys ((SR[WSC]= 0xA602) and then (SR[WSC] = 0xB480)) to restart the counter value and have the timeout change being made within the appropriate time window, preventing the counter from reaching zero.

ERR052243: TMU : Unexpected temperature readings during monitoring mode

Description

Invalid temperature measurements may be observed across the temperature range specified in the device data sheet. The invalid temperature can be read from any remote site and from any capture or report registers. The invalid change in temperature can be positive or negative and the resulting temperature can be outside the calibrated range, in which case the TSR[ORL] or TSR[ORH] bit will be set. If the TSR[ORL] or TSR[ORH] bit is set, as noted in the reference manual, only immediate interrupts (IHTT and ILTT) will assert. As a result, only the modules connected to these interrupts will be alerted.

Workaround

If an FCCU module exists, do not configure any fault reaction associated with the TMU as reset.

For SoCs with an HSE-H firmware implementation, do not set the attribute to enable the TMU monitoring inside the HSE. For HSE-M firmware implementation, do not change the High or Low Temperature Average Critical Thresholds (TMHTACTR and TMLTACTR) as set by HSE firmware and do not set the Rising or Falling Temperature Rate Critical Threshold Interrupt Enables (RTRCTIE and FTRCTIE) in the Interrupt Enable (TIER) register. The HSE Firmware Reference Manual and the HSE FW release notes should be reviewed for any updates related to this erratum.

Implement a software-based filter to detect if an invalid measurement is reported by the TMU and respond accordingly. This can be accomplished in many ways with the following being an example.

Establish a temperature change absolute value between two successive measurements that will define an invalid measurement. The temperature change absolute value should be greater than the acceptable change in die temperature between two successive temperature measurements in the application. This temperature change absolute value can be used to filter the application from invalid measurements reported by the TMU.

If the application detects a potentially invalid measurement, either through manual polling of the rise or fall rate registers (TMRTRCR and TMFTRCR) or, for HSE-H firmware implementations only, through the associated enabled interrupts, compare the values of the TMRTRCR and TMFTRCR registers after the next temperature measurement has been completed by the TMU.

If the absolute value of the difference of these two measurements is less than the established acceptable change, the initial measurement is to be discarded for all enabled sites and no reaction needs to be taken. Then, write a 1 to clear the Valid Reading bit (bit 31) of both rise and fall rate registers (TMRTRCR and TMFTRCR) to start recording new data.

ERR052262: NETC: Minimum-size frames are not preempted

Description

This behavior affects frame preemption when used together with Credit Based Shaping (FQTSS, AKA Qav) or Enhancements for Scheduled Traffic (EST AKA Qbv), where NETC's TGE field =1. Furthermore, when using frame preemption together with Qbv, the behavior does not occur if the hold/release mechanism is used. This is the most common configuration for using Qbv and frame preemption together. When using frame preemption together with Qav, this behavior will be seen.

In the TGE=1, non-hold/release mode, the behavior is that NETC does not initiate preemption of frames that are exactly the smallest possible preemptible size.

Preemption initiation requires that a minimum number of bytes have been transmitted for a fragment and the number of bytes remaining for the preempted fragment satisfies the minimum size of a valid Ethernet frame (64 bytes). The minimum non-final fragment size is specified in the field Remote Additional Fragment Size (RAFS) of the Port MAC Merge Control and Status register (MAC_MERGE_MMCSR).

The defect is that hardware is using a minimum non-final fragment size of the value specified in MAC_MERGE_MMCSR[RAFS] + 1 byte. Illustrative examples per RAFS setting under the defect:

- For RAFS=0, a 124B fragment (including FCS) should, but will not, be turned into 60B+FCS and 60B+FCS fragments. This fragment will not be preempted. A 125B fragment can be preempted and become 61B+FCS and 60B+FCS fragments.

- For RAFS=1, a 188B fragment (including FCS) should, but will not, be turned into 124B+FCS and 60B+FCS fragments. This fragment will not be preempted. A 189B fragment can be preempted and become 125B+FCS and 60B+FCS fragments.

- For RAFS=2, a 252B fragment (including FCS) should, but will not, be turned into 188B+FCS and 60B+FCS fragments. This fragment will not be preempted. A 253B fragment can be preempted and become 189B+FCS and 60B+FCS fragments.

- For RAFS=3, a 316B fragment (including FCS) should, but will not, be turned into 252B+FCS and 60B+FCS fragments. This fragment will not be preempted. A 317B fragment can be preempted and become 253B+FCS and 60B+FCS fragments.

Note that even should this behavior be observed, the implementation remains compliant with IEEE 802.1Q/802.3 specification for Interspersed Express Traffic (IET).

Workaround

No workaround is required. The utility of the preemption feature is not impaired in any significant way.

ERR052288: NETC: The first frame after SequenceRecoveryReset (FRER) is dropped and miscounted

Description

There is a slight variation from the standard in the implementation of the IEEE 802.1CB RECOVERY_TIMEOUT event.

IEEE 802.1CB specifies that the SequenceRecoveryReset function is to be executed immediately after a RECOVERY_TIMEOUT event has occurred, meaning that the first packet received after the RECOVERY_TIEMOUT event, is processed using the sequence recovery function reset variables. In other words, the SequenceRecoveryReset should take effect as soon as the timeout expires.

However, the defect (or variation from the standard) is that the SequneceRecoveryReset only takes effect after the first frame arrives after the RECOVERY_TIMEOUT event . This first frame should be processed using the sequence recovery function reset variables, but it is likely to be discarded as it is processed using the sequence recovery function variables values prior to the RECOVERY_TIMEOUT event.

Below is a more detailed description of how the implementation is processing a RECOVERY_TIMEOUT event.

1. When a RECOVERY_TIMEOUT event occurs, the next packet received is processed by the sequence recovery function under normal operation using the current “RecovSeqNum” and “SequenceHistory” values.

2. Once completing the processing of the first packet received after the RECOVERY_TIMEOUT event, the SequenceRecoveryReset function is executed, which sets the “RecovSeqNum” to “RecovSeqSpace - 1”, clears the“SequenceHistory” array and sets “TakeAny” to true. The next packet received (i.e. second packet received after the RECOVERY_TIMEOUT event has occurred) is processed using the sequence recovery function reset variables.

Workaround

If given conditions of the erratum are met (a rare occasion), a workaround should be implemented at the system level e.g., retransmitting lost packets if Rx end misses multiple frames in a sequence, or, tolerate and account for one additional packet packet drop following a Sequence Recovery function reset (multiple frame drop).

ERR052289: CANEXCEL: Unexpected Incorrect Configuration error detected during TX/RX MD Pointer Configuration

Description

If DMA transfer of Message Descriptor(MD) pointer configuration from system memory to CANEXCEL memory is ongoing and any other MD event (Push DSCCONTROL.DCSYSPUSH[n] or HW pointer incremented DSCCONTROL.DCSTAn[HWPOINTER]) occurs, then CANEXCEL reports an invalid Incorrect Configuration error and enters into error mode SIC.SYSS[IERR]=0b0001.

Workaround

This issue is avoided by installing the patch described in NXP Technical Note TN00185, available at https://www.nxp.com. This procedure is performed during initialization of the CANEXCEL module after receiving Freeze Acknowledge from CANEXCEL and during freeze mode.

ERR052290: CANEXCEL: Incorrect RX-SMB Overrun reporting in case of XL frame with DLC less than 64.

Description

• CANEXCEL sets unexpected RX SMB Overrun Error SIC.SYSS[CRXOERR] for the following cases and drops the incoming frame-

o XL frame with Data Length Code (DLC) value less than 64, CANEXCEL is not able to process the Acceptance Field (AF) of the incoming frame by the time the frame is completely received in CANEXCEL memory.

o If next incoming frame is of smaller DLC and is successfully received by the time DMA is busy serving previous frames transfer.

o If RX Control field is received after ID filtering of the frame has been completed and incoming frame is of smaller DLC.

Workaround

This issue is avoided by installing the patch described in NXP Technical Note TN00185, available at https://www.nxp.com. This procedure is performed during initialization of the CANEXCEL module after receiving Freeze Acknowledge from CANEXCEL and during freeze mode.

ERR052291: CANEXCEL: Data Inconsistency observed between transmitted and received frame in case of LOA

Description

CANEXCEL might not be able to transmit a correct frame data on the CAN-BUS, if transmitting frame data is not completely copied from system memory to CANEXCEL memory before Loss of Arbitration (LOA) is observed for the current frame. Hence the frame stored in a system memory for transmission is different from frame transfer on the CAN BUS. If the frame is copied completely from system memory to CANEXCEL memory before the LOA is observed on the CAN BUS, then CANEXCEL will send valid frame data.

Workaround

This issue is avoided by installing the patch described in NXP Technical Note TN00185, available at https://www.nxp.com. This procedure is performed during initialization of the CANEXCEL module after receiving Freeze Acknowledge from CANEXCEL and during freeze mode.

ERR052350: CANEXCEL: MSG_DESCRIPTORS.TXCTRL_n[PRIO] > 3 is not considered in case of re-insertion.

Description

If MSG_DESCRIPTORS.TXCTRL_n[PRIO] > 3 and in case of retransmission, CANEXCEL considers only 2 bits of PRIO rather than 4 bits.

Workaround

System must program MSG_DESCRIPTORS.TXCTRL_n[PRIO] to value <=3.

ERR052356: CANEXCEL: Soft-Reset limitation

Description

Soft-Reset cannot be used while CANEXCEL is transmitting or receiving

Workaround

Place CANEXCEL in freeze mode before using soft-reset

ERR052367: GTM: (MCS) Unexpected parallel memory access after MCS error

Description

If an MCS enters an error state (e.g. due to a division by zero) and its pipeline has already fetched a subsequent memory access instruction with a parallel memory access, the access is initiated although the instruction should be suppressed.

The unexpected memory access occurs only, if the MCS is not configured in round robin mode and the delay between the occurrence of the error and the memory access instruction is either 1 or 2 cluster clock cycles.

Workaround

Change scheduling of MCS program in a way that no parallel memory access can occur only after 2 cluster clock cycles of an instruction that can possibly enter the error state (e.g. adding 2 extra NOPs).

ERR052368: GTM: (MCS) Unexpected parallel memory access on ECC error

Description

If the MCS from cluster i fetches a memory access instruction where an ECC error is reported on its input MCSi_RAMx_ECC_ERR, the initiating MCS channel is stopped reliably and the ECC error is reported as expected.

However, if the corrupted memory access instruction code would be a parallel memory access, the MCS will initiate that memory access instead of suppressing it.

Workaround

Application software that responds to the ECC error report in the parallel memory access case needs to be aware the access has also occurred in addition to being reported.

ERR052374: GTM: (DPLL) Unregular pulse generation and wrong PMT results

Description

With the configuration of DPLL_CTRL_NUTC.NUTE = 2*(DPLL_CTRL_0.TNU +1)-1 for TRIGGER or DPLL_CTRL_NUSC.NUSE = 2* (DPLL_CTRL_0.SNU +1) -1 for STATE the prediction of the pulse generation frequency is incorrect.

This results in unregular pulse generation and wrong PMT results.

This problem occurs on either CCM[0]_TBU_TS1 or CCM[0]_TBU_TS2 or both depending on the DPLL operation mode (normal or emergency mode, or synchronous motor control).

Workaround

Avoid these configurations: DPLL_CTRL_NUTC.NUTE = 2*(DPLL_CTRL_0.TNU +1)-1 or DPLL_CTRL_NUSC.NUSE = 2* (DPLL_CTRL_0.SNU +1) -1.

Instead, use for example full scale configurations as described in the specification.

ERR052376: GTM: ((A)TOM) Missing edge on output signal (A)TOM_OUT

Description

If an (A)TOM channel is configured to be triggered by a previous channel by setting of (A)TOM[i]_CH[x]_CTRL.RST_CCU0=1 (SOMP mode in ATOM) and there is a pipeline/synchronization register within the trigger chain between the triggering channel and the triggered channel, the edge to the inverse SL at the output signal (A)TOM_OUT is not generated for (A)TOM[i]_CH[x]_CM1.CM1<2 and (A)TOM[i]_CH[x]_CM0.CM0>(A)TOM[i]_CH[x]_CM1.CM1.

The problem only occurs if the selected clock resolution for the triggered channel has a divider factor of more than 1 related to the cluster clock CLS[i]_CLK.

Since GTM Gen3, the problem does not occur if the pipeline/synchronization register is internally of (A)TOM module and the clock divider for the cluster clock CLS[i]_CLK is configured with a clock divider of 2 by setting of GTM_CLS_CLK_CFG.CLS[j]_CLK_DIV = 0b10.

Workaround

Workaround 1:

If available use channels without a pipeline/synchronization register within the trigger chain between the triggering channel and the triggered channel.

Workaround 2a - Applicable for the error case with (A)TOM[i]_CH[x]_CM1.CM1=1:

Switch the order of the edges, so that (A)TOM[i]_CH[x]_CM0.CM0 defines the first edge and (A)TOM[i]_CH[x]_CM1.CM1 the second edge. Additionally invert the SL value to get the same waveform on the output signal (A)TOM_OUT.

Note: In this case, to generate 0% duty cycle, still use (A)TOM[i]_CH[x]_CM1.CM1=0 and (A)TOM[i]_CH[x]_CM0.CM0>MAX.

Workaround 2b - Applicable for the error case with (A)TOM[i]_CH[x]_CM1.CM1=0:

Set (A)TOM[i]_CH[x]_CM0.CM0=MAX and (A)TOM[i]_CH[x]_SR0.SR0=MAX by writing them before the target period. Set (A)TOM[i]_CH[x]_CM1.CM1 to the original application value of (A)TOM[i]_CH[x]_CM0.CM0. Additionally invert the SL value to get the same waveform on the output signal (A)TOM_OUT.

Workaround 3:

Use a clock resolution for the triggered channel with a divider value of 1 related to the cluster clock.

ERR052381: GTM: ((A)TOM) Edge at output signal (A)TOM_OUT does not occur

Description

If the channel is configured to be triggered by a preceding channel with (A)TOM[i]_CH[x]_CTRL.RST_CCU0=0b1 (SOMP mode for ATOM) and the duty cycle switches from 100% duty cycle with (A)TOM[i]_CH[x]_CM0.CM0=0x0 and (A)TOM[i]_CH[x]_CM1.CM1>MAX to a left-aligned PWM or to 0% duty cycle with (A)TOM[i]_CH[x]_CM1.CM1=0x0 and (A)TOM[i]_CH[x]_CM0.CM0>0 for left-aligned PWM or (A)TOM[i]_CH[x]_CM0.CM0>MAX for 0% duty cycle, the expected edge on the output signal (A)TOM_OUT to the inverted (A)TOM[i]_CH[x]_CTRL.SL value does not occur.

Hint: If the setting after switching to a left-aligned PWM or to 0% duty cycle is not changed the edge appears at the beginning of the next period.

Workaround

Workaround 1:

In the period before the change to a left-aligned PWM or to 0% duty cycle, set the value of (A)TOM[i]_CH[x]_CM1.CM1 to MAX instead of greater than MAX. This can be done asynchronously by writing the bit field (A)TOM[i]_CH[x]_CM1.CM1 within the period.

Alternatively, it can be done via the synchronous update mechanism by writing the bit field (A)TOM[i]_CH[x]_SR1.SR1 two periods before switching to a left-aligned PWM or to 0% duty cycle.

Workaround 2:

For 0% duty cycle on the output signal use the setting for 100% duty cycle with (A)TOM[i]_CH[x]_CM0.CM0 = 0 and (A)TOM[i]_CH[x]_CM1.CM1 > MAX and toggle (A)TOM[i]_CH[x]_CTRL.SL value synchronously by writing to (A)TOM[i]_CH[x]_CTRL_SR.SL_SR.

ERR052384: GTM: CPU bus access is not acknowledged

Description

If a cluster is switched off by writing GTM_CLS_CLK_CFG.CLS[j]_CLK_DIV = 0b00 while the MCS inside this cluster is executing a bus access, further CPU bus accesses to the switched off cluster are not acknowledged with an AEI_READY signal and the corresponding AEI_STATUS = 0b10.

Workaround

Disable the MCS bus access by writing MCS_AEM_DIS.DIS_CLS[j] = 0b10 before switching off the cluster with GTM_CLS_CLK_CFG.CLS[j]_CLK_DIV = 0b00. Enable the MCS bus access by writing MCS_AEM_DIS.DIS_CLS[j] = 0b01 after switching on the cluster again.

ERR052386: GTM: (MCS) No resume action after clearing hardware breakpoints

Description

If both hardware breakpoints are configured to potentially trigger at the same point in time (e.g. instruction breakpoint at the same address), and the selected scope is a system halt (MCS[i]_HBP[h]_CTRL.SCOPE=2 with h=0 and h=1), the MCS enters the halted state as expected if both breakpoints are hit. However, after a corresponding write access to both of the HBP status registers (MCS[i]_HBP[h]_STATUS with h=0 and h=1) the MCS erroneously doesn't resume its execution.

Workaround

Avoid breakpoint configurations for both HBPs that will activate the system halt at the same point in time.

ERR052403: FlexCAN: CAN frame drops in Enhanced RX FIFO when message buffer (MB) is locked for more than 1 CAN frame time (33 us)

Description

If Message Buffer (MB) and Enhanced RX FIFO both are configured for reception, and FlexCAN Message Buffer is locked for a long time (more than 1 CAN frame, 33us), FlexCAN receives some frames in FIFO and then start dropping the frames.

Workaround

There are two possible workarounds:

1) Core must read the message buffer within the 33 us after locking the MB.

2) Avoid using a few specific Message Buffers (MBs) as listed below:

ERR052438: FlexCAN: CAN frame may drop when using Enhanced RX FIFO

Description

An incoming CAN frame will be lost (i.e not latched into its expected Enhanced Rx FIFO data element), if both following two conditions are met simultaneously. There will be no indication that the frame was lost.

Conditions:

1. A write access is made to the message buffer Control and Status word (MB_CS) of a specific message buffer corresponding to the expected Enhanced Rx FIFO data element. Each Enhanced Rx FIFO data element corresponds to different message buffers impacted by this erratum and cannot be determined by software.

2. Depending on the timestamp configuration, the write access is made when receiving a frame at one certain Controller Host Interface (CHI) clock cycle either:

a. Around the time between the seventh bit of EOF and the second bit of IFS if timestamp is disabled (CTRL2[TSTAMPCAP] = 00b) or

b. Around the time between the fifth bit of EOF and seventh bit of EOF if timestamp is enabled (CTRL2[TSTAMPCAP] = 01b or 10b or 11b)

Workaround

To avoid the potential for dropped CAN frames, one of the following options may be implemented:

Workaround #1 : Disable Enhanced RX FIFO feature.

Workaround #2 : If the Enhanced RX FIFO feature is enabled, restrictions apply to certain Message Buffer (MB) numbers for both RX and TX. Either do not use these MB's at all or at minimum, avoid updating the Control and Status word of these MBs when any reception to the Enhanced Rx FIFO could occur. This means, it would be safe to update the Control and Status word of these MBs when the FlexCAN is for example in Freeze mode or when it is ensured against frame reception from the CAN bus. Below are the MB numbers that have these restrictions:

ERR052453: FlexCAN: TX frame is not able to transmit when message buffer is programmed in a very small-time window during reception

Description

A transmit (TX) frame is not able to transmit when the message buffer (MB) is programmed in a very small-time window during reception.

The message remains in the message buffer and will get transmitted if any other message buffer is programmed for transmission or the same message buffer is overwritten, or frame reception is occurring.

Workaround

If the number of Message Buffers in use is less than or equal to 126, this issue will not occur. For example, if there are 128 total message buffers available, do not use MBs #127 and #128.

ERR052455: AES_ACCEL: Missing CRC check on upper part of a 256 bit key

Description

The ACE keystore supports both 128 bit and 256 bit keys. To install a 256 bit key, two keyslots are linked together by setting the SIZE flag of the first keyslot. The second keyslot then provides the remaining 128 bits of key data. The keyslots are protected by a 16 bit CRC pattern. This pattern is checked each time a key is loaded for a crypto operation.

However, when a 256 bit key is loaded, only the attributes and key values of the first key slot are checked, the 128 bit of key data originating from the second keyslot are not checked. If one of these 128 bits in the second keyslot was corrupted, the circuit would not detect this.

Workaround

Any one of the following two workarounds could be used:

1 - Instead of a 256-bit key a 128-bit key could be used.

Or

2 - Correct operation with a 256 bit key could be verified by a 3 step sequence - first executing the crypto operation, second using the key for an operation for which the outcome is already known and confirming it will provide the expected result, and third executing the crypto operation again to confirm that there wasn't a temporary error on the key during the first execution.

ERR052465: LFAST: LVDS receiver pad fault is latched when receiver is enabled before the common voltage is settled on the line

Description

During the LVDS Fast Asynchronous Serial Transmission (LFAST) startup procedure: It is possible for the LVDS receiver pad (Rx) to be enabled before the common voltage is reached on the line. If the LFAST startup procedure continues, communication between the Master and Slave might not be established correctly at high speed.

Workaround

In order to properly detect status of the line prior to starting LFAST communiction, the LVDS pad receiver state has been routed to a GPR and an FCCU fault that can be sampled by software. If the LVDS receiver pad (Rx) is enabled prior to the common mode voltage being reached, a fault is asserted to LVFCCU1 (bits 18 and 19 corresponding to LFAST 0 and 1 respectively). These indicators are also routed to GPR1 LVFCCU1S register bits 18 and 19 (corresponding to LFAST0 and LFAST1 status). When the proper voltage is reached, the fault indicator from the pad is cleared and the GPR status register will be cleared (set to 0). Software should check for this 0 status in the GPR, followed by clearing the FCCU fault status, before finally continuing the LFAST startup procedure.

ERR052501: NETC: PCE does not report correct memory ID for faults from Replication and Egress Packet memory

Description

The PCE block processes the table management commands for the Replication and Egress Packet memory (switch only). If there is a multi-bit ECC, or EDC (if applicable), error during a table management command, the incorrect memory ID will be reported (Common Memory).

Workaround

There is no workaround to avoid the issue. Application software should be aware that the incorrect memory ID is reported, but this has no impact on the recovery action which should involve a full reset regardless of which memory was the source of the error.

ERR052547: OMU: Initiator hang may occur when disabling OMU via OMU_OER[OE]

Description

When disabling the Overlay Management Unit (OMU) via clearing OMU_OER[OE]=0 while AXI traffic is flowing through the OMU, it is possible that OMU will drop one AXI transaction on the address read channel. If this occurs, it will likely result in a hang scenario for the requesting initiator. A similar hazard exists when enabling OMU via setting OMU_OER[OE]=1 while AXI traffic is flowing through OMU. This case will not result in a hang but can result in other undefined behavior and should be avoided.

Workaround

When calibration remap services are needed, the application software must ensure that OMU is enabled via OMU_OER[OE] while there is no traffic flowing through the AXI interface. When calibration is complete and remap services are no longer needed, the application software must use OMU zone disable (OMU_ZERn[ZxE]=0) rather than disabling via clearing OMU_OER[OE].

ERR052558: FlexCAN: Message buffer (MB) overrun status is cleared when reading Enhanced RX FIFO (ERF)

Description

Message buffer status becomes "full" when a frame arrives, and status becomes "overrun" when a second message arrives in the same message buffer, if first message has still not been read. If frame reception is happening in ERF and the frame is being read from ERF, these reads could incorrectly clear the MB overrun status. As a result, the overrun event can be missed by the application.

Workaround

Use one of the following workarounds:

Workaround #1: Don’t use Enhanced RX FIFO (ERF).

Workaround #2: Don’t use any of the message buffers from MB0 to MB7 for reception if ERF is enabled. MB0 to MB7 can be used for transmission.

ERR052663: NETC: missing support for 802.1Qci PSFP streamBlockedDueToOversizeFrame

Description

The 802.1Q-2022 standard defines the ability to drop oversized frames in the Stream Filter table with streamBlockedDueToOversizeFrame parameter (see 8.6.5.3.1 Maximum SDU Size Filtering). Once an oversized frame has been dropped, the Stream Filter instance can be configured to block all incoming frames after.

The defect is that the ability to block all subsequent frames, after an oversized frame, is missing.

Workaround

None. streamBlockedDueToOversizeFrame feature is not available.

Software could identify such discard then add a rule to discard that flow. (but this would not be a tight coupling between detection and blocking).

ERR052805: DDRC: Can Violate Refresh Recovery for Second Rank

Description

If two ranks are enabled for the DDR Controller (DDRC), then the refresh recovery time can be violated by 1 DRAM cycle from an auto refresh to a following self-refresh (issued for a periodic training) for the second rank. This will be a violation if the calculated refresh recovery time divided by 2 has a remainder of 1.

Workaround

If the currently calculated refresh recovery (or per-bank refresh recovery) divided by 2 has a remainder of 1, then add 1 to this value. This applies to {TIMING_CFG_3[EXT_REFREC], TIMING_CFG_1[REFREC]}. This also applies to TIMING_CFG_9[REFREC_PB] if using per-bank refreshes.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Suitability for use in automotive applications (functional safety) — This NXP product has been qualified for use in automotive applications. It has been developed in accordance with ISO 26262, and has been ASIL classified accordingly. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as “Critical Applications”), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

1	**Mask Set Errata for Mask 0P91J**	**1**
1.1	Revision History	1
1.2	Errata and Information Summary	2
2	**Known Errata**	**10**
	Legal information	**100**

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: https://www.nxp.com

Document feedback

Date of release: 1 July 2025
Document identifier: S32E2_0P91J